

## ✓ Real-time Stream Simulation & Event Trigger

### Vivek's Computer Vision based stream event trigger

Simulates a real-time stream, performing detection on every third frame and triggering an alert if three or more people appear on the screen (three's a crowd!). The alerts are logged in a JSON file with timestamps, and are also plotted with a timeline of alert occurrences. The alerts are overlayed and the output visuals are exported.

**Pre-requisites:-** Rename the short video to 'input\_video.mp4' in the current directory, and create two folders: 'output\_video' and 'output\_images' in the directory you run your code from

Install the necessary packages

```
!pip install opencv-python
!pip install ultralytics
!pip install matplotlib
```

```

Requirement already satisfied: opencv-python in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: numpy<2.3.0, >=2 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: ultralytics in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: numpy>=1.23.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: matplotlib>=3.3.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: opencv-python>=4.6.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: pillow>=7.1.2 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: pyyaml>=5.3.1 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: requests>=2.23.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: torch>=1.8.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: torchvision>=0.9.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: tqdm>=4.64.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: psutil in /usr/local/lib/python3.11/dist-packages (fr
Requirement already satisfied: py-cpuinfo in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: pandas>=1.1.4 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: ultralytics-thop>=2.0.0 in /usr/local/lib/python3.11/c
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: cycycler>=0.10 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.11/dist
Requirement already satisfied: nvtx>=2020.1 in /usr/local/lib/python3.11/dist-packages

```

```

Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packa
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (f
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.11
Requirement already satisfied: networkx in /usr/local/lib/python3.11/dist-packages (f
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages (frc
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages (frc
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.4.127 in /usr/local/lib/pyt
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.4.127 in /usr/local/lib/p
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.4.127 in /usr/local/lib/pyt
Requirement already satisfied: nvidia-cudnn-cu12==9.1.0.70 in /usr/local/lib/python3.
Requirement already satisfied: nvidia-cublas-cu12==12.4.5.8 in /usr/local/lib/python3
Requirement already satisfied: nvidia-cufft-cu12==11.2.1.3 in /usr/local/lib/python3.
Requirement already satisfied: nvidia-curand-cu12==10.3.5.147 in /usr/local/lib/pythc
Requirement already satisfied: nvidia-cusolver-cu12==11.6.1.9 in /usr/local/lib/pythc
Requirement already satisfied: nvidia-cuspars-cu12==12.3.1.170 in /usr/local/lib/pyt
Requirement already satisfied: nvidia-cusparselt-cu12==0.6.2 in /usr/local/lib/pythor
Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in /usr/local/lib/python3.11/
Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in /usr/local/lib/python3.1
Requirement already satisfied: nvidia-nvjitlink-cu12==12.4.127 in /usr/local/lib/pyth
Requirement already satisfied: triton==3.2.0 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.11/dist-packag
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.11/dist-p
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (f
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-pack
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-pac
Requirement already satisfied: cycycler>=0.10 in /usr/local/lib/python3.11/dist-package
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-pa
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-pa

```

## Importing the libraries

```

from ultralytics import YOLO
import cv2
import json
import matplotlib.pyplot as plt
from google.colab.patches import cv2_imshow

```

## Loading the yolov10 model, the video, and defining integers and lists

```

model = YOLO('yolov10n.pt')

#loading video
video_path = '/content/StreamSim/input_video.mp4'
cap = cv2.VideoCapture(video_path) # use cv2.VideoCapture('cam.no.') for streaming

fps = cap.get(cv2.CAP_PROP_FPS) # get the fps of the video

```

```

fps = cap.get(cv2.CAP_PROP_FPS)          # get the fps of the video

frame_skip = 3                          # number of skipped frames
frame_index = 0                          # integer defining the number of the frame every
scale_percent = 50                       # percentage the video will be resized to
all_detections = []                      # stores detected labels, box-coordinates, conf.
time_list = []                           # stores the timestamps of people detected
people_count_list = []                   # stores the number of people detected

```

### Main Loop:

```

ret = True
while ret:
    ret, frame = cap.read()
    if ret:
        results = model.track(frame, persist = True)          # detect
        frame_ = results[0].plot()                             # plots the detected

        height, width, _ = frame_.shape                       # resizing the image
        new_height = int(height * scale_percent / 100)
        new_width = int(width * scale_percent / 100)
        resized = cv2.resize(frame_,(new_width,new_height), interpolation=cv2.INTER_AREA)

        if frame_index % frame_skip == 0:                     # executes every frame_skip frames
            frame_detections = []                               # stores detections
            person_detected = 0                                 # stores number of people detected

            for r in results:                                   # loop stores objects
                for box in r.bboxes:
                    cls_id = int(box.cls)
                    label = model.model.names[cls_id]
                    conf = float(box.conf)

                    if (label == 'person') & (conf > 0.5):    # if person is detected & confidence > 0.5
                        person_detected += 1
                        if person_detected >= 3:               # if more than 3 people, store
                            detection_time = frame_index / fps
                            frame_detections.append([label, detection_time, conf])
                            time_list.append(detection_time)
                            people_count_list.append(person_detected)

            if frame_detections:                                # If more than 3 people detected
                frame_json = {
                    "frame number": frame_index,
                    "time (seconds)": detection_time,
                    "frame detections": frame_detections
                }
                all_detections.append(frame_json)                # update all_detections

```

```
text_width = int(new_width/5)
text_height = int(new_height *3/4)
resized = cv2.putText(resized, "TOO MANY PEOPLE!!!",          #
                      (text_width,text_height),
                      cv2.FONT_HERSHEY_SIMPLEX , 0.7 , (0 , 0 , 255) , 2)

cv2.imshow( resized)          # used instead of cv2.imshow('frame', resized)
if cv2.waitKey(25) & 0xFF == ord('q'): # outputs image in window 'frame', waits for
    break
path = str('/content/StreamSim/output_images/frame_' + str(frame_index) + '.jpg')
cv2.imwrite(path, resized)          # ;

frame_index += 1          #
```















































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































































