

### **This use case is for generating alert:**

1. When person is detected in a given area.
2. Person count Live count is increased.
3. Live count is appended.

### **Workflow:**

1. The SSD model detects [person] class in the video source.
2. Each person detected is assigned an ID (Person ID).
3. This person ID is stored in a python list to make sure alerts for same ID is not generated again.
4. A config.json file is maintained, where following parameters are present:  
    vid\_source (video source[videofile, rtsp, camera])
5. The parameters can be modified from inside the config.json file.
6. The detection only runs inside the ROI defined using coordinates.
7. When person enters ROI the Live count is updated and displayed on the screen.
8. Using the length of object id, total count is also maintained keeping track of total number of person visiting the ROI.
9. A live person count is displayed on the inference, it shows the number of people present in the frame at the given time.

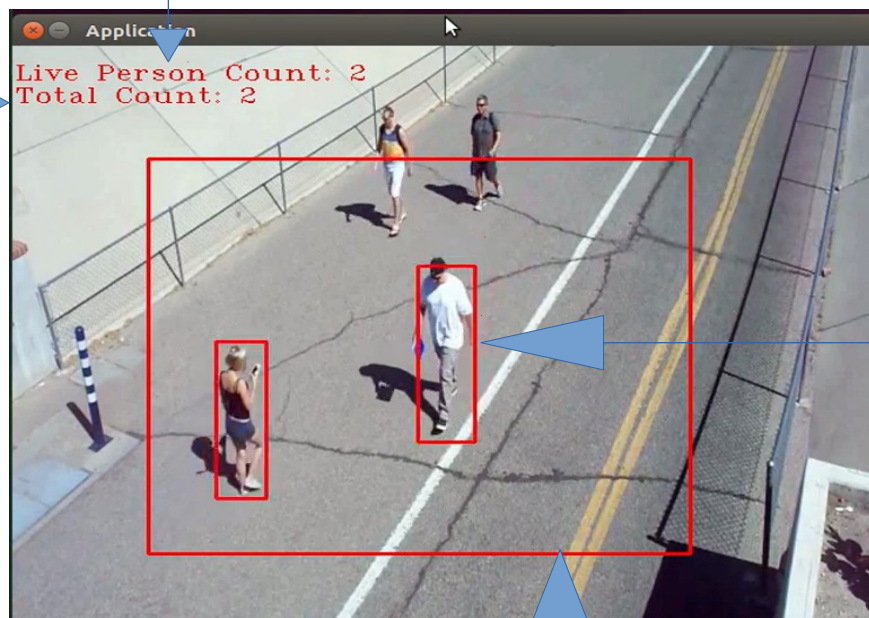
**Note:** To use Opencv with CUDA and use opencv\_dnn module install Opencv from source.  
In the script the following line of code needs to be added to use Opencv with CUDA

```
# set CUDA as the preferable backend and target
print("[INFO] setting preferable backend and target to CUDA...")
detector.setPreferableBackend(cv2.dnn.DNN_BACKEND_CUDA)
detector.setPreferableTarget(cv2.dnn.DNN_TARGET_CUDA)
```

**Backend:**Following inference is generated upon running the script

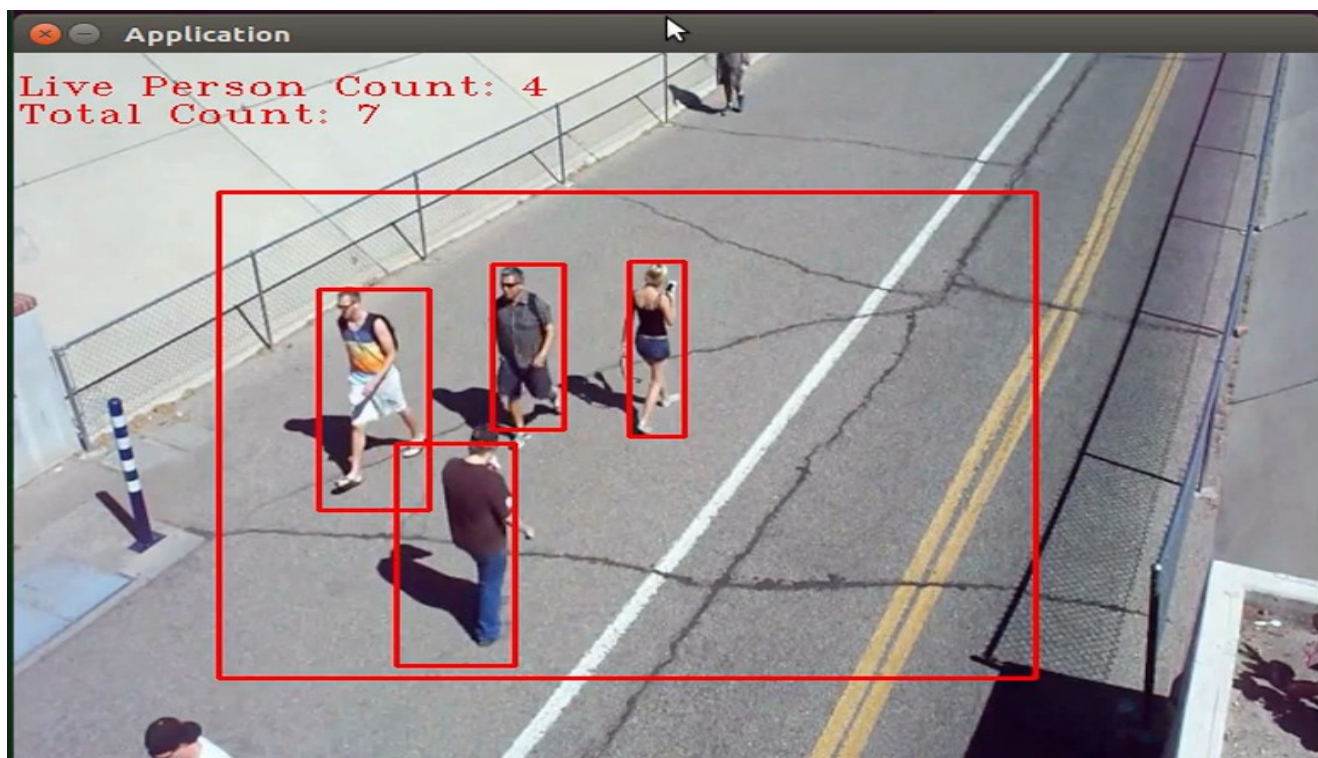
Live person count,  
display the number of  
people present in the  
**frame.**

Total Count,  
displays the  
number of  
person  
visited the  
**Frame.**



Bounding  
box for  
each person  
detected.

Bounding box for  
ROI



## **Inference using webcam on Jetson Nano:Technical Description:**

The usecase is built on jetson Nano(4GB dev kit)

The usecase inference was executed and tested on jetson Nano.

SSD Model is used for detecting people.

## **Project Structure: Person\_Detection\_and\_count**

```
--- 1. Person_Detection_and_count.py
--- 2. config.json
--- 3. model
--- MobileNetSSD_deploy.caffemodel
--- MobileNetSSD_deploy.prototxt.txt
--- 4. tracker
--- centroidtracker.py
```

**Parameters and values for generating alert are defined in config.json**  
**config.json contains the following parameters:**

1. vid\_source (video source[videofile, rtsp, camera])

**Note:** To use Opencv with CUDA and use opencv dnn module install Opencv from source.

## **Steps to run the the script:**

1. Define the following in the config.json file:

1. video source

2. Open the terminal in the location where Person\_Detection\_and\_count.py is present.

3. run the following command in the terminal

```
$ python3 Person_Detection_and_count.py
```