Edge-AI Based Real-Time Violence Detection using ESP32-CAM

Venkat Bhavan Tati
Matrikel-Nr: 1575011
Industrial AI – Edge AI in Industrial Applications (AI5252)
WiSe 25/26

https://github.com/venkatbhavan/Edge-AI-Violence-Detection.git

---

1. Motivation and Problem Statement

Violence and physical aggression in public and industrial environments pose significant safety and security risks. Traditional surveillance systems rely on continuous human monitoring, which is inefficient, costly, and prone to delayed or missed responses. In large-scale environments such as industrial facilities, campuses, or public infrastructure, manual monitoring becomes increasingly impractical.

With the growing availability of low-cost embedded vision devices, intelligent systems capable of automatically analyzing human behavior in real time are becoming essential. Such systems enhance situational awareness and support human operators by generating early warnings.

This project aims to design a low-cost Edge-AI surveillance system capable of detecting violent behavior automatically in real time, suitable for industrial safety monitoring and smart infrastructure applications.

---

2. Project Objective

The objective is to design and implement a distributed Edge-AI system consisting of:

- ESP32-CAM as edge sensing device
- Lightweight CNN-based violence detection model
- Embedded inference service accessible via network

The system performs binary classification (violence vs non-violence) and provides:

- Real-time visualization
- Local edge inference
- Remote access via mobile browser

The architecture is modular and extensible for:

- Weapon detection
- Temporal sequence modeling
- Alert notification integration

---

3. System Architecture

The system consists of two edge components and remote client devices:

Edge Device 1 – ESP32-CAM
Responsible for live video acquisition.

Edge Device 2 – Edge Inference Node (Laptop)
Performs preprocessing, deep learning inference, and web service hosting.

Remote Client Devices – Smartphone / Tablet
Access inference results via embedded web interface.

System Workflow

1. ESP32-CAM captures live video.
2. Frames are transmitted via Wi-Fi.
3. Edge inference node performs:
    - Frame preprocessing

- ○ Motion extraction
- ○ CNN classification

4. Results are visualized locally and made available via an embedded web interface.
5. Mobile devices access inference results through IP address.

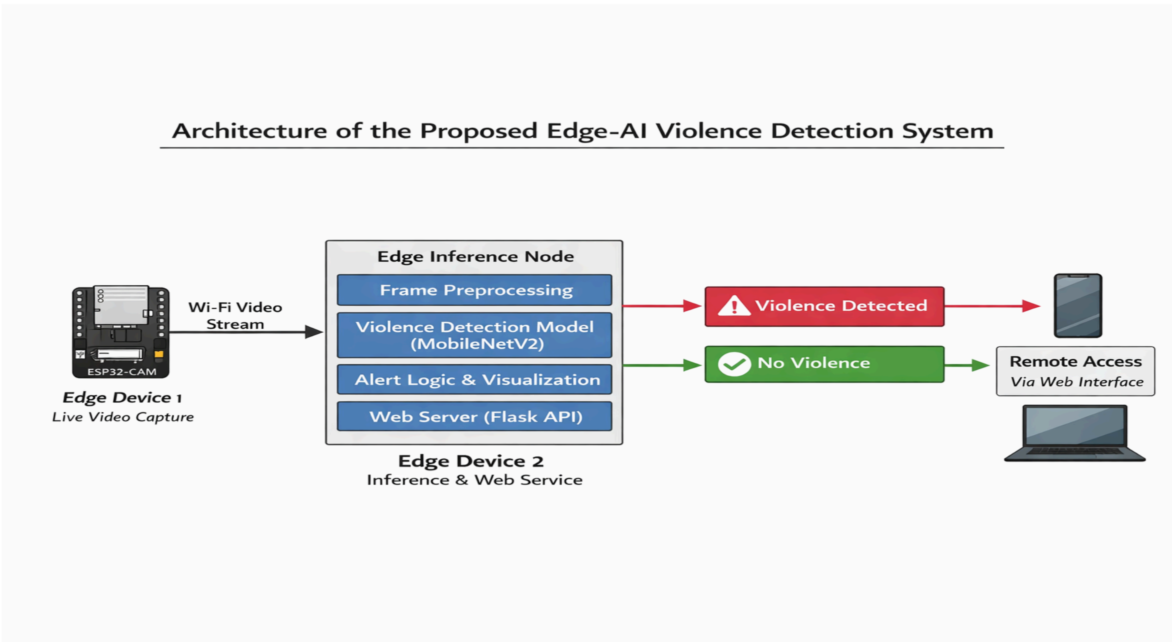

Figure 1: Architecture of the Proposed Edge-AI Violence Detection System

---

4. AI Methodology

4.1 Problem Formulation

Binary classification:

- Class 0 – Non-Violence
- Class 1 – Violence

4.2 Motion-Based Representation

To emphasize dynamic behavior and reduce background noise:

$$\text{Motion}(t) = |\ \text{Frame}(t) - \text{Frame}(t-1)\ |$$

This improves robustness against lighting variation and static backgrounds.

4.3 Model Architecture

The model is based on **MobileNetV2**, chosen for:

- Depthwise separable convolutions
- Low computational complexity
- Suitability for edge deployment

Input: 96×96 grayscale
Output: Sigmoid activation (violence probability)

Transfer learning from ImageNet weights was applied.

4.4 Inference Pipeline

The real-time inference pipeline consists of:

1. Frame capture
2. Grayscale conversion
3. Resize to 96×96
4. Normalization (0–1 range)
5. CNN forward pass
6. Threshold decision logic
7. Overlay visualization

This pipeline ensures deterministic and low-latency inference suitable for edge deployment.

---

5. Data and Tools

Dataset

- Public violence/non-violence video datasets
- Converted to motion-based image samples
- Balanced classes
- Realistic camera perspectives

Software Tools

- Python
- OpenCV
- TensorFlow / Keras
- Flask
- NumPy

Hardware

- ESP32-CAM
- Laptop (Edge inference node)
- Smartphone (Client device)

---

6. Edge-AI Architecture and Multi-Edge Deployment

6.1 Sensing Edge

ESP32-CAM performs real-time acquisition but not inference.

6.2 Inference Edge Node

Laptop acts as local edge compute device performing:

- Motion preprocessing
- CNN inference
- Decision thresholding
- Embedded web service

Although implemented on a laptop, this represents deployable edge hardware such as:

- Industrial PC
- NVIDIA Jetson
- Raspberry Pi (accelerated)

Inference remains local and does not depend on cloud infrastructure.

6.3 Monitoring Client

Mobile devices access results via IP-based web interface.

This demonstrates multi-device accessibility and distributed Edge-AI deployment capability.

---

7. Model Evaluation and Performance Metrics

Performance metrics:

- Accuracy
- Precision
- Recall
- Weighted F1-Score

Accuracy alone is insufficient in safety-critical systems. False negatives and false positives must be explicitly considered.

7.1 Classification Report

```
==============================
Classification Report:
==============================
              precision    recall  f1-score   support

Non_violence       0.97      0.93      0.95     24418
    violence       0.95      0.97      0.96     30667

    accuracy                           0.96     55085
   macro avg       0.96      0.95      0.96     55085
weighted avg       0.96      0.96      0.96     55085


Weighted F1 Score: 0.9565
```

Figure 2: Classification Report of the Violence Detection Model

Results:

- Accuracy: 96%
- Precision (Violence): 0.95
- Recall (Violence): 0.97

- Weighted F1-Score: 0.9565

The high recall for the violence class indicates reliable detection of critical events.

7.2 Confusion Matrix

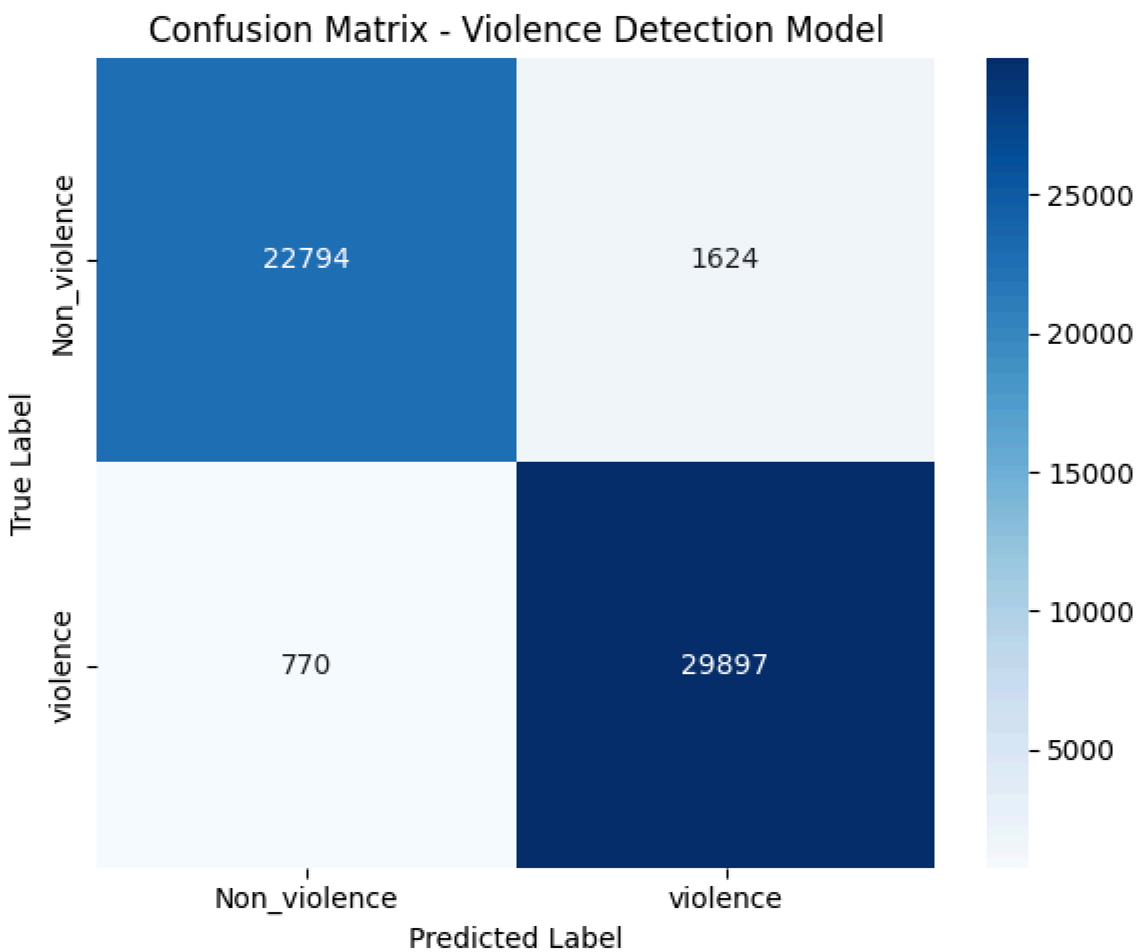

Figure 3: Confusion Matrix of the Violence Detection Model

Correct classifications:

- 22,794 non-violent samples
- 29,897 violent samples

Errors:

- False negatives: 770
- False positives: 1,624

Low false negative rate is particularly important for safety monitoring.

---

8. GDPR, Privacy and Ethical Considerations

The system follows privacy-by-design principles.

Data Minimization

No persistent video storage. Frames processed in volatile memory only.

Local Processing

No cloud transmission. Inference occurs locally.

No Biometric Identification

The system does not perform:

- Facial recognition
- Identity tracking
- Biometric profiling

Controlled Network Deployment

Operates in local network environment. Industrial deployment may include:

- TLS encryption
- Authentication
- Role-based access control
- Logging policies

The system is designed as an assistive tool, not an autonomous enforcement system.

In real-world industrial deployment, a Data Protection Impact Assessment (DPIA) would be conducted to evaluate potential privacy risks. The current prototype is designed for research and demonstration purposes and does not store or process identifiable personal data beyond transient frame analysis.

---

9. Why Edge Impulse Was Not Used

Edge Impulse was evaluated but not selected due to:

- Custom motion preprocessing requirements
- Need for architectural flexibility
- Full control over training pipeline
- Custom web-based deployment

Direct TensorFlow implementation provided:

- Greater transparency
- Industrial flexibility
- Deployment customization

---

10. Limitations

- Limited dataset diversity
- Motion-based sensitivity
- ESP32 Wi-Fi latency
- No hardware acceleration

11. Conclusion

This project demonstrates a distributed Edge-AI system combining:

- Edge sensing (ESP32-CAM)
- Lightweight CNN inference
- Real-time classification
- Network-accessible monitoring
- GDPR-conscious design

The distributed sensing–inference–monitoring architecture reflects modern Industrial Edge-AI paradigms where intelligent decision-making occurs close to the data source.

The architecture reflects practical industrial Edge-AI deployment models and provides a scalable foundation for future expansion.