THE LATEST BREAKTHROUGH EXPLAINED

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

Motivation & Problem Statement

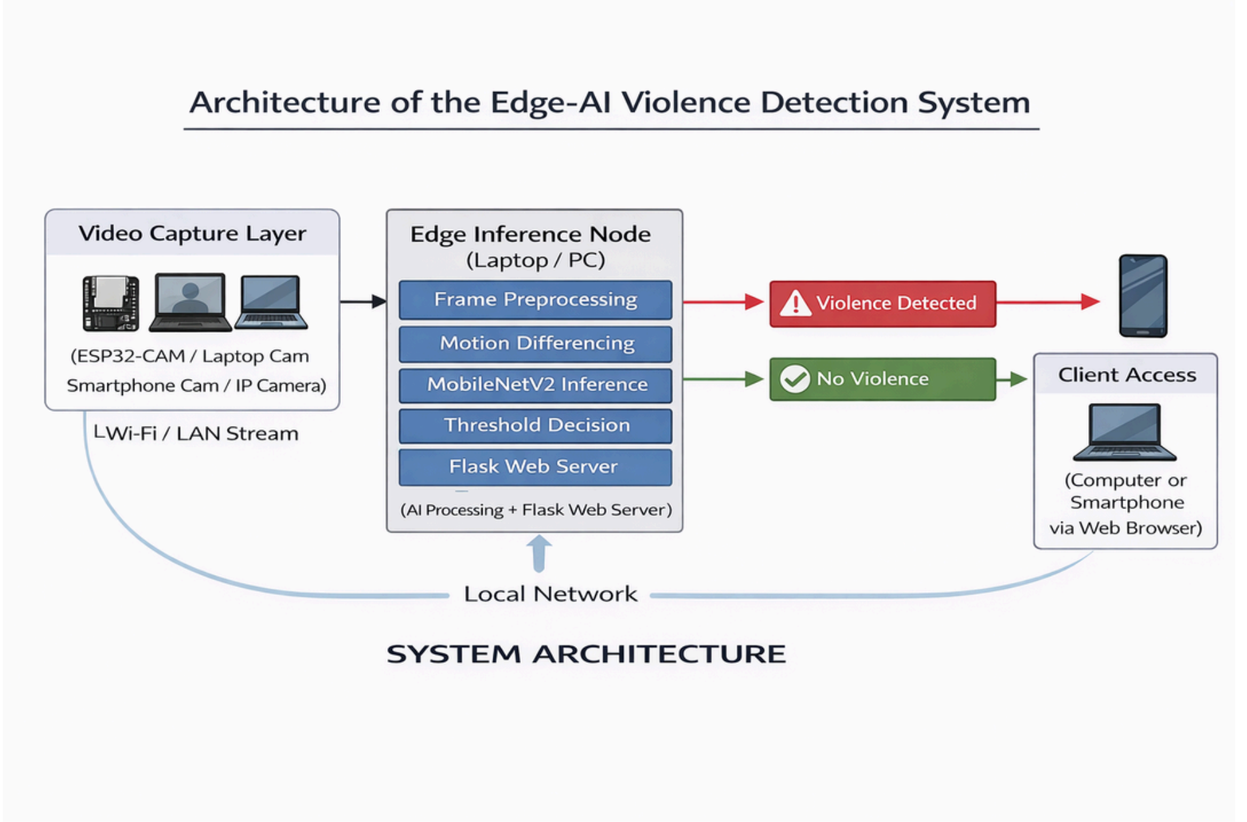
- VIOLENCE DETECTION IN INDUSTRIAL & PUBLIC ENVIRONMENTS
- MANUAL SURVEILLANCE IS INEFFICIENT AND ERROR-PRONE
- DELAYED HUMAN REACTION INCREASES SAFETY RISK
- NEED FOR INTELLIGENT, AUTOMATED, REAL-TIME MONITORING

Project Objective

- Develop distributed Edge-AI violence detection system
- Binary classification: Violence vs Non-Violence
- Real-time inference at edge
- Remote access via mobile browser
- GDPR-conscious architecture

System Architecture

- Video Capture Layer → ESP32-CAM / Laptop Cam / Smartphone Cam / IP Camera
- Edge Inference Node → Laptop / PC (AI Processing + Flask Web Server)
- Client Access → Computer or Smartphone via Web Browser
- Communication → Local Network (Wi-Fi / LAN)



Problem

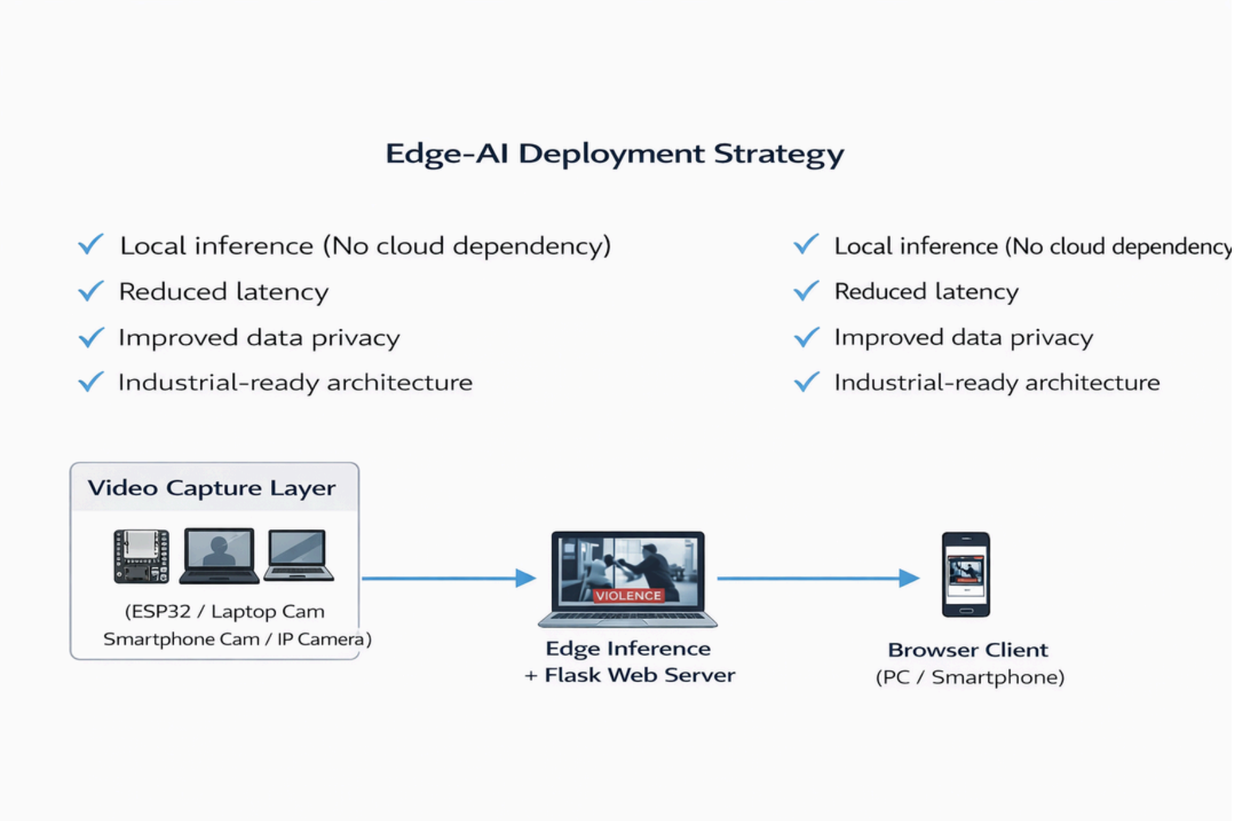
- The pre-5G internet experience was slower and limited Manual surveillance is inefficient
- Delayed response to violent incidents
- Privacy risks in cloud-based systems to certain spectrum types.

Edge-AI Solution

- Edge sensing (ESP32-CAM)
- Local AI inference
- No cloud dependency
- Real-time alert

Edge-AI Deployment Strategy

- Local inference (No cloud dependency)
- Reduced latency
- Improved data privacy
- Industrial-ready architecture



AI Methodology

- Binary classification problem
- Motion-based frame differencing
- MobileNetV2 lightweight CNN
- Transfer learning applied

Instead of using raw frames directly, I compute motion information using frame differencing:

Motion at time t = |Frame(t) − Frame(t−1)|

MobileNetV2 Architecture

- Depthwise separable convolutions
- Reduced computational cost
- Suitable for edge environments
- 96×96 grayscale input
- Sigmoid output (violence probability)

Inference Pipeline

- **Grayscale Conversion**
- Convert RGB frame to grayscale to reduce computational complexity.
- **Resize (96 × 96)**
- Resize frame to match model input dimensions.
- **Normalization**
- Scale pixel values for stable and efficient inference.
- **CNN Forward Pass**
- Feed processed frame into MobileNetV2 model.
- **Threshold Decision**
- Apply sigmoid probability threshold to classify:
- Violence or Non-Violence.
- **Overlay & Display**
- Display prediction result on video stream via Flask web interface.

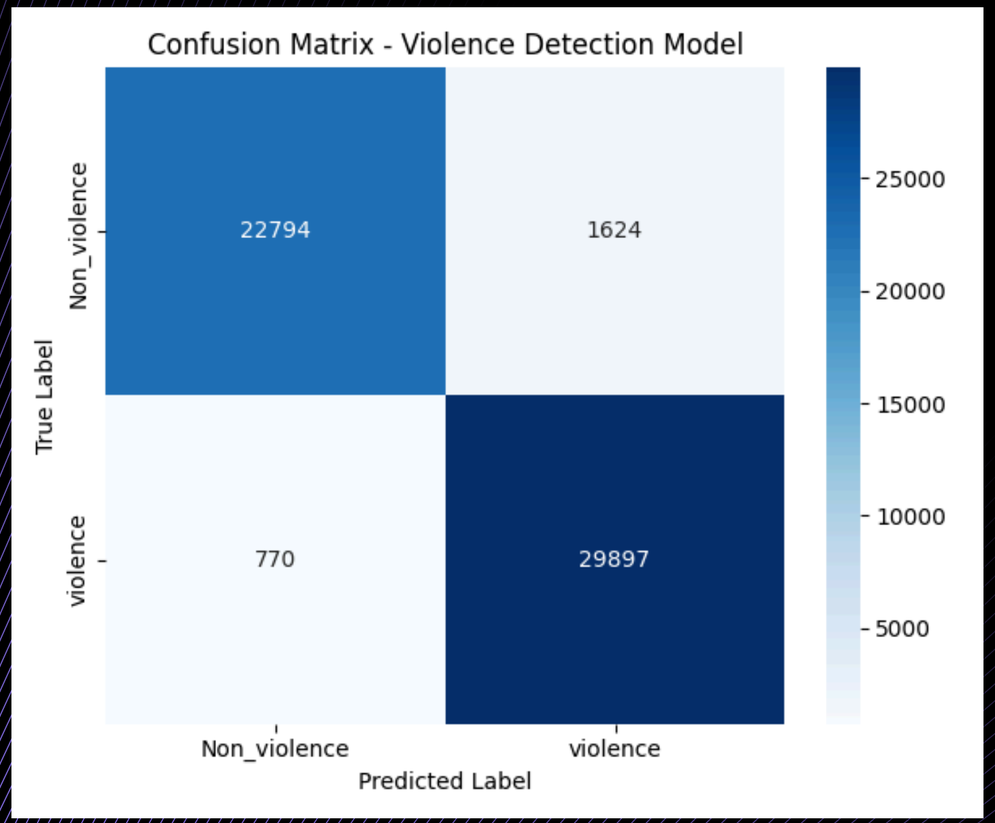
Performance Metrics

```
==============================
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
```

PERFORMANCE METRICS

- Accuracy: 96%
- Precision (Violence): 0.95
- Recall (Violence): 0.97
- Weighted F1-Score: 0.9565

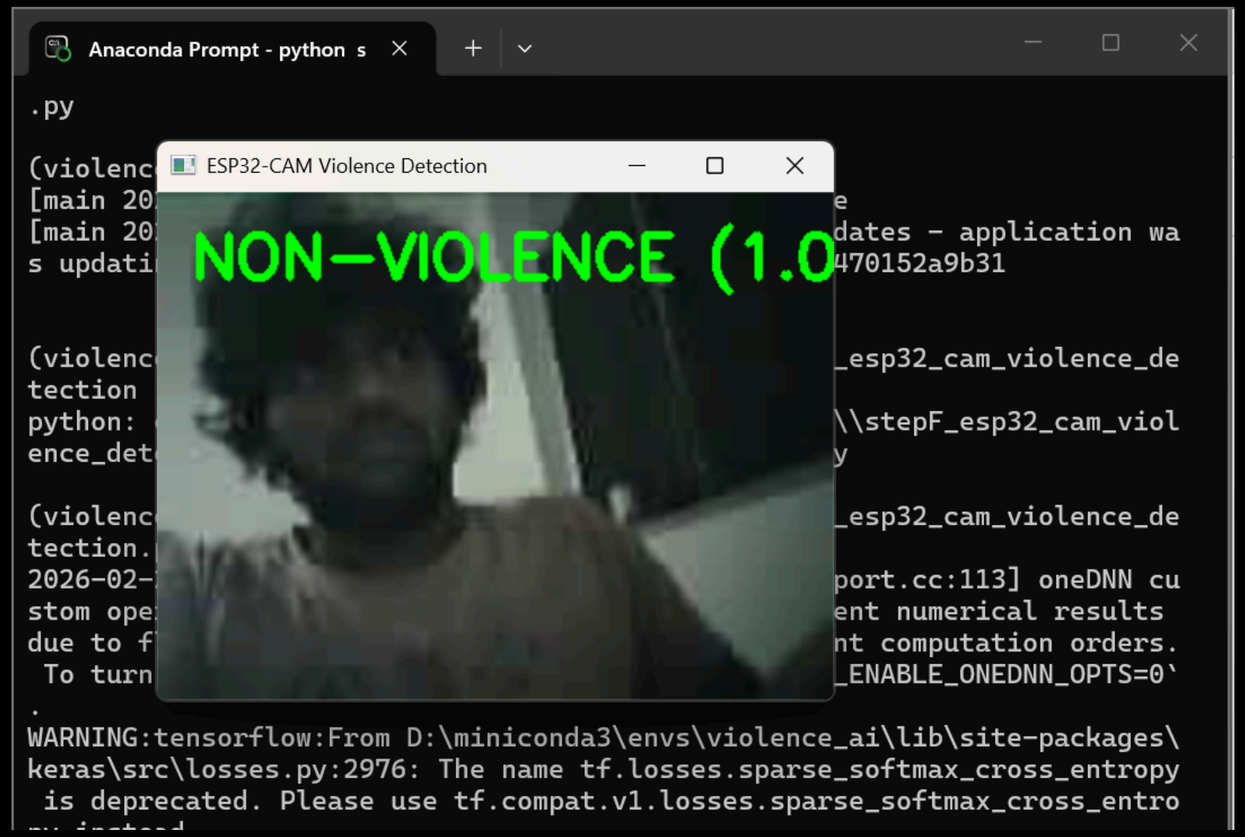
Confusion Matrix

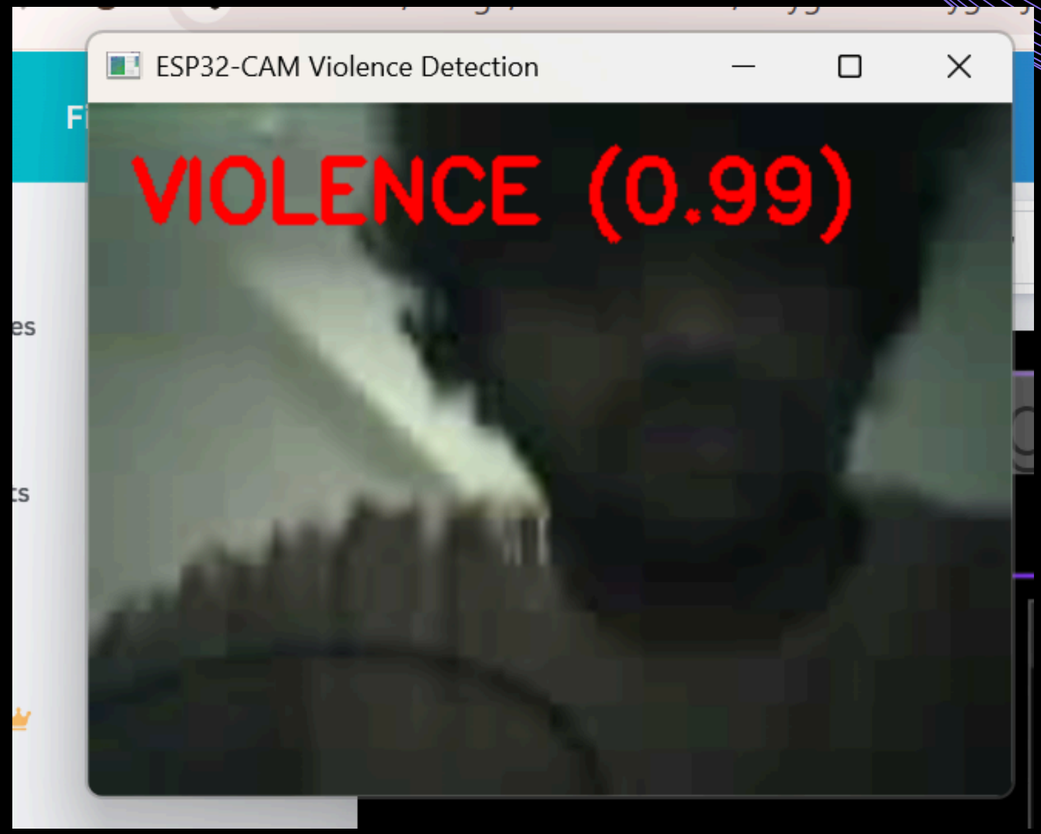


CONFUSION MATRIX

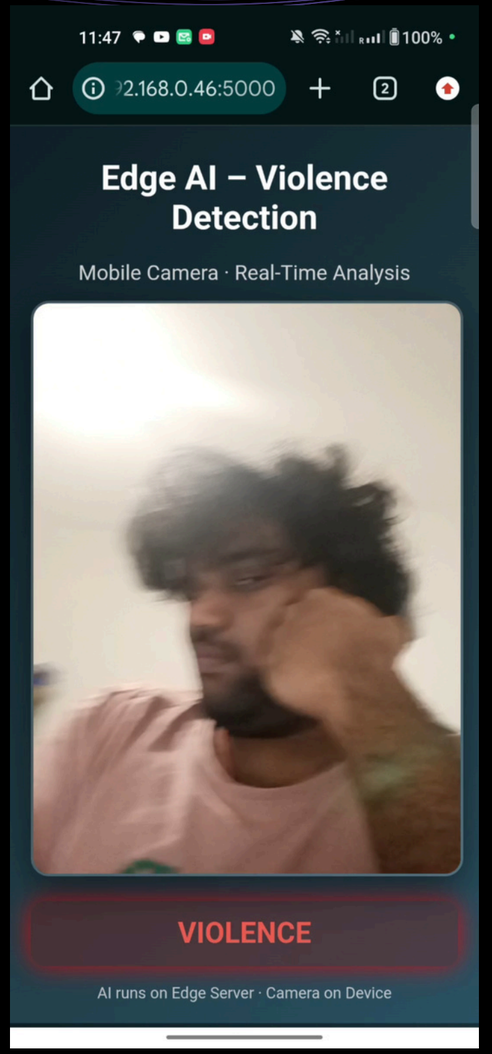
- 29,897 violent samples correctly classified
- Low false negative rate
- Safety-critical performance

Detection using esp32 cam

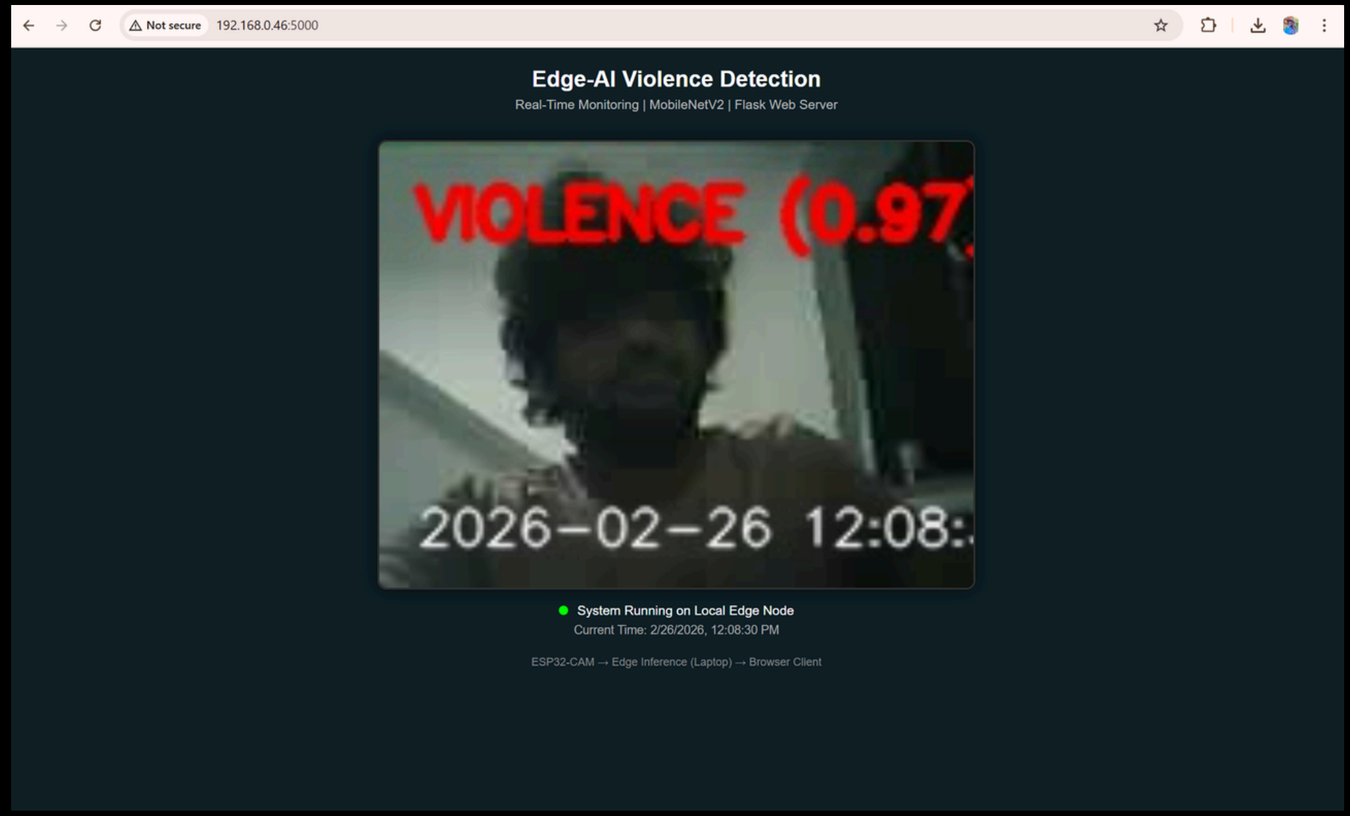




Detection using phone(flask web server)



Detection using esp32 cam(flask web server)



GDPR & Privacy Compliance

GDPR & PRIVACY COMPLIANCE

- No video storage
- No cloud transmission
- No biometric identification
- Local network processing
- Privacy-by-design architecture

Why Direct TensorFlow Implementation?

- CUSTOM MOTION PREPROCESSING
- FULL ARCHITECTURAL CONTROL
- FLEXIBLE DEPLOYMENT
- INDUSTRIAL WORKFLOW ALIGNMENT

Limitations & Future Work

- MOTION SENSITIVITY
- ESP32 WI-FI LATENCY
- DATASET DIVERSITY
- FUTURE: WEAPON DETECTION
- FUTURE: TEMPORAL MODELING

Conclusion

- DISTRIBUTED EDGE-AI ARCHITECTURE
- REAL-TIME VIOLENCE DETECTION
- MULTI-DEVICE ACCESSIBILITY
- GDPR-CONSCIOUS SYSTEM
- INDUSTRIALLY SCALABLE

Do you have any questions?

Send it to us! We hope you learned something new.