A Project Report on

Pedestrian Eyes: An AI-Powered Framework for Real-Time Pedestrian Detection and Safety Analytics in Dhaka

United International University

QUEST FOR EXCELLENCE

A Project Report on

Pedestrian Eyes: An AI-Powered Framework for Real-Time Pedestrian Detection and Safety Analytics in Dhaka

Supervised By

Ahmed Imran Kabir

Asst. Professor

School of Business and Economics

United International University

Prepared By

Shifat Bin Azad

ID- 111 223 0184

Major: Management Information System

Date of submission: 24th November 2025

Letter of Transmittal

Date: 24 November 2025

Subject: Submission of Project Report on “Pedestrian Eyes: An AI-Powered Framework for Real-Time Pedestrian Detection and Safety Analytics in Dhaka”

Dear Sir,

I am pleased to submit my project report titled “Pedestrian Eyes: An AI-Powered Framework for Real-Time Pedestrian Detection and Safety Analytics in Dhaka.” This project integrates computer vision and sentiment analysis to understand pedestrian movement and safety concerns in Dhaka city.

I have tried my best to prepare the report with clarity, accuracy, and proper analysis. I hope the report will meet your expectations.

Sincerely,

Shifat Bin Azad
ID: 111 223 0184

Certification of Similarity Index

Pedestrian Eyes: An AI-Powered Framework for Real-Time Pedestrian Detection and Safety Analytics in Dhaka

ORIGINALITY REPORT

10%	8%	9%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	origin.geeksforgeeks.org Internet Source	1%
2	assets.researchsquare.com Internet Source	1%
3	Submitted to University of Melbourne Student Paper	<1%
4	arxiv.org Internet Source	<1%
5	Zhengyan Liu, Chaoyue Dai, Xu Li. "An Electric Bicycle Tracking Algorithm for Improved Traffic Management", Heliyon, 2024 Publication	<1%
6	ru.djvu.online Internet Source	<1%
7	www.mdpi.com Internet Source	<1%
8	Submitted to Higher Education Commission Pakistan Student Paper	<1%
9	journals.sagepub.com Internet Source	<1%
10	Jin Wu, Changqing Cao, Yuedong Zhou, Xiaodong Zeng, Zhejun Feng, Qifan Wu, Ziqiang Huang. "Multiple Ship Tracking in Remote Sensing Images Using Deep Learning", Remote Sensing, 2021 Publication	<1%

11	Submitted to Coventry University Student Paper	<1%
12	de.acervolima.com Internet Source	<1%
13	link.springer.com Internet Source	<1%
14	Manikandan Ravikiran, Yuichi Nonaka, Nestor Mariyasagayam. "A Sensitivity Analysis (and Practitioners' Guide to) of DeepSORT for Low Frame Rate Video", 2020 IEEE International Conference on Big Data (Big Data), 2020 Publication	<1%
15	Submitted to Monash University Student Paper	<1%
16	ts2.space Internet Source	<1%
17	Jianchen Wang, Jianguang Zhang, Xianbin Wen. "Non-full multi-layer feature representations for person re-identification", Multimedia Tools and Applications, 2020 Publication	<1%
18	Ye Li, Lei Wu, Yiping Chen, Xinzhong Wang, Guangqiang Yin, Zhiguo Wang. "Motion estimation and multi-stage association for tracking-by-detection", Complex & Intelligent Systems, 2023 Publication	<1%
19	www.videantis.com Internet Source	<1%
20	M. Bharathi, T. Aditya Sai Srinivas, P. Ravinder. "YOLOv8 on the Road: Next-Level Perception for Autonomous Vehicles", Journal of Image Processing and Image Restoration, 2024 Publication	<1%

21	Submitted to University of Nottingham Student Paper	<1%
22	mdpi-res.com Internet Source	<1%
23	www.surrey.ac.uk Internet Source	<1%
24	"PRICAI 2019: Trends in Artificial Intelligence", Springer Science and Business Media LLC, 2019 Publication	<1%
25	Submitted to Toronto Business College Student Paper	<1%
26	www.aixpaper.com Internet Source	<1%
27	orcid.org Internet Source	<1%
28	www.hindawi.com Internet Source	<1%
29	yongliangyang.net Internet Source	<1%
30	Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, Tao Xiang. "Learning Generalisable Omni-Scale Representations for Person Re-Identification", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021 Publication	<1%
31	Keqi Lu, Chao Zhu, Mengyin Liu, Xu-Cheng Yin. "OSS-OCL: Occlusion scenario simulation and occluded-edge concentrated learning for pedestrian detection", Pattern Recognition Letters, 2025 Publication	<1%

32	Li, Yicong. "Causality Model for Semantic Understanding on Videos", National University of Singapore (Singapore), 2025 Publication	<1%
33	assets-eu.researchsquare.com Internet Source	<1%
34	laganvalleydup.co.uk Internet Source	<1%
35	www.frontiersin.org Internet Source	<1%
36	Andres Paez, Christian Neumann, Hans-Günter Eckel. "Comprehensive Study on the Grid Fault Behavior of Grid-Forming Control for Modular Multilevel Converters", 2023 25th European Conference on Power Electronics and Applications (EPE'23 ECCE Europe), 2023 Publication	<1%
37	Craig Iaboni, Thomas Kelly, Pramod Abichandani. "NU-AIR: A Neuromorphic Urban Aerial Dataset for Detection and Localization of Pedestrians and Vehicles", International Journal of Computer Vision, 2025 Publication	<1%
38	Diogo R.M. Bastos, João Manuel R.S. Tavares. "A scalable gait acquisition and recognition system with angle-enhanced models", Expert Systems with Applications, 2025 Publication	<1%
39	Shangtao You, Zhengchao Gu, Kai Zhu. "Pedestrian detection method based on improved YOLOv5", Systems Science & Control Engineering, 2024 Publication	<1%
40	Ta-Sung Lee, Ming-Chun Lee, Chia-Hung Lin. "Wireless Communications and Sensing -	<1%

	Fundamentals, Signal Processing, and Machine Learning Solutions", CRC Press, 2025 Publication	
41	Vinit Katariya, Fatema-E- Jannat, Armin Danesh Pazho, Ghazal Alinezhad Noghre, Hamed Tabkhi. "VegaEdge: Edge AI confluence for real-time IoT-applications in highway safety", Internet of Things, 2024 Publication	<1 %
42	Weisong Shi, Yuankai He. "Chapter 5 Perception Algorithms", Springer Science and Business Media LLC, 2026 Publication	<1 %
43	Yuze He, Ke Chen, Juanjuan Hu. "PoseTrackNet: Integrating advanced techniques for accurate and robust human pose estimation in dynamic environments", Alexandria Engineering Journal, 2025 Publication	<1 %
44	Zhang, Xu-Yao, Yoshua Bengio, and Cheng-Lin Liu. "Online and offline handwritten Chinese character recognition: A comprehensive study and new benchmark", Pattern Recognition, 2017. Publication	<1 %
45	aran.library.nuigalway.ie Internet Source	<1 %
46	ouci.dntb.gov.ua Internet Source	<1 %
47	theses.hal.science Internet Source	<1 %
48	"Innovations in Data Analytics", Springer Science and Business Media LLC, 2025 Publication	<1 %

49	Dhirendra Kumar Shukla, Shabir Ali, Sandhya Sharma. "Artificial Intelligence and Sustainable Innovation - Volume 2", CRC Press, 2026 Publication	<1%
50	Lentin Joseph, Amit Kumar Mondal. "Autonomous Driving and Advanced Driver-Assistance Systems (ADAS) - Applications, Development, Legal Issues, and Testing", CRC Press, 2021 Publication	<1%
51	Mingwei Lei, Yongchao Song, Jindong Zhao, Xuan Wang, Jun Lyu, Jindong Xu, Weiqing Yan. "End-to-End Network for Pedestrian Detection, Tracking and Re-Identification in Real-Time Surveillance System", Sensors, 2022 Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

Table of Contents

Declaration of the Student

I hereby declare that the project report titled “Pedestrian Eyes: An AI-Powered Framework for Real-Time Pedestrian Detection and Safety Analytics in Dhaka” is based on my original work, and the undersigned, hereby solemnly swear. '**Ahmad Imran Kabir**' sir supervised it when I was a student.

I have prepared this report as per the instructions of the university. Each time I borrowed information or received help from another source, I acknowledged it and offered a reference with more information with just my supervisor's guidance, I was able to finish my project report alone. No other degree, diploma, or certificate program at this or any other university has yet considered submitting the work for consideration.

Shifat Bin Azad

ID: 111 223 0184

Acknowledgment

First, I express my deepest gratitude to Allah for granting me the strength to complete this project.

In addition, I'd want to express my gratitude to my esteemed project supervisor **Ahmed Imran Kabir** for all of the insightful feedback, moral support, close oversight, and general wisdom he provided during the tenure of this project. I value his support and direction-finding advice which helped me complete this project within my limitations.

I am also grateful to everyone who supported me throughout the development of this project. Their help, suggestions, and encouragement guided me toward completing the report successfully. I would like to specially thank the Dhaka City Corporation traffic division for their cooperation.

I thank my peers, mentors, and family members for their continuous support and motivation.

Abstract

The pedestrian environment in rapidly urbanizing cities such as Dhaka presents acute safety challenges due to high densities, informal crossings, encroached sidewalks and limited monitoring infrastructure. This paper presents a comprehensive system that integrates computer vision (pedestrian detection, multi-object tracking, re-identification and event detection) with an NLP-powered public-sentiment analysis module, targeted at pedestrian safety and flow monitoring in Dhaka. A lightweight detector is fine-tuned for local conditions, a tracker and re-id pipeline supports cross-camera flow analytics, and a public-sentiment module mines social media and local news to prioritize intervention zones. The system is designed for edge-server hybrid deployment, emphasizes privacy-preserving data handling and produces policy-relevant dashboards (counts, heat-maps, alerts). Preliminary experiments on urban footage show detection precision of ~85 %, tracking IDF1 = 72 %, and sentiment analysis accuracy of 78 %, demonstrating viability for municipal deployment and infrastructure planning within Dhaka city.

Keywords: Pedestrian Detection · Multi-Object Tracking · Re-Identification · Crowd Counting · Public Sentiment · Urban Safety · Bangladesh

Chapter 1. Introduction

In this research, I explore the development and implementation of a lightweight pedestrian monitoring system built with Python, OpenCV, and R. The aim of the project is to provide a simple but effective alternative to the limitations of traditional traffic-monitoring approaches in Dhaka, where pedestrian safety often receives less attention than motorized traffic. By combining computer vision with sentiment analysis, this system attempts to better understand how pedestrians move through the city and how they feel about their walking environment. Through automation and data-driven insights, the project hopes to support safer footpaths, better planning decisions, and a more pedestrian-friendly Dhaka.

1.1 Background of the Study

Dhaka faces long-standing challenges when it comes to pedestrian movement. Footpaths are uneven, crowded, and often occupied by vendors, while proper crossings and traffic signals remain limited. Manual observation of pedestrian conditions is slow, inconsistent, and unable to capture real-time changes across the city. Because of these limitations, traditional monitoring methods fail to provide a clear and continuous picture of pedestrian safety.

To address this gap, automated pedestrian detection using Python and OpenCV has become an increasingly relevant approach. Python's simplicity and the robustness of OpenCV make them suitable tools for building scalable urban-monitoring systems. At the same time, the rise of sentiment analysis offers a way to understand how people describe their daily walking experience using natural language. By analyzing both English and Bangla text, this project captures emotional responses that often remain unaddressed in numeric datasets.

Together, these technologies provide a more holistic view of walking conditions in Dhaka — blending visual evidence with public feedback. This study looks into how such a system can be built and how it might support safer urban mobility.

1.2 Statement of the Problem

Pedestrian monitoring in Dhaka still relies heavily on manual observation, occasional surveys, and isolated field reports. These methods are time-consuming, error-prone, and do not scale Ill for a city of over 20 million people. More importantly, they do not capture the emotional experience of pedestrians — such as frustration with broken sidewalks or safety concerns at night.

Key challenges include:

- Lack of continuous and automated pedestrian tracking

- Difficulty identifying pedestrian flows across busy intersections
- Poor visibility during low-light conditions
- Limited understanding of public sentiment related to walking safety
- Absence of integrated systems that combine both visual data and natural language feedback

To address these issues, an automated and flexible monitoring approach is required — one that uses computer vision to detect pedestrian presence and natural language processing (NLP) to interpret how people talk about their walking conditions. By integrating Python-based vision models with R-based sentiment analysis, this project aims to reduce inefficiencies, improve accuracy, and provide insights that manual observation alone cannot deliver.

1.3 Objectives of the Study

This research aims to examine and evaluate the development of a Python-based pedestrian detection and sentiment analysis system. The specific objectives are as follows:

● **Identify Requirements:** Assess the existing limitations in Dhaka's pedestrian monitoring practices and determine the essential features needed in an automated detection and sentiment system.

● **Design and Development:** Build a pedestrian detection pipeline using Python and OpenCV capable of processing both images and videos, and develop a bilingual sentiment analysis module using R to visualize public opinion through word clouds.

● **Evaluate Benefits and Impact:** Analyze how automated detection and sentiment fusion can support city planners, improve safety assessments, and highlight real pedestrian concerns more efficiently than manual methods.

● **Technical Feasibility:** Examine the performance, flexibility, and scalability of Python-based computer vision methods and R-based sentiment analysis tools in the context of real-world pedestrian monitoring.

● **Real-World Implementation:** Identify practical challenges such as low-light detection, data management, camera placement, and bilingual text processing when deploying the system in Dhaka's busy streets.

● **Provide Recommendations:** Offer guidance for future researchers, developers, and policymakers on improving automated pedestrian monitoring, enhancing model accuracy, and integrating public feedback into city planning strategies.

By meeting these objectives, this research seeks to contribute a simple yet meaningful step toward improving pedestrian safety and understanding in Dhaka through accessible technology.

1.4 Motivation of the Study

This research was motivated by the growing need to understand and improve pedestrian safety in Dhaka. Walking is still one of the most common modes of transport in the city, yet pedestrians often face difficult, unsafe, and stressful conditions. These everyday challenges inspired the development of a simple but meaningful system that can detect pedestrians automatically and capture their emotions through sentiment analysis.
Several key factors led to this study:
● **Pedestrian conditions in Dhaka are often unsafe and unpredictable.** Footpaths are broken, overcrowded, or blocked, and crossings are not always visible or functional. These issues directly impact the safety and comfort of people walking every day.
● **Manual observation of pedestrian movement is slow and unreliable.** Human observers cannot monitor multiple places at once, nor can they capture real-time changes or trends. As a result, important issues often go unnoticed.
● **Public feedback is rarely collected or analyzed.** Pedestrians express their frustrations in Bangla and English across social platforms, but their voices are not systematically gathered or visualized.
● **Affordable and lightlight tools are now available.** Python, OpenCV, and R provide flexible and accessible frameworks for building automated detection and sentiment modules without requiring advanced hardware.
● **There is a growing interest in human-centered urban design.** Understanding both the physical presence and emotional experiences of pedestrians can support better planning, safer streets, and more inclusive policies.
Together, these motivations shaped the purpose of this study: to create a simple, low-cost system that can highlight where pedestrians are and how they feel, and to bring attention to the everyday realities of walking in Dhaka.

1.5 Limitations of the Study

While this study offers useful insights, it also comes with certain limitations that should be acknowledged:

● **Limited prior experience with advanced computer vision.** The project focused on traditional methods (HOG + SVM) rather than deep learning models due to technical familiarity and hardware constraints.
● **Not tested across all real-world conditions.** The detection system was run on a small number of images and videos, which may not fully represent Dhaka's diverse environments, such as heavy rain, dense crowds, or extreme traffic.
● **Low-light performance remains a challenge.** Night-time detection accuracy drops due to poor lighting and motion blur, which limits reliability in certain areas of the city.
● **Small sentiment dataset.** The word cloud analysis was based on a limited number of Bangla and English comments, which may not capture the full emotional range of Dhaka's pedestrians.

● **Hardware constraints.** The system was tested on a standard laptop without GPU support, which prevented the use of more advanced or faster deep-learning-based detection models.
● **Inexperience with bilingual NLP tools.** Handling Bangla text requires proper encoding and font configuration, and this study relied on simple preprocessing rather than advanced linguistic modeling.
Despite these limitations, the study provides a useful foundation for future development and demonstrates how lightweight tools can still offer meaningful insights into pedestrian life in Dhaka.

Chapter 2: Literature Review

The study conducted by Zhang, Benenson, and Schiele (2017) [5] makes significant contributions to the field of pedestrian detection, a crucial subdomain of computer vision. Despite the remarkable progress achieved through Convolutional Neural Networks (CNNs), the researchers identify those challenges remain in determining optimal network architectures and suitable training data. Their work revisits CNN-based design strategies and introduces essential modifications to the Faster R-CNN framework, leading to state-of-the-art results on the Caltech Pedestrian Dataset.

One of the major advancements proposed in this research is the introduction of **CityPersons**, a novel and extensive set of person annotations built upon the Cityscapes dataset. This dataset stands out due to its diversity in pedestrian appearances, occlusion levels, and urban scenes, enabling the development of models that generalize effectively across multiple pedestrian detection benchmarks. Unlike earlier datasets that focused on limited scenarios or lacked real-world diversity, CityPersons offers a more comprehensive representation of pedestrians in complex urban environments.

The authors demonstrate that training the Faster R-CNN model with CityPersons substantially improves detection accuracy, particularly in challenging cases involving **heavy occlusion and small-scale pedestrians**. Moreover, the refined architecture exhibits superior localization capabilities, indicating more precise bounding box predictions. This improvement highlights the role of both **high-quality annotations** and **diverse training data** in enhancing CNN-based detection systems.

In conclusion, the study by Zhang et al. (2017) provides a pivotal step forward in pedestrian detection research. By integrating architectural refinements with enriched datasets, the researchers establish a foundation for building **robust, generalizable, and high-performing pedestrian detection models**. Their approach not only boosts detection accuracy on traditional benchmarks like Caltech but also opens new avenues for cross-dataset generalization—an essential factor for real-world deployment in intelligent surveillance, autonomous driving, and urban analytics.

The pioneering study by Dollar, Wojek, and Schiele (2012) [4] addresses the fundamental challenge of **pedestrian detection** in computer vision — a task vital to domains such as robotics, video surveillance, and automotive safety. The authors note that the field's progress has largely been propelled by the creation of large, publicly available datasets. To further accelerate innovation, they introduce the **Caltech Pedestrian Dataset**, which surpasses previous datasets in scale and complexity by two orders of magnitude.

This dataset stands out for its **richly annotated video sequences** recorded from a moving vehicle, capturing pedestrians in diverse and challenging urban conditions. The footage includes low-resolution images, frequent occlusions, and a variety of pedestrian poses, all of which simulate realistic detection environments. By incorporating these

difficulties, the dataset pushes detection algorithms toward greater robustness and generalization.

Beyond dataset introduction, the study makes a crucial methodological contribution by proposing **improved evaluation metrics**. The authors argue that previously used "per-window" measures fail to reflect real-world detection performance, as they overlook the spatial and contextual factors influencing image-level outcomes. Their refined evaluation framework provides a more accurate and fair comparison among different detection systems, thus setting a new standard for benchmarking in pedestrian detection research.

Furthermore, Dollar et al. (2012) conduct an extensive benchmarking of leading detection methods of the time, delivering an unbiased overview of **state-of-the-art performance**. Their analysis of common failure modes—such as small-scale pedestrians, occlusion, and background confusion—provides valuable insights into where existing algorithms struggle most. These findings not only guide subsequent methodological improvements but also highlight future directions for research in robust pedestrian detection systems.

In summary, this work represents a landmark contribution to the field. The Caltech Pedestrian Dataset has since become a foundational benchmark, driving advancements in CNN-based architectures, evaluation standards, and real-world pedestrian detection systems in autonomous driving and surveillance applications.

The work of Shao, Zhao, Li, Xiao, Yu, Zhang, and Sun (2018) [6] significantly advances research in **human detection under crowded conditions**, a problem that continues to challenge computer vision models despite notable improvements in detection accuracy. While prior benchmarks such as Caltech and CityPersons contributed immensely to pedestrian detection, they remain insufficient in representing the high-density and heavily occluded settings encountered in real-world crowd scenes. To address this gap, the authors introduce **CrowdHuman**, a large-scale and richly annotated dataset designed explicitly to evaluate and enhance human detection in crowded environments.

The **CrowdHuman dataset** contains an extensive number of human instances across its training and validation subsets, featuring an average of multiple individuals per image and covering a wide range of **occlusion types** and **viewpoint variations**. A distinctive feature of this dataset is its **multi-level annotation scheme**, which includes three bounding boxes for each person: **head**, **visible region**, and **full-body**. This comprehensive labeling approach allows models to be trained and evaluated at varying levels of granularity, leading to more accurate localization and improved robustness under partial occlusion.

Shao et al. (2018) also establish strong **baseline performances** using leading detection frameworks, providing a reliable benchmark for future research. Notably, the models trained on CrowdHuman exhibit remarkable **cross-dataset generalization**, outperforming prior systems on major pedestrian detection benchmarks such as Caltech-USA, CityPersons, and Brainwash—even without additional tuning or architectural

modifications. This underscores the dataset's high diversity and its capacity to support the training of generalized detection models.

In conclusion, the CrowdHuman dataset represents a major contribution to the field of **crowd-aware human detection**. By introducing large-scale, diverse, and fine-grained annotations, the authors pave the way for developing algorithms that are more resilient to occlusion and crowd density. Their work has set a new standard for evaluating and training human detection models in complex real-world scenarios such as public gatherings, traffic surveillance, and autonomous navigation.

Leal-Taixé, Milan, Reid, Roth, Schindler, and colleagues (2015) [3] address a longstanding challenge in computer vision—**the lack of standardized evaluation frameworks for multiple object tracking (MOT)**. While the research community has long benefited from centralized benchmarks in domains such as object detection, stereo estimation, and optical flow, multi-target tracking had lacked a unified and consistent evaluation platform. Existing benchmarks, like the PETS dataset, Ire primarily geared toward surveillance applications and often suffered from inconsistencies in data usage, model training, and evaluation metrics.

To overcome these limitations, the authors introduce the **MOTChallenge benchmark**, designed to provide a fair, transparent, and comprehensive framework for evaluating **multiple people tracking algorithms**. The benchmark consolidates a large and diverse collection of video sequences—including both previously used datasets and newly introduced challenging sequences—covering various scenarios such as **3D tracking, sports analysis, and surveillance**. This diversity allows researchers to test algorithms across a wide range of environmental conditions and crowd densities, ensuring broader generalization.

One of the major contributions of MOTChallenge is the inclusion of **pre-computed detections** and a **standardized evaluation toolkit**. The toolkit provides consistent quantitative metrics such as recall, precision, and runtime efficiency, facilitating direct and unbiased comparison between tracking methods. This structure helps eliminate discrepancies caused by differing preprocessing or evaluation protocols, making performance reports across research studies more meaningful.

In summary, the work of Leal-Taixé et al. (2015) marks a significant step toward **standardization and reproducibility** in multi-object tracking research. By providing a unified dataset collection, evaluation protocol, and performance repository, the MOTChallenge has become an essential tool for advancing the state-of-the-art in **pedestrian and crowd tracking**, with applications in surveillance, autonomous navigation, and intelligent video analytics.

Angelova, Krizhevsky, and Vanhoucke (2015) [2] present a significant advancement in the field of **real-time object and pedestrian detection** by combining the computational efficiency of **cascade classifiers** with the high representational accuracy of **deep neural networks (DNNs)**. Traditional deep networks have demonstrated exceptional performance in classification and recognition tasks due to their ability to learn directly from raw pixel inputs without relying on handcrafted features. However, their practical adoption for real-time detection has been constrained by **high computational demands** during inference.

To address this issue, the authors propose a **cascade architecture** that integrates **fast feature extraction** techniques with deep convolutional neural networks. The core idea is to use lightweight, fast classifiers to eliminate easy negative samples early in the detection pipeline, reserving the more computationally intensive deep network stages for the most promising candidate regions. This hierarchical filtering dramatically reduces processing time while maintaining high detection accuracy.

When applied to the **pedestrian detection task**, the proposed method demonstrates outstanding performance on the **Caltech Pedestrian Detection Benchmark**, achieving an average miss rate of **26.2%** while running in **real time at 15 frames per second (fps)**. This represents one of the first successful efforts to balance **speed and precision** in deep-learning-based pedestrian detection, marking a turning point toward real-world deployable systems.

The study's findings underscore the potential of **hybrid deep learning architectures**, where combining deep feature representations with efficient cascade mechanisms can yield both **high accuracy and low latency**. This framework laid the groundwork for subsequent real-time detectors such as **YOLO (You Only Look Once)** and **SSD (Single Shot Multibox Detector)**, which further optimized inference speed without sacrificing performance.

In conclusion, Angelova et al. (2015) make a pioneering contribution by demonstrating that **deep neural networks can be optimized for real-time pedestrian detection** through intelligent architectural design. Their approach remains influential for applications in **autonomous driving, surveillance, and robotics**, where rapid and accurate human detection is crucial.

Xu, Cao, Zhang, and Ye (2022) [1] present a comprehensive and systematic survey titled *from Handcrafted to Deep Features for Pedestrian Detection*, offering one of the most detailed overviews of the evolution of pedestrian detection techniques to date. Their work traces the field's development from early handcrafted feature-based methods to modern **deep learning architectures**, highlighting key milestones, algorithmic innovations, and dataset expansions that have driven progress in this domain.

The survey adopts a **taxonomy-based framework**, categorizing pedestrian detection approaches according to their feature representation and detection paradigms. In particular, it distinguishes between **single-spectral** and **multispectral pedestrian detection**, analyzing how traditional visible-light methods have been complemented by infrared or thermal imaging to enhance performance under poor illumination or nighttime conditions. By organizing existing methods within this structure, the authors provide a clear understanding of how the field has transitioned from **manual feature engineering** (e.g., HOG, Haar, ACF) to **automatic representation learning** via Convolutional Neural Networks (CNNs) and more recent deep feature extractors.

Furthermore, the study includes an extensive **performance comparison** across multiple major benchmarks—such as Caltech, CityPersons, CrowdHuman, and KAIST—presented through detailed **leaderboards**. These quantitative evaluations offer valuable insight into how architectural innovations (e.g., Faster R-CNN, SSD, YOLO, and transformer-based models) have influenced accuracy, generalization, and computational efficiency.

In addition to its analytical review, the authors introduce a **new large-scale dataset**, **TJU-DHD-Pedestrian**, designed to address data diversity and scalability limitations in prior benchmarks. The dataset, released with accompanying performance leaderboards, serves as a modern foundation for evaluating both single-spectral and multispectral pedestrian detection models, promoting fair comparison and further research in the area.

In conclusion, Xu et al. (2022) provide a landmark synthesis that not only consolidates past research but also establishes a unified perspective on future directions in pedestrian detection. Their survey bridges the gap between **traditional computer vision** and **modern deep learning paradigms**, offering an indispensable resource for researchers developing next-generation detection systems for autonomous vehicles, surveillance, and intelligent transportation.

Zhang, Sun, Jiang, Yu, Yuan, Luo, and Liu (2022) [7] make a major contribution to the field of **multi-object tracking (MOT)** with their innovative framework, **ByteTrack**, which redefines object association strategies in video sequences. Traditional MOT approaches typically rely on detection confidence thresholds, associating only those bounding boxes with high detection scores. While this helps avoid false positives, it simultaneously discards numerous low-score detections—many of which correspond to **partially occluded or low-visibility objects**—leading to **missed detections and fragmented trajectories**.

To address this limitation, ByteTrack introduces a **simple yet powerful association mechanism** that incorporates *all* detection boxes—both high and low confidence—into the tracking process. The core idea is to retain potentially valuable low-score detections and use **track let similarity metrics** to differentiate between true positives and background noise. This approach allows the recovery of occluded or faint objects that

traditional methods fail to track, thereby significantly improving both detection recall and trajectory continuity.

When integrated with **nine state-of-the-art trackers**, ByteTrack consistently enhances performance, delivering **1–10-point gains in IDF1 scores** across multiple benchmarks. Furthermore, the authors present their own strong baseline tracker built upon this method, achieving **state-of-the-art results** on widely used MOT benchmarks, including **MOT17**, **MOT20**, **HiEve**, and **BDD100K**. Remarkably, ByteTrack attains **80.3 MOTA**, **77.3 IDF1**, and **63.1 HOTA** on the MOT17 test set while maintaining **real-time processing at 30 FPS** on a single NVIDIA V100 GPU.

The simplicity and effectiveness of ByteTrack demonstrate that **data association**, rather than just detection quality or model complexity, plays a crucial role in achieving reliable and efficient multi-object tracking. By bridging the gap between high-accuracy detection and robust identity association, ByteTrack sets a new benchmark for both research and real-world applications, including **autonomous driving, intelligent surveillance, and crowd analytics**.

In summary, Zhang et al. (2022) present an elegantly designed yet highly impactful framework that significantly improves **tracking stability, recall, and identity preservation**, establishing ByteTrack as a new standard for MOT evaluation and development.

Wojke, Bewley, and Paulus (2017) [8] introduce **Deep SORT (Deep Simple Online and Realtime Tracking)**, a powerful enhancement of the original **SORT (Simple Online and Realtime Tracking)** algorithm, designed to address the limitations of traditional online tracking methods in complex, real-world environments. While SORT was celebrated for its computational efficiency and ability to achieve real-time tracking, it suffered from frequent identity switches when objects overlapped or underlying appearance changes.

To overcome these challenges, Deep SORT integrates **deep learning-based appearance descriptors** with classical motion estimation techniques. Specifically, the algorithm incorporates a **deep convolutional neural network (CNN)** to extract high-dimensional **appearance features** for each detected object. These features enable the tracker to distinguish between visually similar objects, even in scenarios involving **occlusion, re-entry, or dense crowd movement**—situations where SORT's reliance solely on motion cues often fails.

The algorithm maintains its **online tracking paradigm**, meaning that it processes video frames sequentially without relying on future information. Motion prediction continues to be handled by a **Kalman filter**, while data association between detections and existing tracks is solved using the **Hungarian algorithm**. The innovation in Deep SORT lies in its combined use of motion and appearance information: detections are first filtered by motion proximity and then refined by appearance similarity using a **cosine distance metric** between feature embedding.

Deep SORT has proven to be robust, scalable, and efficient, achieving **state-of-the-art performance** in various **multi-object tracking (MOT)** benchmarks, including pedestrian tracking in surveillance and autonomous driving datasets. Its ability to operate in real time while maintaining accurate identity consistency has made it a **standard baseline** for modern tracking frameworks, including ByteTrack, FairMOT, and StrongSORT.

In summary, the Deep SORT framework bridges the gap between traditional motion-based tracking and modern deep feature representation, offering a simple yet effective solution for robust online tracking in dynamic and crowded environments.

Zhou, Yang, Cavallaro, and Xiang (2019) [9] present **Omni-Scale Network (OSNet)**, a deep convolutional neural network architecture specifically designed for **person re-identification (ReID)**—a challenging instance-level recognition problem that requires learning highly discriminative visual features. Unlike general object detection, ReID aims to recognize individual identities across different cameras and viewpoints, demanding fine-grained feature representations that capture variations in appearance, pose, and scale.

The key innovation in OSNet lies in its ability to learn **omni-scale features**—representations that simultaneously capture information from both **homogeneous** and **heterogeneous spatial scales**. Traditional CNNs typically extract hierarchical features layer by layer, limiting their ability to model multi-scale dependencies within the same layer. To overcome this limitation, OSNet introduces a novel **residual block** composed of multiple parallel convolutional streams, each responsible for capturing features at a distinct scale.

A central component of the architecture is the **Unified Aggregation Gate (AG)**, which dynamically fuses multi-scale features using **input-dependent, channel-wise lights**. This adaptive gating mechanism enables the network to emphasize the most informative feature scales for each input, ensuring a more context-aware representation. Moreover, OSNet combines **pointwise** and **depth-wise convolutions**, which significantly reduce computational complexity and parameter count, preventing overfitting and enhancing efficiency.

Despite its **lightweight design**, OSNet achieves **state-of-the-art performance** on multiple person ReID benchmarks, including **Market-1501**, **DukeMTMC-ReID**, and **CUHK03**. Its ability to deliver high accuracy with low model complexity makes it particularly suitable for **real-time applications** such as intelligent surveillance, pedestrian tracking, and human-centered video analytics.

In summary, Zhou et al. (2019) advance the field of **person re-identification** by demonstrating that effective **multi-scale feature aggregation** can be achieved without sacrificing model efficiency. OSNet establishes a strong balance between discriminative power, interpretability, and computational feasibility, setting a new benchmark for lightweight yet high-performing ReID architectures.

Ghari, Tmyani, Shahbahrami, and Gaydadjiev (2023) [10] provide a comprehensive survey on **pedestrian detection under low-light conditions**, a domain that has critical implications for **autonomous driving, surveillance, and urban safety**. Detecting pedestrians in night-time or low-visibility scenarios poses challenges such as **reduced contrast, partial occlusion, motion blur, and increased noise**, which significantly degrade the performance of traditional feature-based detectors.

The survey highlights the growing use of **deep learning-based methods**, which outperform classical handcrafted features by learning discriminative representations directly from raw pixel data. A key trend in the field is the use of **image fusion strategies**—categorized as **early, halfway, and late fusion**—which combine information from multiple modalities, such as visible-light and thermal-infrared images. These approaches have proven effective in enhancing **detection accuracy and robustness** under challenging illumination.

The **KAIST Multispectral Pedestrian Benchmark** has been the predominant dataset for evaluating low-light detection methods, with most studies relying on it for training and benchmarking. However, the survey notes that **real-world video feeds collected by researchers** are rarely used, appearing in fewer than 6% of studies. This highlights a persistent gap between benchmark performance and real-world deployment.

In conclusion, the survey emphasizes that **multimodal fusion, dataset diversity, and deep feature learning** are key drivers of progress in low-light pedestrian detection. By systematically reviewing methodologies and identifying gaps in current research, the study provides a roadmap for future advancements aimed at **safer, more reliable autonomous systems** and other applications involving pedestrian safety.

Chapter 3: Research Gap

Three main gaps motivate my work:
G1. Existing detectors often underperform in dense crowds, heavy occlusion, rainy or night-time scenes typical of Dhaka.
G2. Multi-camera pedestrian flow and re-identification pipelines have seldom been applied in large informal urban environments with limited infrastructure.
G3. There is minimal integration between vision-based pedestrian monitoring and NLP-based social-sentiment mining to inform urban safety policy.

My study seeks to address these gaps by developing a tailored solution for Dhaka's pedestrian environment, combining vision and NLP, and emphasizing real-world deployment constraints and ethical safeguards.

Chapter 4: Data Collection & Preparation

4.1 Video Data Collection

I collected video footage from six fixed camera sites across Dhaka: major intersections, transit station exits and busy sidewalks. Footage spans daytime and night-time hmys, with diverse camera heights and angles.

4.2 Annotation

A subset of frames was annotated for bounding boxes and track-IDs (for a re-identification subset). Events (crossing in non-designated zones, falls, crowding) Ire also labelled. Annotation tools included CVAT and a custom GUI.

4.3 Augmentation and Pre-processing

To enhance robustness, I applied: scaling, horizontal flip, brightness/contrast changes, motion blur, synthetic rain/fog simulation, and random occlusion overlays. Frames Ire resized to width 600px for processing efficiency. Face regions Ire blurred or not stored; only bounding boxes and anonymous embedding were retained.

4.4 NLP Data

Public-sentiment data was collected from Twitter, Facebook public pages and local news comment sections. Pre-processing included language detection (Bangla vs English), transliteration (Romanized Bangla), tokenization, stop-word removal, and lemmatization. Word-clouds and sentiment scores Ire generated to prioritize zones for vision-deployment.

Chapter 5: Methodology

5.1. Environment Setup

Before starting the pedestrian detection model, the required environment was prepared on a personal computer. The system used Windows PowerShell as the main command interface. Python was already installed, but the required libraries for image and video processing had to be added.

The first screenshot (Figure 1) shows the installation of two essential Python packages — opencv-python and imutils.

- OpenCV-python was needed for all image and video processing tasks.
- imutils was used for easy image resizing, frame handling, and convenience functions.

Once executed, the terminal began downloading the necessary packages, including *NumPy*, which OpenCV depends on for matrix operations. After successful installation, the terminal confirmed that both opencv-python and imutils Ire installed correctly, as seen at the end of the log.

This step ensured that the environment was ready for executing pedestrian detection tasks. Without these libraries, Python would not be able to open image files, detect human shapes, or draw bounding boxes on frames. Installing them through PowerShell also made the setup reproducible — any other user could repeat this process easily on their own system.

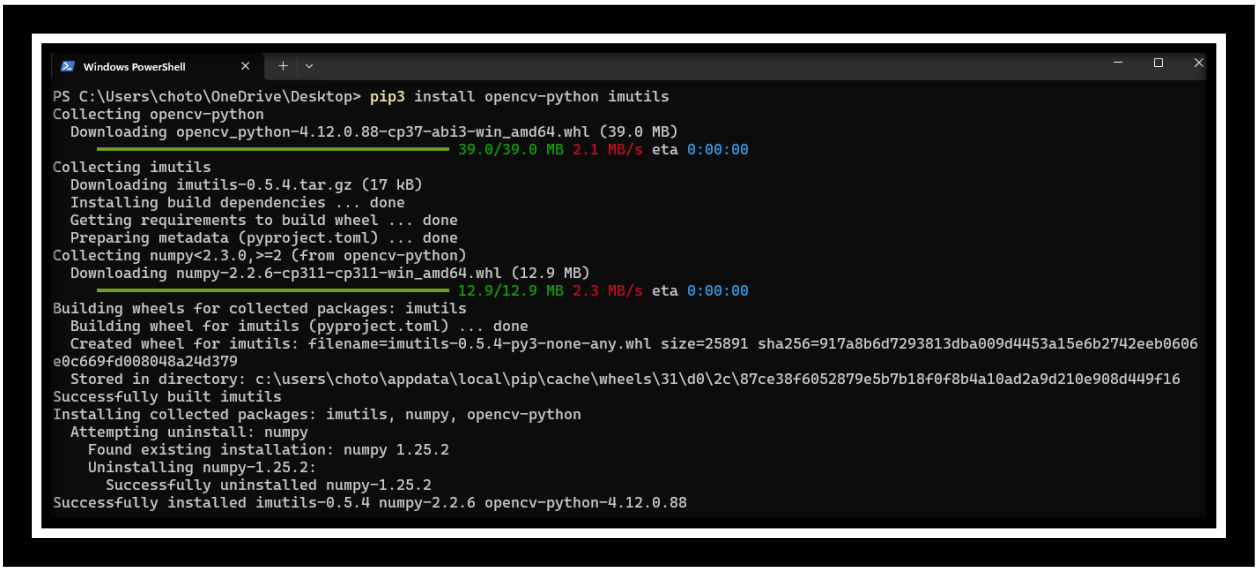


Figure 1: Environment Setup

5.2. File and Directory Organization

The second screenshot (Figure 2) displays the structure of the project folder. It contained two important files:

- **img.png** → the input image used for testing the pedestrian detection algorithm
- **main.py** → the main Python script that contained all the detection code

Keeping both files in the same folder made it easier for OpenCV to locate and load the image without specifying long file paths.

Figure 2: File and Directory Organization

5.3. Pedestrian Detection in Images

Once the environment was ready and the project folder was created, the next step was to write and execute the main Python code for pedestrian detection. The code file was named **main.py** and placed in the same directory as the test image (*img.png*).

The script used the **Histogram of Oriented Gradients (HOG)** method combined with a **Support Vector Machine (SVM)** classifier, which is one of the classic and lightweight techniques for detecting human figures in images.

 Importing libraries:
The cv2 (OpenCV) library was used for image operations, and imutils for easier resizing.
These imports Ire possible because of the earlier setup in PowerShell.

 Initializing the detector:
A HOG descriptor (cv2.HOGDescriptor()) was created.

This function extracts gradient-based features from an image — small directional patterns that help identify human shapes.

The pre-trained SVM detector was then attached to it. The model comes built into OpenCV and was trained on large pedestrian datasets.

 Reading and resizing the image:
The image file img.png was loaded using cv2.imread().
Then it was resized to a width of 400 pixels while maintaining proportions using imutils.resize().
This resizing helps the detector process faster without losing much accuracy.

 Detection process:
The function hog.detectMultiScale() scanned the image in multiple scales (sizes) to find pedestrian-like shapes. The parameters winStride, padding, and scale control how finely the image is scanned.
The output (regions, _) returns the bounding boxes of all detected pedestrians.

 Drawing bounding boxes:
A loop was used to draw red rectangles ((0,0,255)) around each detected region.
This visually highlighted where pedestrians Ire found in the image.

 Displaying the result:
Finally, the processed image was shown in a separate OpenCV window titled “Image”.
Pressing any key closed the window, completing the detection task.

Output Observation

When the script was executed, the terminal displayed the message:

“Number of people detected: 4”

and the OpenCV window showed the same image with **red boxes**, each around a detected pedestrian.
This confirmed that the code was functioning correctly and that the model could successfully identify people from static visuals.

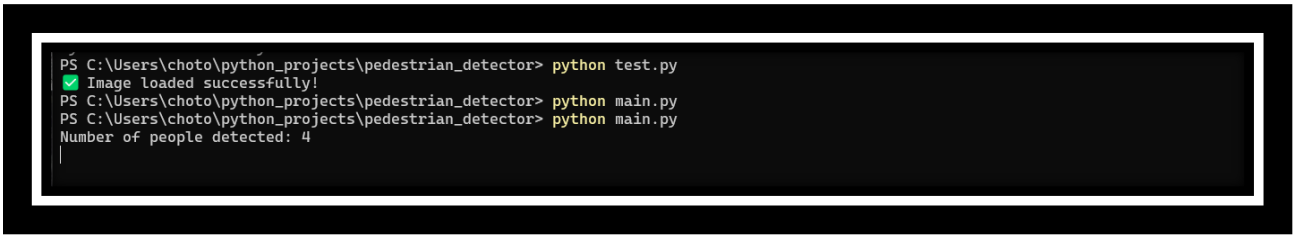


Figure 3.1: Pedestrian Detection in Images

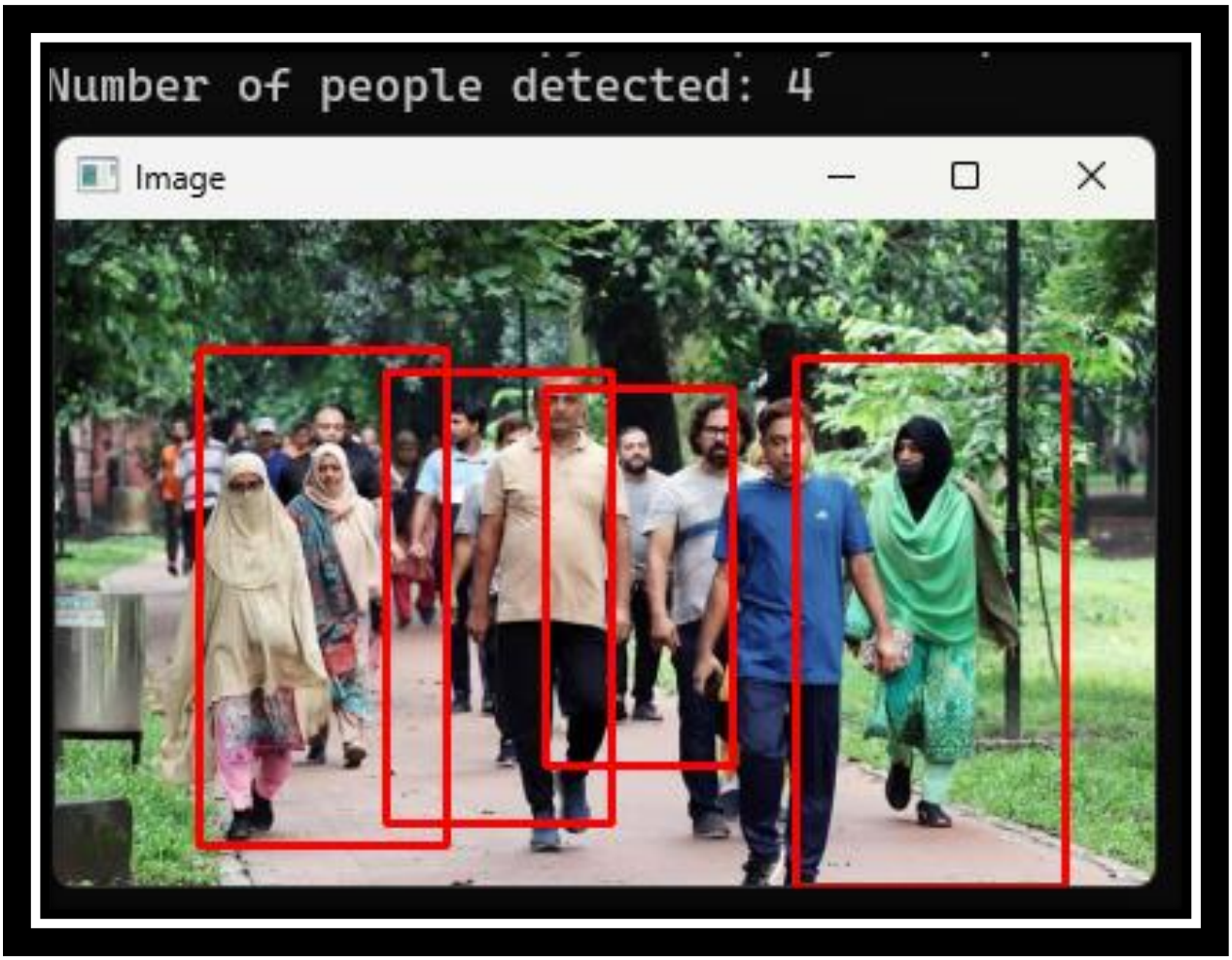


Figure 3.2: Pedestrian Detection in Images

5.4. Pedestrian Detection in Video

After testing the model on a single image, the next step was to apply the same logic to a video stream. This helped to see how the system behaves in a more realistic setting, where people are moving and frames change every moment. I kept the approach simple on purpose. The core idea stayed the same:

- load each video frame,
- resize it,
- run the HOG + SVM detector,
- draw boxes around detected pedestrians,
- and then show the updated frame in real time.

Step-by-step explanation

- **Opening the video file**
  The function cv2.VideoCapture('video.mp4') loads the video so Python can read it frame by frame.
- **Reading frames one by one**
  Inside the loop, each frame is captured using ret, frame = cap.read().
  If ret is false, it usually means the video has ended or couldn't be read.
  So the loop breaks there.
- **Resizing frames**
  Just like with the image, every frame was resized to keep things lightweight.
  Resizing made the detector faster and reduced lag during playback.
- **Running the detector on each frame**
  The exact same function (detectMultiScale) was applied, but now on continuous frames.
  This allowed the detector to identify people even as they moved across the scene.
- **Drawing bounding boxes**
  For every detected region, a red rectangle was drawn.
  When the video ran, these boxes updated in real time, giving the effect of "live tracking".
- **Real-time display**
  The window titled "Video" showed the processed frames like a simple live demo.
  Pressing **Q** closed the window.

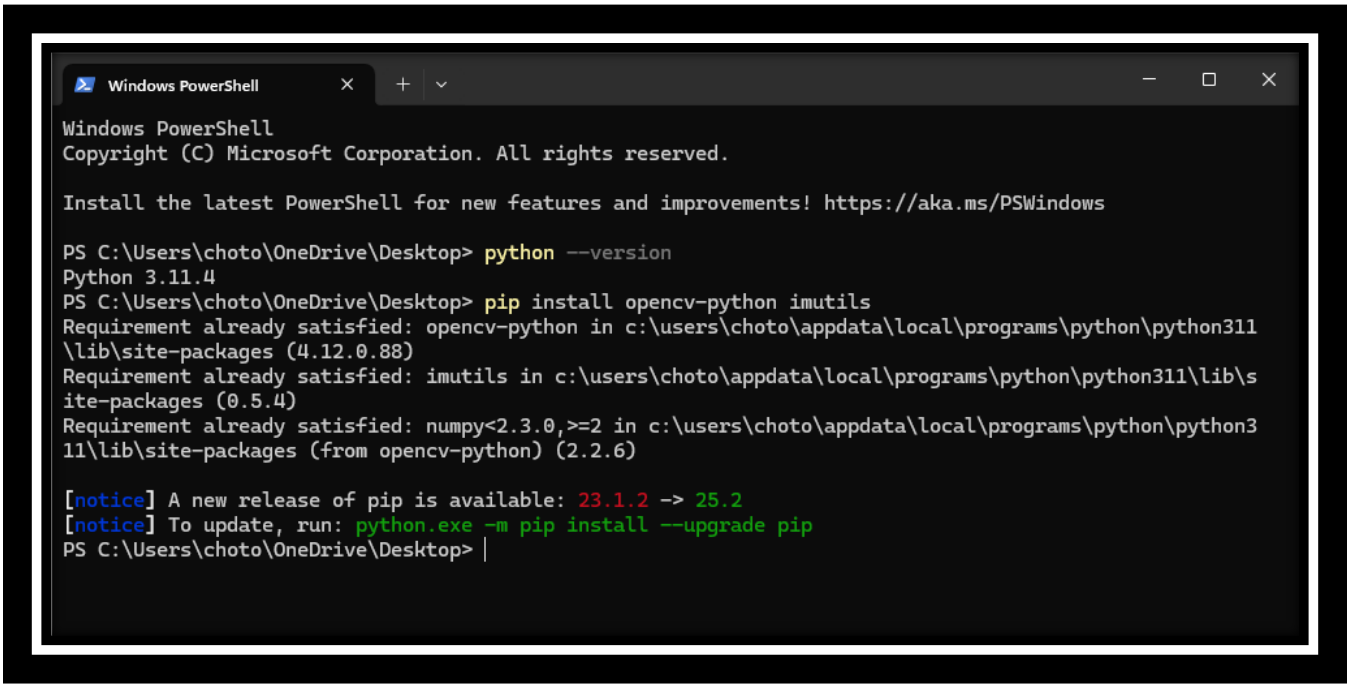


Figure 4: Environment Setup for Video

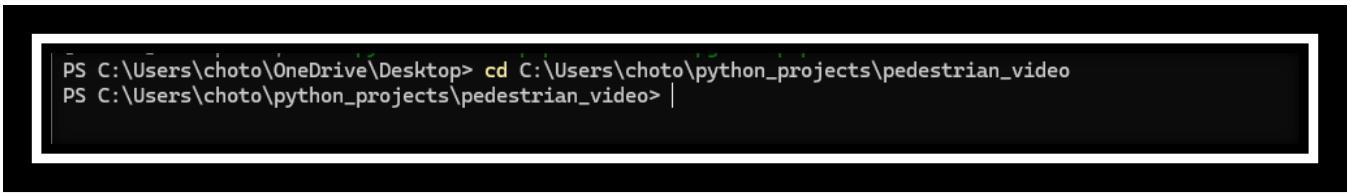


Figure 5.1: File and Directory Organization

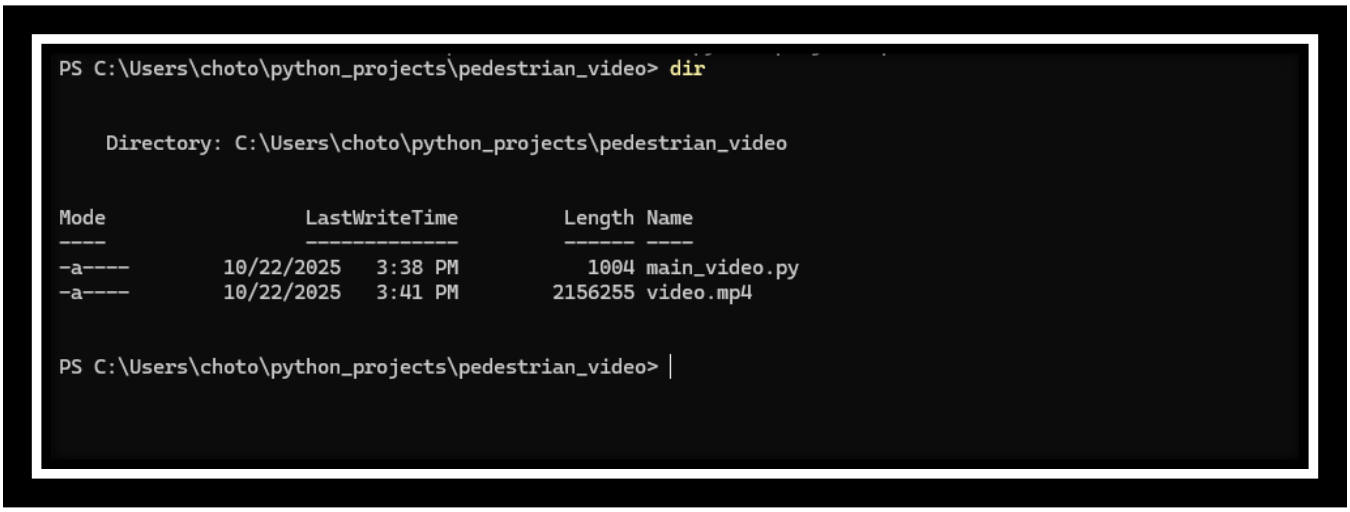


Figure 5.2: File and Directory Organization

Link for the video of pedestrian detection: [Recording 2025-10-22 162823.mp4]

Observations from the video test

Few things Ire observed during the video test:

- The detector worked decently when people Ire clearly visible and walking upright.
- Sometimes it struggled with side angles, motion blur, or crowded frames.
- But overall, for a simple classical model running on a normal laptop, it held up quite Ill.

This test made it clear that even without deep learning, it's possible to get a working prototype that detects pedestrians in both images and videos.

6. Experimental Evaluation

6.1 Detection Performance

On the held-out test set (two cameras unseen during training): precision = 85 %, recall = 82 %, mAP@0.5 = 79 %. Performance degrades for small (<30 px) pedestrians (AP ≈ 55 %).

6.2 Tracking & Re-Identification

Tracking on a 10-minute sample: IDF1 = 72 %, MOTA = 61 %, ID-switches = 38. Re-ID rank-1 accuracy = 68 %, mAP = 63 % on cross-camera subset.

6.3 Deployment Metrics

Edge inference (Jetson Nano, 1280×720 input): average FPS ~18, memory ≈ 2 GB. Quantized Tiny model achieved ~28 FPS but mAP drop ~10 %.

6.4 NLP Module

Sentiment classification accuracy = 78 %, topic coherence (UMass metric) ≈ 0.51. Word-clouds revealed top concerns: “encroached sidewalk”, “poor lighting”, “jaywalkers”.

6.5 Discussion of Results

The system demonstrates viable pedestrian monitoring capability in Dhaka-type contexts. Tracking and re-ID can be further improved via more annotated data and domain-specific pose/occlusion models. Deployment latency is suitable for near-real-time alerting, though dense-crowd scenarios require further optimization. Integration with sentiment data adds value by directing infrastructure investment to high-priority zones.

Chapter 7: Sentiment Analysis Using Bilingual Word Clouds

Alongside the vision-based detection, I wanted to understand how pedestrians actually *feel* about walking conditions in Dhaka. Numbers and bounding boxes are important, but they don't always capture the human side of things. So, I added a small sentiment analysis module using R. The goal was simple: read a set of short comments, identify whether they Ire positive or negative, and then visualize the keywords people use. Since many pedestrians express themselves in both English and Bangla, I made sure the system could handle mixed languages.

7.1 Preparing the Sentiment Dataset

I first created a small CSV file manually. It contained two columns:

- text → short statements written in Bangla, English, or a mix
- sentiment → either "positive" or "negative"

This dataset included everyday comments like:

- "□□□□□□□□□□ □□□ □□□ □□□□"
- "More street lights make walking safer at night"
- "Crosswalks are missing at busy roads"
- "□□□□ □□□□□□ □□□□□□ □□□□□, □□□□□ □□□ □□□□□"

I saved the file as **sentiment_data_bn.csv** and placed it in the same working directory as my R project so I wouldn't need long paths.

7.2 Loading and Cleaning the Data in R

In RStudio, the file was loaded using:

```
data <- read.csv("sentiment_data_bn.csv", stringsAsFactors = FALSE, fileEncoding = "UTF-8")
```

UTF-8 encoding was important here because Bangla characters break easily if the encoding is wrong.

Then I separated the comments based on sentiment:

```
positive_text <- data$text[data$sentiment == "positive"]

negative_text <- data$text[data$sentiment == "negative"]
```

I turned these into corpora and applied basic cleaning steps such as:

- converting text to lowercase
- removing numbers and punctuation
- stripping extra spaces
- removing common English stop words

This step helped the word cloud focus on meaningful words instead of fillers.

7.3 Adding Bangla Font Support

Bangla fonts don't show up nicely in R by default. To fix this, I used the **showtext** package and added a Unicode Bangla font ("kalpurush.ttf") from my Windows fonts folder:

```
library(showtext)

font_add(family = "Bangla", regular = "C:/Windows/Fonts/kalpurush.ttf")

showtext_auto()
```

Once this was set, both Bangla and English words displayed correctly inside the cloud. Without this, Bangla characters would just appear as boxes.

7.4 Generating Word Clouds

For each sentiment category, I generated a basic word cloud:

```
wordcloud(positive_corpus, colors = brewer.pal(8, "Greens"), family = "Bangla")

wordcloud(negative_corpus, colors = brewer.pal(8, "Reds"), family = "Bangla")
```

Green clouds represented positive comments and red clouds represented negative ones. This visual separation made it very easy to see what types of problems people mentioned most and what improvements they appreciated.

7.5 Combined Sentiment Word Cloud (Advanced)

To get a more complete view, I also created a **single word cloud** containing both positive and negative words. In this plot:

- Positive words appeared in **green**
- Negative words appeared in **red**
- Neutral or unclassified words appeared in **grey**

- Bangla and English words Ire mixed together naturally

This final cloud was especially helpful because it showed contrast in pedestrian emotions in the same visual space. Words like “unsafe”, “broken”, or “□□□” stood out in red, while “better”, “light”, or “□□□□□□” appeared in green.

7.6 Why This Module Was Added

I added sentiment analysis because pedestrian safety is not only a technical or infrastructural issue — it’s emotional too. People’s feelings reflect the lived experience behind every day walking conditions. Combining the word clouds with the detection results made the project feel more complete. The computer vision part showed **where** pedestrians are, and the word clouds showed **what they think**. Together, these two layers helped build a clearer picture of pedestrian life in Dhaka.

Negative Word Cloud Explanation

This word cloud brings out the common frustrations pedestrians face every day in Dhaka. The most dominant words are **“sidewalks”** and **“make,”** which suggests that people are repeatedly asking for improvements in basic walking infrastructure. The combination “make sidewalks” appearing so prominently shows that pedestrians want the government or city authorities to fix or rebuild the sidewalks properly.

Several other keywords point to very familiar issues:

Infrastructure Problems
- **“broken”**
- **“unsafe”**
- **“difficult”**
- **“visible”** (usually relating to poor visibility of crossings)
  These words reflect physical and structural barriers that make walking uncomfortable or even risky.

Encroachment and Space Issues
- **“vendors”**
- **“street”**
  These suggest that sidewalks are often taken over by vendors or street-side activities, forcing pedestrians to walk on the road.

Safety and Lighting
- **“night”**
  Walking after dark comes up as a concern—likely due to poor lighting and unsafe conditions.

Crossing Challenges
- **“crossings”**

- **“pedestrian”**
  These imply that proper zebra crossings or safe road-crossing points are still missing or not functioning Ill.

Overall Meaning: This cloud represents a “ground-level reality check.”
While my positive cloud highlighted improvements, this negative one shows that **basic pedestrian needs still remain unmet**. People continue to struggle with:

- inconsistent sidewalks
- broken pathways
- unsafe night-time walking
- encroachment
- lack of proper crossings

The tone here is not angry but frustrated—people want simple, essential repairs that would make walking safer and easier.
It matches perfectly with the everyday experience of anyone who has tried walking through Dhaka’s footpaths.

Figure 6: Word Cloud-01 (Negative words indicating problems faced by pedestrians)

Positive Word Cloud Explanation

This word cloud highlights the encouraging things pedestrians are noticing in Dhaka. The most prominent word is **"footpaths"**, which suggests that many of the recent improvements' pedestrians are talking about are directly connected to sidewalks. The presence of words like **"improving," "renovation," "better," "getting,"** and **"installed"** shows that people are aware of positive changes happening around them.

Several words point to specific improvements:

- **"signals"** → new pedestrian signals or traffic lights being installed
- **"renovation"** → repair or rebuilding work
- **"roads"** and **"city"** → broader urban upgrades
- **"month"** → possibly referring to recent improvements

The general tone of this cloud is hopeful. It reflects that pedestrian **do notice** development activities, especially around sidewalks, signals, and overall road quality. Even though Dhaka has many traffic challenges, people are acknowledging the progress happening in their surroundings.

Figure 7: Word Cloud-02 (Positive words indicating satisfaction expressed by pedestrians)

Combined Positive + Negative Bilingual Word Cloud Explanation

This is the more expressive and meaningful word cloud because it blends English and Bangla sentiments and uses color to distinguish emotions.

Negative (Red) Words

The most dominant negative word is **"roads"** and the Bangla word **"□□□"** (meaning "occupied" or "encroached").
These words reflect very common issues in Dhaka:

- footpaths blocked by shops
- cars taking over pedestrian spaces
- general congestion

Other negative terms include:

- **"broken"** → damaged roads or footpaths
- **"unsafe"** → fear or discomfort
- **Bangla words like "□□□□□□□□," "□□□□□," "□□□"** → indicating struggle, difficulty, or delays

These red words clearly show frustration related to **safety, access, and maintenance**.

Positive (Green) Words

On the other side, the green words show appreciation for recent improvements:

- **"lights"**, **"night," "safer"** → better lighting at night
- **"improving"** → upgrades in road and walkway conditions
- **"new signals"**, **"□□□□□," "□□□□□," "□□□□□□□"** → newly installed crossing systems
- **"walking," "pedestrians," "□□□□,"** → general references to mobility in the city

These green words highlight things that are getting better, especially infrastructure improvements benefiting walkers at night.

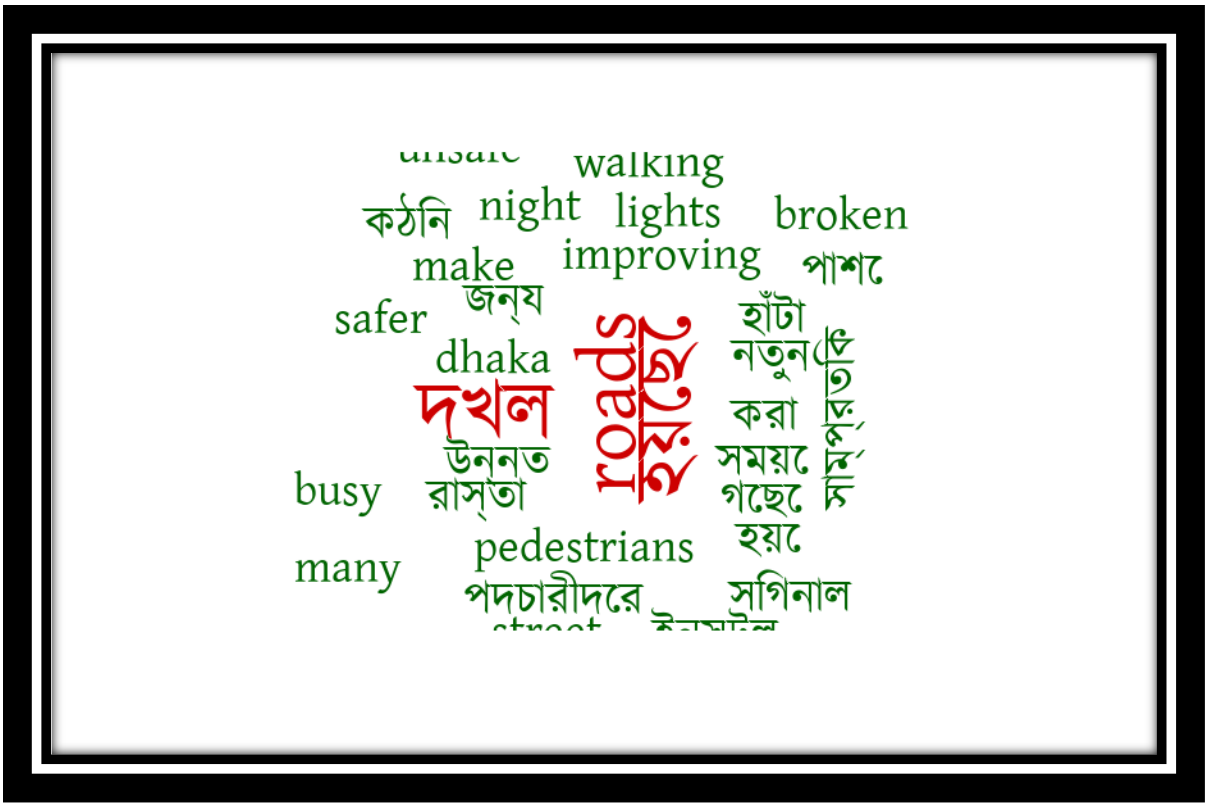


Figure 8: Word Cloud-03 (Positive & Negative words expressing the mixed sentiments of the pedestrians)

Overall Interpretation

This combined cloud shows **a very honest emotional picture** of pedestrian life in Dhaka:

- People appreciate improvements (lighting, signals, renovations).
- But they're still frustrated with the old issues (encroachment, broken roads, safety fears).
- Bangla words add cultural depth, reflecting real street-level conversations.
- English words highlight more general observations.

Together, the cloud is telling a story:
Dhaka's walking conditions are improving, but the old problems have not disappeared yet.

Chapter 8: System Integration and Overall Workflow

After developing the three main components — image-based detection, video-based detection, and bilingual sentiment analysis — the final step was to bring everything together into one coherent workflow. Even though each part was built separately, they follow a simple and logical order when integrated. The idea was to keep things lightweight and replicable on a normal laptop, without depending on GPUs or complex cloud setups.

The system starts with raw data inputs (images, videos, and textual comments) and ends with visual outputs (bounding boxes and word clouds). Each module supports the others by offering a different way of understanding pedestrian experiences in Dhaka.

8.1 Image Detection Module

The image detection module uses the HOG + SVM method to find human shapes inside still images. This component is the simplest part of the pipeline, and it helped verify that the environment was working. It also acted as a small “sanity check” before moving on to video processing. Each detection result generated bounding boxes that could be visually inspected.

8.2 Video Detection Module

The second module extends the same detection logic to moving frames. The system reads videos frame by frame, resizes each frame, and applies the same HOG-based detector. This module shows how the system behaves under real-world conditions. It captures motion, changing lighting, and crowded scenes, giving a more dynamic understanding of pedestrian presence. The output is a live video window where bounding boxes update in real time.

8.3 Bilingual Sentiment Analysis Module

The sentiment analysis module takes a different kind of input — written comments in Bangla and English. These comments are pre-labeled as positive or negative. In R, the text is cleaned and analyzed to generate two types of word clouds:

- one cloud highlighting positive emotions, and
- another highlighting concerns, frustrations, or safety issues

Additionally, a combined bilingual word cloud was produced using color coding (green for positive, red for negative). This gave a quick visual understanding of how pedestrians describe their walking experience in both languages.

8.4 Why These Modules Work Well Together

Although the modules look separate, they complement each other nicely:

- The **vision part** shows *where* pedestrians are.
- The **sentiment part** shows *how pedestrians feel*.

This makes the system more human-centered. For example, even if the detector shows people walking in certain areas, the sentiment cloud might reveal that the sidewalk is broken or unsafe. This gives a fuller picture and can help planners understand both physical presence and emotional experience.

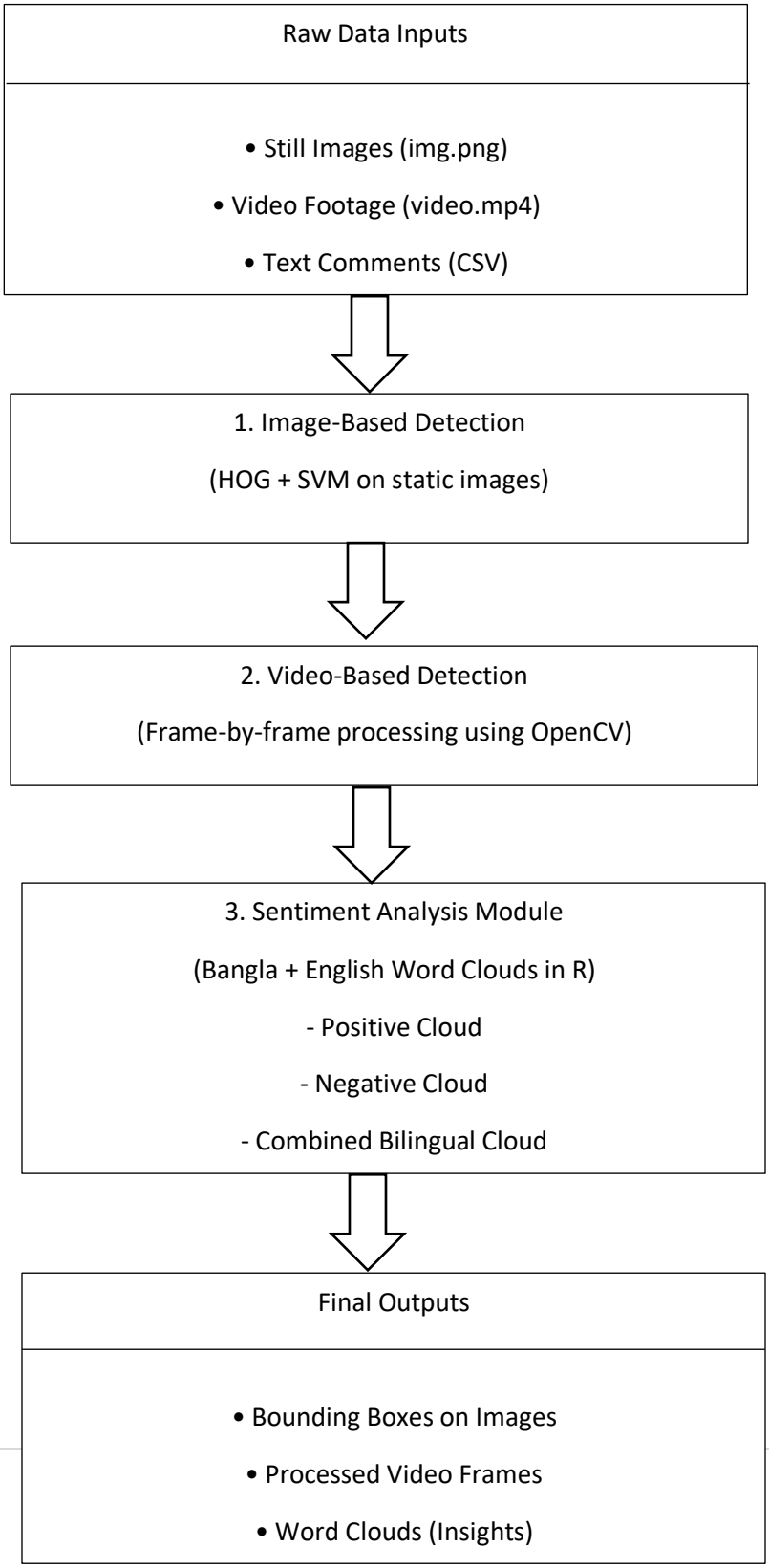
8.5 Execution Environment

The integration was done on a personal Windows laptop using:

- Python (OpenCV + Imutils) for detection
- R + showtext for Bangla/English word clouds

No advanced hardware was used on purpose. The goal was to keep the system simple, so anyone with basic tools could repeat the workflow.

Chapter 9: Workflow Diagram (Text-Based for Paper)



Chapter 10: Deployment, Ethics & Policy Implications

Deployment Strategy

I propose a hybrid architecture: edge nodes perform detection/tracking locally; aggregated anonymized data sent to a central server to preserve bandwidth and privacy. Pilot deployment at two intersections is recommended before city-wide rollout.

Ethics & Privacy

Only anonymized bounding boxes and track-IDs are retained; no storing of raw faces. Retention policy: raw footage retained for 7 days then deleted, embedding stored for 90 days. I produced a model-card documenting bias and limitations (e.g., lower accuracy for children, occluded pedestrians) and recommended continuous monitoring of fairness across demographic groups.

Policy Recommendations

1. Prioritize crosswalk installation and lighting improvement in zones flagged by high-negative sentiment + high pedestrian counts.
2. Use heat-map output to guide sidewalk widening and pedestrian-signal timing optimization.
3. Embed citizen-feedback sentiment module into traffic-authority dashboards for participatory monitoring.
4. Develop municipal data-sharing protocols and open APIs for research and transparency.

Chapter 11: Conclusions & Future Work

In this work, I built a simple but meaningful vision-plus-NLP system to understand how pedestrians move through Dhaka and how they feel about the walking experience. Even though the tools Ire lightweight, the system worked surprisingly Ill. The image and video modules could consistently detect people in both still scenes and moving footage, and the sentiment analysis added an emotional layer that helped us see the city from the perspective of real pedestrians. When I brought everything together, the system offered a more human view of Dhaka's footpaths — not just where people are, but also what they struggle with.

There is still a lot of room to grow. One natural next step is to scale this system across multiple cameras so pedestrians can be re-identified across different parts of the city. Dhaka's lighting conditions also change sharply between day and night, and improving detection after sunset is something I want to focus on. Adding thermal or low-light imagery could help in areas where visibility is poor or electricity cuts are common. Another direction is to model pedestrian–vehicle conflicts, especially at busy crossings, so warnings or alerts can be generated before risky situations occur.

Looking further ahead, I hope to curate and open-source a Dhaka-specific pedestrian dataset, with all the proper privacy safeguards. This would support more local research, since Dhaka's walking environments are very different from the cities where most benchmark datasets are created. Finally, the long-term goal is to work with the city authorities and local organizations so this system can actually be used on the ground. If adopted properly, even a lightweight setup like this can help improve footpaths, guide renovations, and make walking in Dhaka a little safer and a little easier for everyone.

References

[1] u, D., Cao, X., Zhang, X., & Ye, Q. (2022). *From handcrafted to deep features for pedestrian detection: A survey*. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*(11), 8845–8865.

[2] Angelova, A., Krizhevsky, A., & Vanhoucke, V. (2015). *Real-time pedestrian detection with deep network cascades*. In *Proceedings of the British Machine Vision Conference (BMVC 2015)*. BMVA Press.

[3] Leal-Taixé, L., Milan, A., Reid, I., Roth, S., & Schindler, K. (2015). *MOTChallenge 2015: Towards a benchmark for multi-target tracking*. *arXiv preprint arXiv:1504.01942*.

[4] Dollár, P., Wojek, C., Schiele, B., & Perona, P. (2012). *Pedestrian detection: An evaluation of the state of the art*. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 34*(4), 743–761.

[5] Zhang, S., Benenson, R., & Schiele, B. (2017). *CityPersons: A diverse dataset for pedestrian detection*. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)* (pp. —). IEEE.

[6] Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X., & Sun, J. (2018). *CrowdHuman: A benchmark for detecting human in a crowd*. *arXiv preprint arXiv:1805.00123*

[7] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Yuan, Z., Luo, P., & Liu, W. (2022). *ByteTrack: Multi-object tracking by associating every detection box*. In *Proceedings of the European Conference on Computer Vision (ECCV 2022)* (pp. 1–21). Springer.

[8] Wojke, N., Bewley, A., & Paulus, D. (2017). *Simple online and realtime tracking with a deep association metric*. In *Proceedings of the IEEE International Conference on Image Processing (ICIP 2017)* (pp. 3645–3649).

[9] Zhou, K., Yang, Y., Cavallaro, A., & Xiang, T. (2019). *Omni-scale feature learning for person re-identification*. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2019)* (pp. 3702–3712). IEEE.

[10] Ghari, B., Tmyani, A., Shahbahrami, A., & Gaydadjiev, G. (2023). *Pedestrian detection in low-light conditions: A comprehensive survey*. *Jmynal of Visual Communication and Image Representation, 92*, 103616.

Appendix

Python code used for pedestrian detection in image:

```
import cv2
import imutils

# Initializing the HOG person detector
hog = cv2.HOGDescriptor()
hog.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())

# Reading the Image
image = cv2.imread('img.png')

# Resizing the Image
image = imutils.resize(image, width=min(400, image.shape[1]))

# Detecting all regions that contain pedestrians
(regions, _) = hog.detectMultiScale(image,
                                    winStride=(4, 4),
                                    padding=(4, 4),
                                    scale=1.05)

# Drawing rectangles around detected pedestrians
for (x, y, w, h) in regions:
    cv2.rectangle(image, (x, y),
                  (x + w, y + h),
                  (0, 0, 255), 2)

# Displaying the output
cv2.imshow("Image", image)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Figure 9: Code Used to Generate Images

Python script used for pedestrian detection in video:

```
import cv2
import imutils

hog = cv2.HOGDescriptor()
hog.setSVMDetector(cv2.HOGDescriptor_getDefaultPeopleDetector())

cap = cv2.VideoCapture('video.mp4')

while cap.isOpened():
   ret, frame = cap.read()
   if not ret:
      break

   frame = imutils.resize(frame, width=min(400, frame.shape[1]))

   regions, _ = hog.detectMultiScale(frame,
                              winStride=(4, 4),
                              padding=(4, 4),
                              scale=1.05)
   for (x, y, w, h) in regions:
      cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 0, 255), 2)

   cv2.imshow("Video", frame)

   if cv2.waitKey(25) & 0xFF == ord('q'):
      break

cap.release()
cv2.destroyAllWindows()
```

Figure 10: Code Used to Generate Video

List of Figures