

## Video Analysis

Date	11 November 2022
Team ID	PNT2022TMID42321
Project Name	Emerging Methods for Early Detection of Forest Fires

### OpenCV For Video Processing

[Click here for openCV file \(Colab\)](#)

#### openCV.ipynb:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
import cv2
import numpy as np
from google.colab.patches import cv2_imshow
from matplotlib import pyplot as plt
import librosa
from tensorflow.keras.preprocessing import image
from keras.models import load_model
```

```
# Create a VideoCapture object and read from input file
# If the input is the camera, pass 0 instead of the video file name
cap = cv2.VideoCapture('/content/drive/MyDrive/Dataset/forestfire.mp4')
```

```

# Check if camera opened successfully
if (cap.isOpened() == False):
    print("Error opening video stream or file")

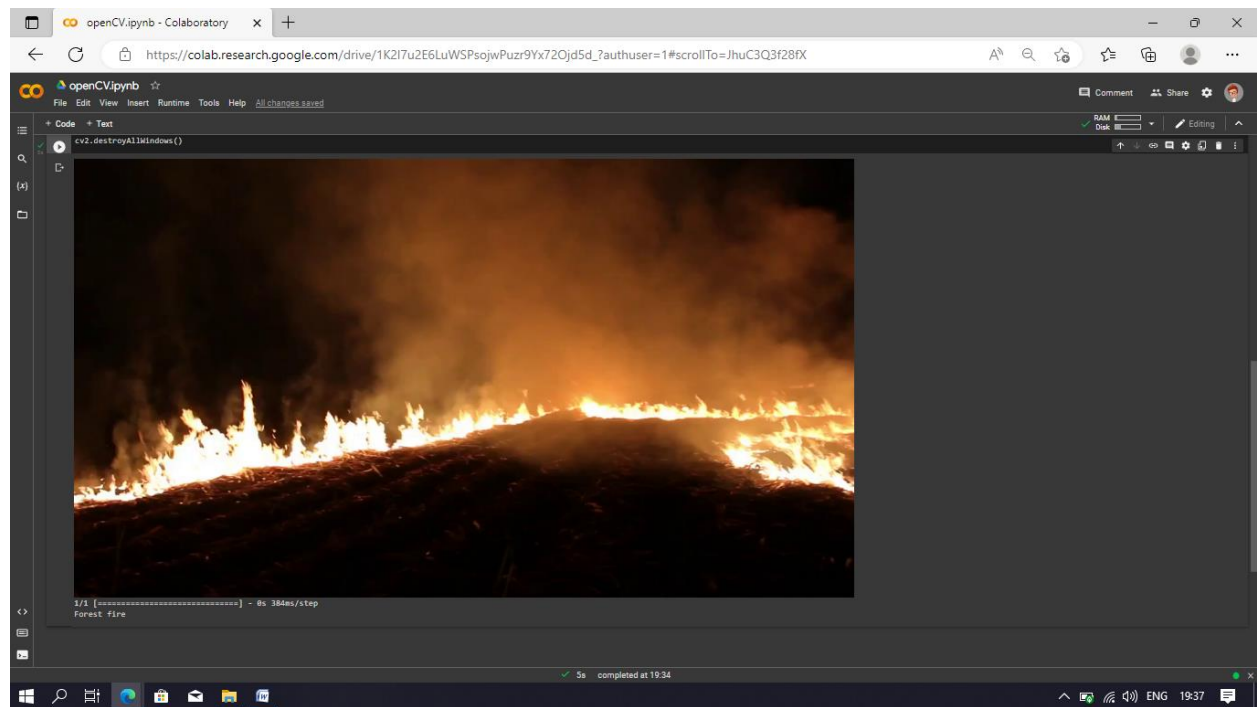
# Read until video is completed
while(cap.isOpened()):
    # Capture frame-by-frame
    ret, frame = cap.read()
    if ret == True:
        x=image.img_to_array(frame)
        res=cv2.resize(x,dsize=(64,64),interpolation=cv2.INTER_CUBIC)
        #expand the image shape
        x=np.expand_dims(res,axis=0)
        model=load_model("/content/drive/MyDrive/Dataset/forest.h5")
        cv2_imshow(frame)
        pred=model.predict(x)
        pred = int(pred[0][0])
        pred
        int(pred)
        if pred==0:
            print('Forest fire')
            break
        else:
            print("no danger")
            break

# When everything done, release the video capture object
cap.release()

# Closes all the frames
cv2.destroyAllWindows()

```

output:



```
1/1 [=====] - 0s 384ms/step
Forest fire
```

# Creating An Account In Twilio Service

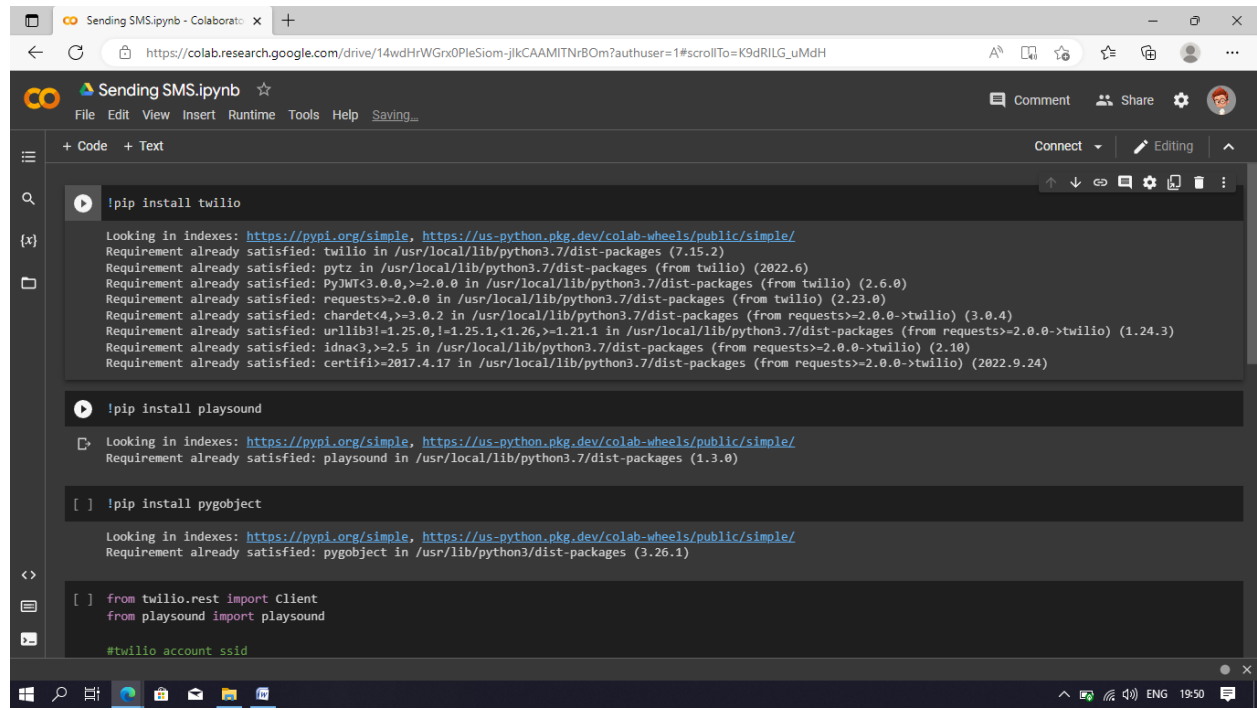
## Login to Twilio & Buy a number:

The screenshot shows the Twilio console interface. The top navigation bar includes 'Console', 'My first Twilio account', a trial status of '\$14.252 Upgrade', a search bar, and links for 'Account', 'Billing', and a user profile 'Akash'. The left sidebar has a 'Develop' tab and a 'Monitor' tab. Under 'Develop', the 'Phone Numbers' section is expanded, showing 'Manage' and 'Active numbers' (which is highlighted). The 'Active Numbers' page displays a table of active phone numbers. The table has columns for 'Number', 'Friendly Name', 'Capabilities' (Voice, SMS, MMS, Fax), and 'Active Configuration'. A single number is listed: '+1 479 397 4371' with a friendly name '(479) 397-4371 US, US'. The active configuration shows webhooks for voice and messaging. Below the table, there are navigation buttons 'Previous' and 'Next'. A disclaimer at the bottom states: '\* Can send/receive calls to domestic numbers only', '† Can send/receive sms to domestic numbers only', '‡ This number does NOT support SIP Trunking', '▲ Can make emergency calls.', and '(national) A non-geographic number'.

## Account SID and Authentication token:

The screenshot shows the Twilio console interface for 'Auth Tokens - United States (US1)'. The top navigation bar is the same as the previous screenshot. The left sidebar has an 'Account' tab and a 'Keys & Credentials' section. Under 'Keys & Credentials', 'API keys & tokens' is selected. The main content area shows 'Auth Tokens - United States (US1)' with a description: 'Auth tokens can be used to authenticate while making API requests. You will need to use HTTP Basic Authentication with user=Account SID and password=AuthToken. Auth tokens are specific to your account and can be used to access all APIs for the account. Please keep the auth tokens in a secure place and rotate them periodically. [Learn more](#), [🔗](#)'. Below this, there are two sections: 'Live credentials' and 'Test credentials'. The 'Live credentials' section shows the 'Account SID' as 'ACc0b32842aa3060ee6f4b2bfa1116247f' and the 'Auth token' as 'ACe7eab6ce6ccb80b5cced3cd20985857a'. A warning box states: 'Sensitive information. Store your token securely to protect your account. [Learn more](#) [🔗](#)'. The 'Test credentials' section shows the 'Test Account SID' as 'ACe7eab6ce6ccb80b5cced3cd20985857a' and the 'Test Auth token' as 'ACe7eab6ce6ccb80b5cced3cd20985857a'. A note at the bottom says 'Keep this somewhere safe and secure'.

## Task 1: Use API to send alert messages: (Sending SMS.ipynb)



The screenshot shows a Google Colab notebook interface. The top bar includes the Colab logo, the notebook title "Sending SMS.ipynb", and a star icon. Below the title is a menu bar with "File", "Edit", "View", "Insert", "Runtime", "Tools", "Help", and "Saving...". On the right side of the top bar are icons for "Comment", "Share", and a user profile. The main area of the notebook is a code editor with a dark background. It contains three code cells. The first cell starts with a play button icon followed by the command `!pip install twilio`. The output of this cell shows the package manager looking in various indexes and confirming that twilio (2.6.0) is already satisfied. The second cell also starts with a play button icon followed by `!pip install playsound`. The output shows playsound (1.3.0) is already satisfied. The third cell starts with a play button icon followed by `!pip install pygobject`. The output shows pygobject (3.26.1) is already satisfied. The bottom of the notebook shows a Windows taskbar with various icons and the system clock indicating 19:50.

```
!pip install twilio

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: twilio in /usr/local/lib/python3.7/dist-packages (7.15.2)
Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from twilio) (2022.6)
Requirement already satisfied: PyJWT<3.0.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from twilio) (2.6.0)
Requirement already satisfied: requests>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from twilio) (2.23.0)
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)
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)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests>=2.0.0->twilio) (2022.9.24)

!pip install playsound

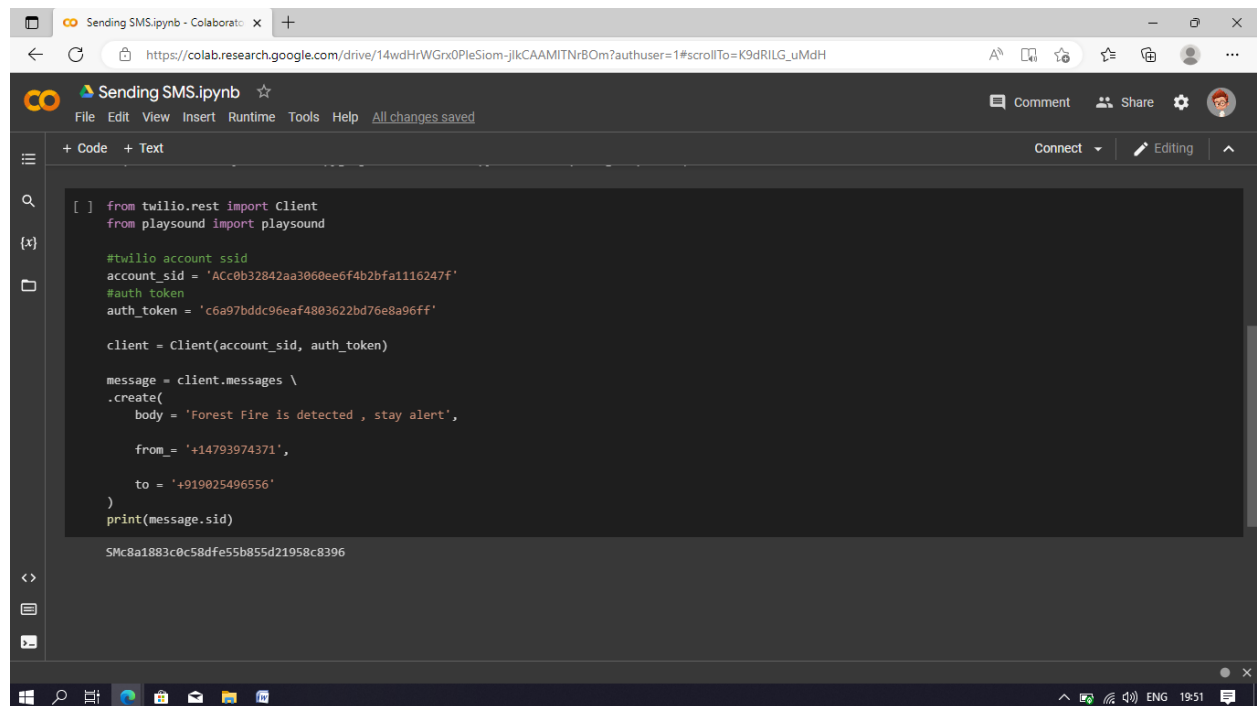
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: playsound in /usr/local/lib/python3.7/dist-packages (1.3.0)

[ ] !pip install pygobject

Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Requirement already satisfied: pygobject in /usr/lib/python3/dist-packages (3.26.1)

[ ] from twilio.rest import Client
    from playsound import playsound

#twilio account ssid
```



The screenshot shows the same Google Colab notebook interface, but now with a single code cell containing the code to send an SMS. The code imports the Client from twilio.rest and playsound from playsound. It defines the account\_sid, auth\_token, and auth\_token variables. It then creates a Client object. The code then creates a message object and calls the create method to send an SMS. The body of the message is "Forest Fire is detected , stay alert". The from\_ number is "+14793974371" and the to number is "+919025496556". The code prints the message.sid. The output of the code cell shows the message.sid value: "SMc8a1883c8c58dfe55b855d21958c8396". The bottom of the notebook shows a Windows taskbar with various icons and the system clock indicating 19:51.

```
[ ] from twilio.rest import Client
    from playsound import playsound

#twilio account ssid
account_sid = 'Acc0b32842aa3060ee6f4b2bfa1116247f'
#auth token
auth_token = 'c6a97bddc96eaf4803622bd76e8a96ff'

client = Client(account_sid, auth_token)

message = client.messages \
    .create(
        body = 'Forest Fire is detected , stay alert',
        from_ = '+14793974371',
        to = '+919025496556'
    )
print(message.sid)

SMc8a1883c8c58dfe55b855d21958c8396
```

**Screenshot of the SMS received:**

