

SKP ENGINEERING COLLEGE

Approved by AICTE New Delhi | Affiliated to Anna University – Chennai
Tiruvannamalai, Tamil Nadu | Phone: +91-4175-252633 | +91-9443105139



DEPARTMENT OF INFORMATION TECHNOLOGY

IT3811 – Project work

Zeroth Review

PEOPLE COUNTING SYSTEM USING MOBILENET, SSD ALGORITHM BY DEEP LEARNING

Submitted by:

Manikandan. S (512221205010)

Ramanan. S (512221205015)

Ramkumar. A (512221205016)

Project Co-Ordinator:

MS.M.SAMHITHA ME.,

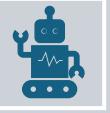
Project Guide:

Dr.V.Raji ME.,Ph.D.,

Objectives:

- **To implement a deep learning-based object detection model** (MobileNet SSD) capable of identifying and localizing people in video frames.
- **To develop a real-time people counting system** that processes continuous video streams and provides accurate headcounts per frame.
- **To utilize MobileNet architecture** for its lightweight nature, ensuring the system can operate efficiently on resource-limited devices.
- **To reduce errors caused by overlapping, occlusion, or dynamic movement** by optimizing detection parameters such as confidence threshold and frame resizing.
- **To display live video frames with bounding boxes and people count annotations** for better visualization and monitoring.
- **To create a modular system** that can be extended for applications such as crowd analysis, smart surveillance, public safety, or smart transportation.

Scope of the project:



Implementation of a lightweight deep learning model (MobileNet SSD) for object detection, specifically targeting the "person" class.



Real-time video processing to detect and count people in each frame from video files or live camera input.



Visual representation of results by drawing bounding boxes and displaying the number of people detected per frame.



Integration with simple hardware setups, including standard webcams and basic computing devices.



Scalability of the model for different environments such as offices, public transport, classrooms, or shopping areas.



Potential for future cloud integration (e.g., Firebase) for data storage and analysis.

Methodology:

Data Input (Video Stream) - The system begins by accepting a **video input**, either from a **pre-recorded file** or a **live webcam feed**.

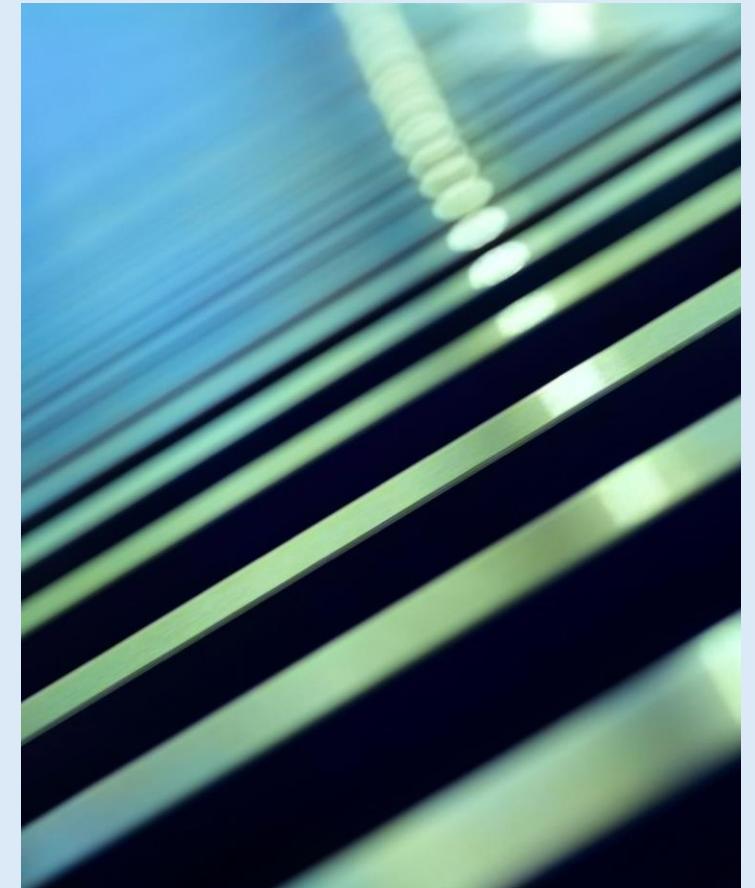
Pre-trained Model Loading - A **MobileNet SSD model**, pre-trained on the **COCO or PASCAL VOC dataset**, is used for object detection.

Frame Preprocessing - Each video frame is **resized and normalized** before being passed to the model.

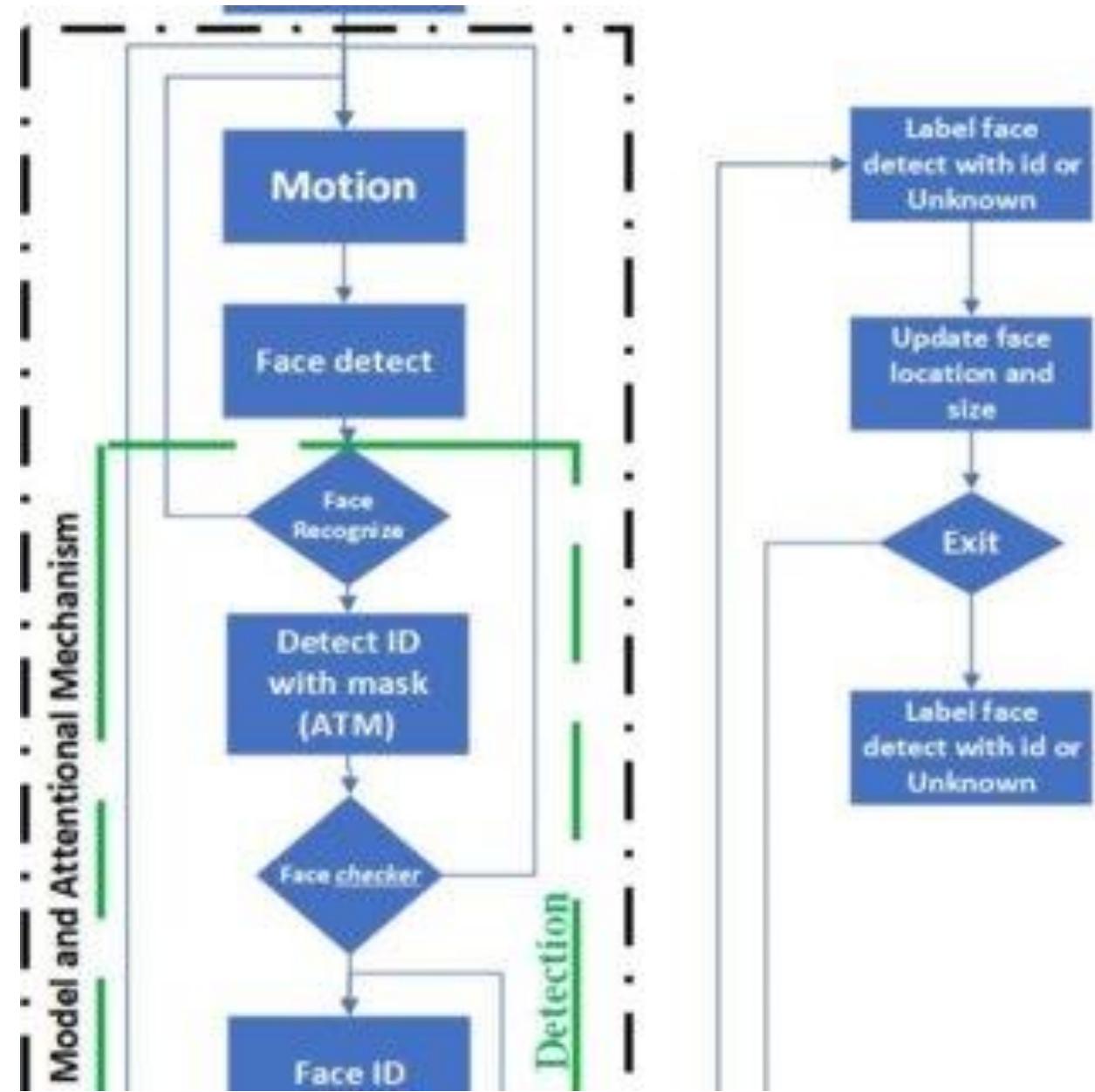
Object Detection (People Detection) - The frame is passed through the **MobileNet SSD network**, which returns detections of all recognized objects.

Counting and Visualization - The final count is displayed on the video feed in real-time.

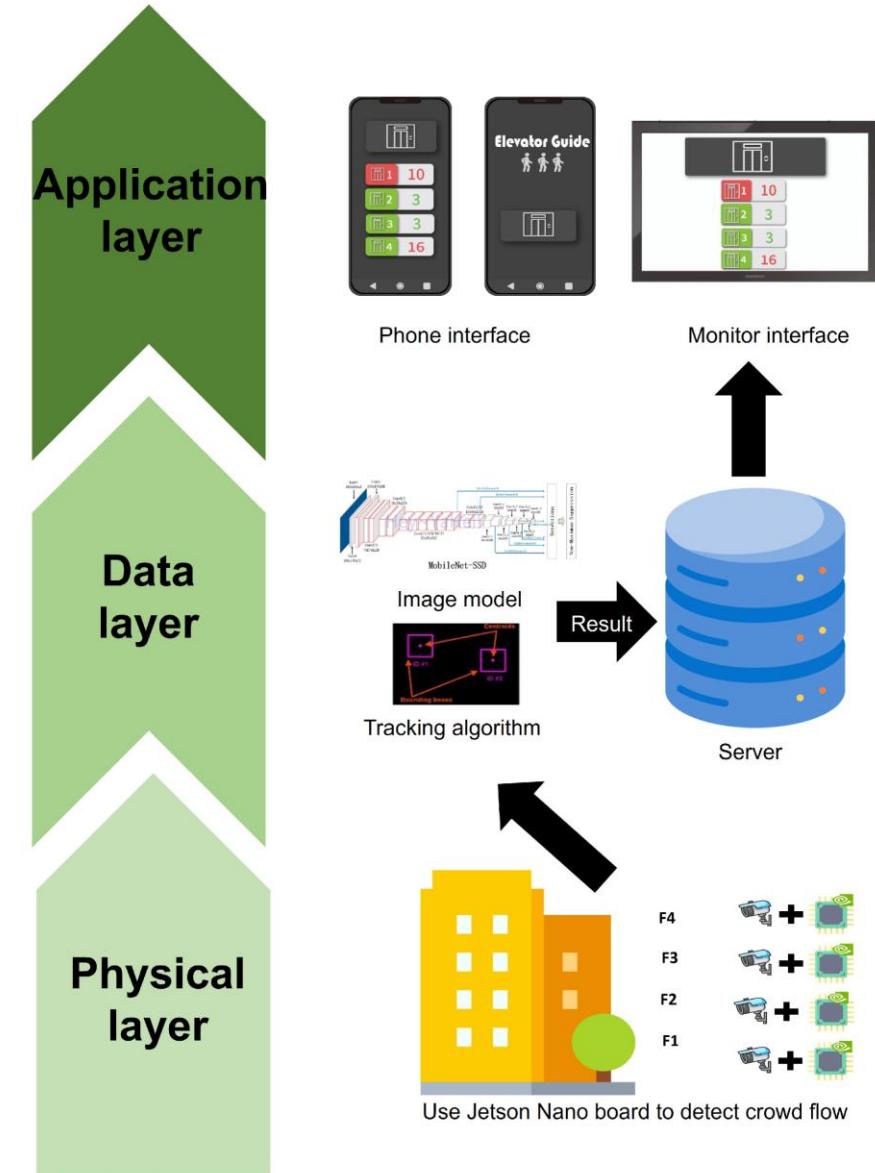
User Interaction and Exit - The system continuously processes frames until the user presses a defined **exit key** (e.g., 'q') to stop the execution.



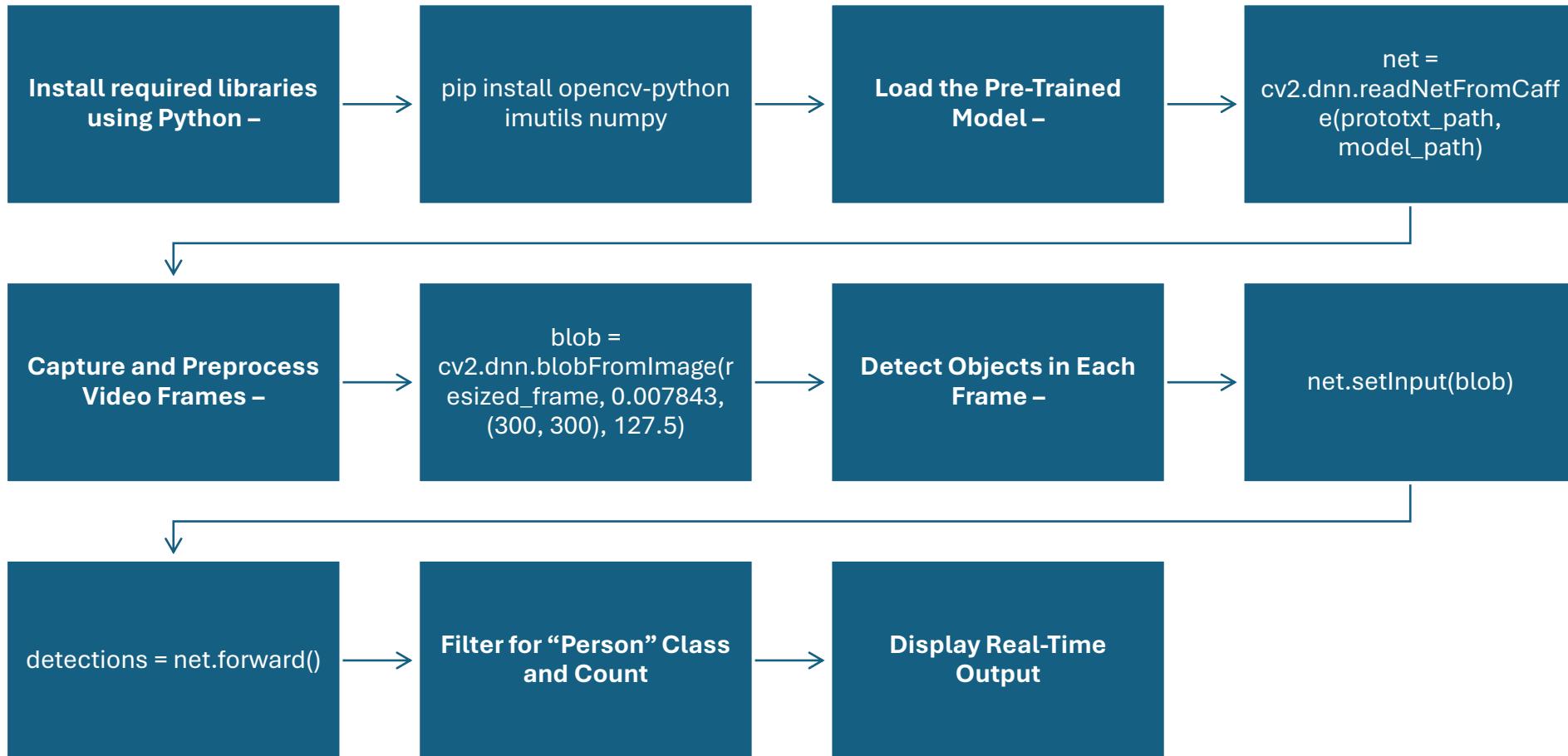
Use case diagram:



Architecture Diagram:



Implementation:



Demonstrate hardware and software components:

Component	Description
Computer or Laptop	Basic system with Python and OpenCV installed. Can be Windows, Mac, or Linux-based.
Webcam / USB Camera	For real-time live feed input (optional – for live detection).
Processor	Recommended: Intel i5/i7 or equivalent. MobileNet SSD can run on CPU.
RAM	Minimum 4GB; 8GB recommended for smooth performance.
Storage	At least 500MB free (for model files and video processing).
(Optional) Embedded Device	Raspberry Pi 4, NVIDIA Jetson Nano, or similar for deployment in smart environments.

Software components:

Component	Description
Python	Programming language used to build and run the system.
OpenCV	Library for image and video processing. Handles video input and frame manipulation.
NumPy	For numerical operations and matrix handling.
Imutils	Simplifies OpenCV functions like resizing and displaying frames.
MobileNet-SSD Model	Pre-trained deep learning model for detecting objects including people.
Caffe Framework	Model is originally trained in Caffe format and loaded using OpenCV DNN module.
IDE (e.g., VS Code, PyCharm)	Used for writing and debugging the Python script.
Operating System	Windows, Linux, or macOS – any OS that supports Python and OpenCV.

An aerial photograph of a long bridge spanning a wide body of water. The bridge has two distinct sections: a shorter one on the left and a much longer one extending towards the right. Both sections feature a dark grey asphalt surface with white dashed lane markings. On the left section, several vehicles are visible, including a white van, a blue truck, and a white car. The water surrounding the bridge is a vibrant turquoise color with small, rhythmic ripples. The overall scene conveys a sense of travel and infrastructure.

Thank you