Accurate Classification of True Face Trajectories in Video Data Using Distance-Based K-NN for Real-Time Surveillance Systems

Shravya Parusha (Shravya P)
Master of Science, Computer Science & Information Systems
Sacred Heart University

Abstract

Real-time object and facial detection systems are increasingly essential in public safety, retail analytics, and intelligent surveillance. Traditional computer vision techniques often struggle with dynamic environments, occlusions, and false trajectory classification. This research presents a real-time facial trajectory classification system using distance-based K-Nearest Neighbors (K-NN) with Earth Mover's Distance (EMD) to improve accuracy in video stream monitoring. The system integrates OpenCV-based preprocessing, YOLO/SSD detection models, centroid tracking, and trajectory validation mechanisms. Experimental results achieved 93% classification accuracy on real surveillance-style datasets while maintaining real-time processing speeds above 25 FPS. The study demonstrates that combining classical ML algorithms with modern object detection frameworks significantly enhances real-time deployability.

---

1. Introduction

Object and people detection systems are fundamental in modern AI-driven surveillance and analytics. Applications include:

- Retail footfall analytics
- Crowd management
- Security surveillance

- Traffic monitoring
- Smart city infrastructure

Traditional techniques such as sliding window detection and background subtraction often suffer from latency and false detections. To overcome these limitations, this research proposes a hybrid architecture combining deep learning object detection with trajectory-based classification using K-NN.

---

2. Problem Statement

Existing video surveillance systems struggle with:

- False-positive trajectory detection
- Occlusion handling
- Identity switching in multi-object tracking
- Real-time processing constraints

The goal of this research is to accurately classify true face trajectories in video streams while maintaining real-time performance and system scalability.

---

3. Methodology

3.1 System Architecture

The system pipeline:

1. Video frame acquisition (OpenCV)
2. Preprocessing (grayscale conversion, background subtraction)
3. Object detection using YOLO / SSD
4. Centroid-based object tracking

5. Feature extraction from trajectory paths
6. Distance-based K-NN classification
7. Visualization and UID assignment

3.2 Object Detection

YOLO/SSD models were selected because:

- Single-shot detection enables real-time performance
- High accuracy on standard datasets
- Efficient inference in live environments

3.3 Trajectory Classification Using K-NN

A custom trajectory classification module was implemented:

- Trajectory features extracted as spatial-temporal vectors
- Earth Mover's Distance (EMD) used to compare trajectory histograms
- K-NN applied to classify true vs false face paths

EMD was preferred over Euclidean distance due to improved similarity comparison for histogram-based movement distributions.

---

4. Implementation

Technologies used:

- Python

- OpenCV
- NumPy, Pandas
- TensorFlow / PyTorch (for detection models)
- MATLAB (initial people counting prototype)

Key implementation highlights:

- Real-time bounding box rendering
- UID generation for tracked objects
- Entry/exit counting logic
- Frame rate optimization using frame sampling

---

5. Experimental Results

Performance Metrics:

- Classification Accuracy: 93%
- FPS: 25–30 (real-time capable)
- Model Load Time: < 3 seconds
- CPU/GPU utilization within acceptable thresholds

Latency optimization techniques reduced inference time significantly by:

- Minimizing preprocessing overhead
- Using optimized frame intervals
- Efficient feature vector storage

---

6. Applications

This system can be deployed in:

- Retail analytics (customer behavior tracking)
- Public safety monitoring
- Airport and transport analytics
- Smart traffic systems
- Industrial monitoring

---

7. Future Enhancements

Future work may include:

- Integration of DeepSORT for improved tracking
- Deployment on edge devices such as NVIDIA Jetson
- Transformer-based detectors (DETR)
- YOLOv8 integration
- Kalman filtering for motion prediction
- Gen-AI-based activity summarization
- Edge deployment using Docker + FastAPI

---

8. Conclusion

This research demonstrates that combining classical machine learning algorithms such as K-NN with modern deep learning object detectors significantly enhances real-time surveillance accuracy. The trajectory-based classification framework improves robustness against false detections and occlusions. The system is scalable, deployable, and suitable for real-world AI-driven monitoring environments.

END OF PAPER