Violence Detection Using AI in Jail

Group no : 15

GUIDED BY:

MR.VIJU P POONTHOTTAM

ASST. PROFESSOR

ADS DEPT

TEAM MEMBERS:

AHAMMED YASEEN O -	MES22AD008
ABHINAND S-	MES22AD002
MOHAMMED RIHAN-	MES22AD043

Introduction

- AI-driven system for jail violence detection.
- Uses deep learning for real-time monitoring.
- Enhances security and reduces manual effort.
- Detects and alerts violent incidents.
- Helps prison staff take quick action.

Literature Review

Title	Author	Method	Result	Accuracy
Violence Detection in Jails and Mental Asylums	Ashwin Saji Kumar et al., 2024 (IJISRT)	LRCN (CNN+LSTM)	High accuracy in detecting violent activities, real-time alerting using a Telegram bot	92%
The Risk Screener Violence (RS-V): Retrospective Prediction of Violent and Aggressive Incidents in Prison	Marjam V. Smeekens et al., 2024 (Frontiers in Psychology)	Risk Screening Tool (RS-V)	RS-V successfully predicts future violent incidents with medium to large predictive values	85%
Preventing Violence and Sexual Assault in Jail: A Situational Crime Prevention Approach	Nancy G. La Vigne et al., 2011 (Urban Institute)	Situational Crime Prevention	Increased inmate safety, reduced violent incidents through environmental control	78%
Real-Time Violence Detection and Alert System	Manjit Kumar Gautam et al., 2024 (IJRASET)	MobileNetV2-based CNN model	Faster violence detection, efficient real-time alerting through image processing and facial recognition	88%

Problem Statement

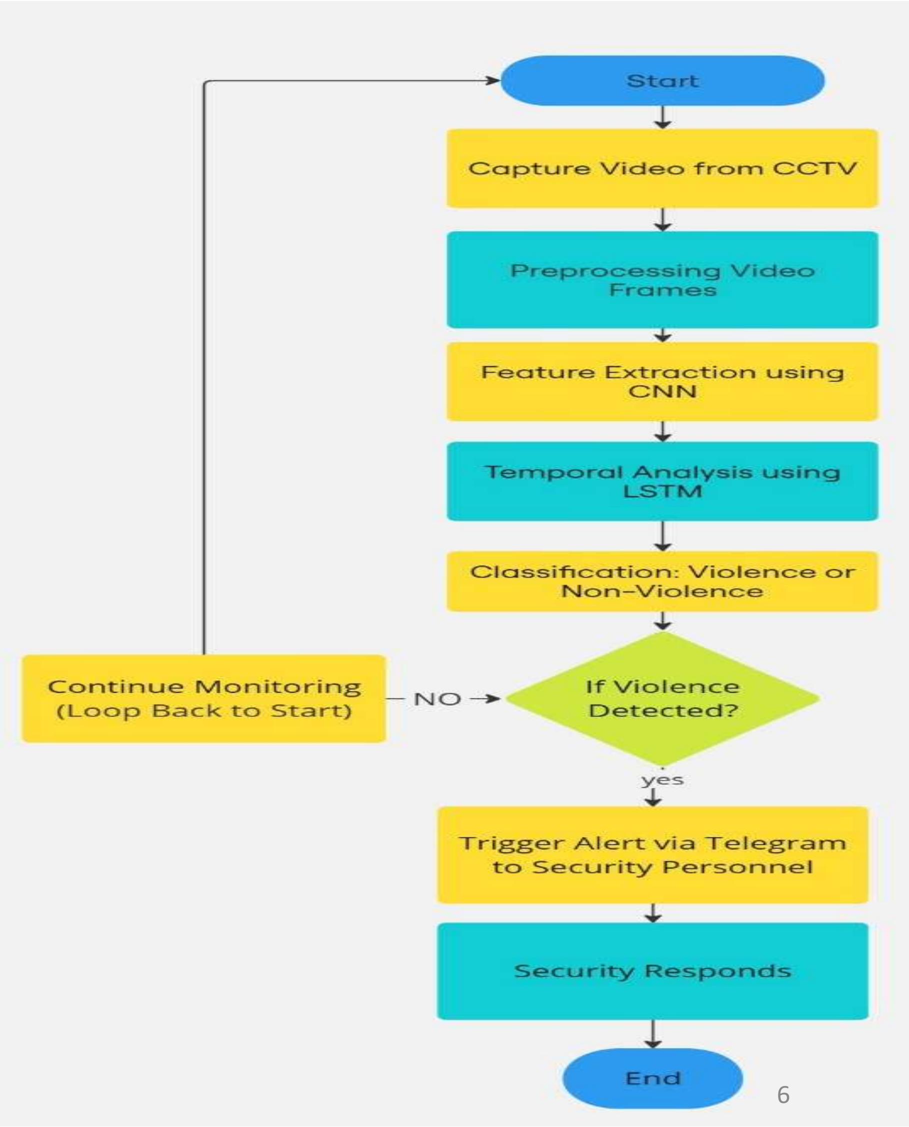
- High inmate violence rates.
- Manual surveillance is inefficient.
- Delayed responses cause injuries.
- Overcrowding increases risks.
- Need for automated solutions.

Methodology

- Uses CNN + LSTM for analysis.
- Detects aggressive behavior.
- Integrates with CCTV cameras.
- Sends alerts via Telegram bot.
- Trained on violence datasets.

Design & Implementation

- Captures and processes video input.
- Extracts frames for feature detection.
- CNN identifies aggressive actions.
- LSTM tracks motion over time.
- Generates confidence score.


Start
Capture Video from CCTV
Preprocessing Video Frames
Feature Extraction using CNN
Temporal Analysis using LSTM
Classification: Violence or Non-Violence
If Violence Detected?
NO
Continue Monitoring (Loop Back to Start)
yes
Trigger Alert via Telegram to Security Personnel
Security Responds
End

Experimental Setup

- **Dataset:** RWF-2000 Violence Dataset
- **Tools Used:** OpenCV, TensorFlow, Python, Google Colab
- **Hardware:** GPU-enabled system for deep learning training
- **Colab Integration:** Model training and evaluation conducted in Google Colab for efficient resource utilization

Working of the Jail Violence Detection System

- **Video Input & Preprocessing**
- **Video Upload**
- Accepts live CCTV footage or pre-recorded videos.
- Users can manually upload videos for analysis
- **Frame Extraction**
- Extracts frames from the video at regular intervals.
- Saves frames as images for further processing.

Image Preprocessing

- Resizes frames (e.g., **64×64 pixels**) for model input.
- Normalizes pixel values between **0 and 1**.
- Converts images to **RGB format** for better analysis

Working of the Jail Violence Detection System

- **Violence Detection Using CNN + LSTM Model**
- **Feature Extraction using CNN**
- Detects aggressive behavior, motion patterns, and key features.

Sequence Learning using LSTM

- Analyzes **multiple frames in sequence** to track movement.
- Differentiates between normal activity and violent actions.

Violence Prediction

- The model classifies video sequences as **Violence** or **Non-Violence**.
- Generates a **confidence score** (e.g., 85% violence probability).

Working of the Jail Violence Detection System

- **Alert System & Results Display**

Real-Time Alert Generation

- Sends **instant alerts** when violence is detected.
- Can be integrated with a **Telegram bot** to notify prison staff.

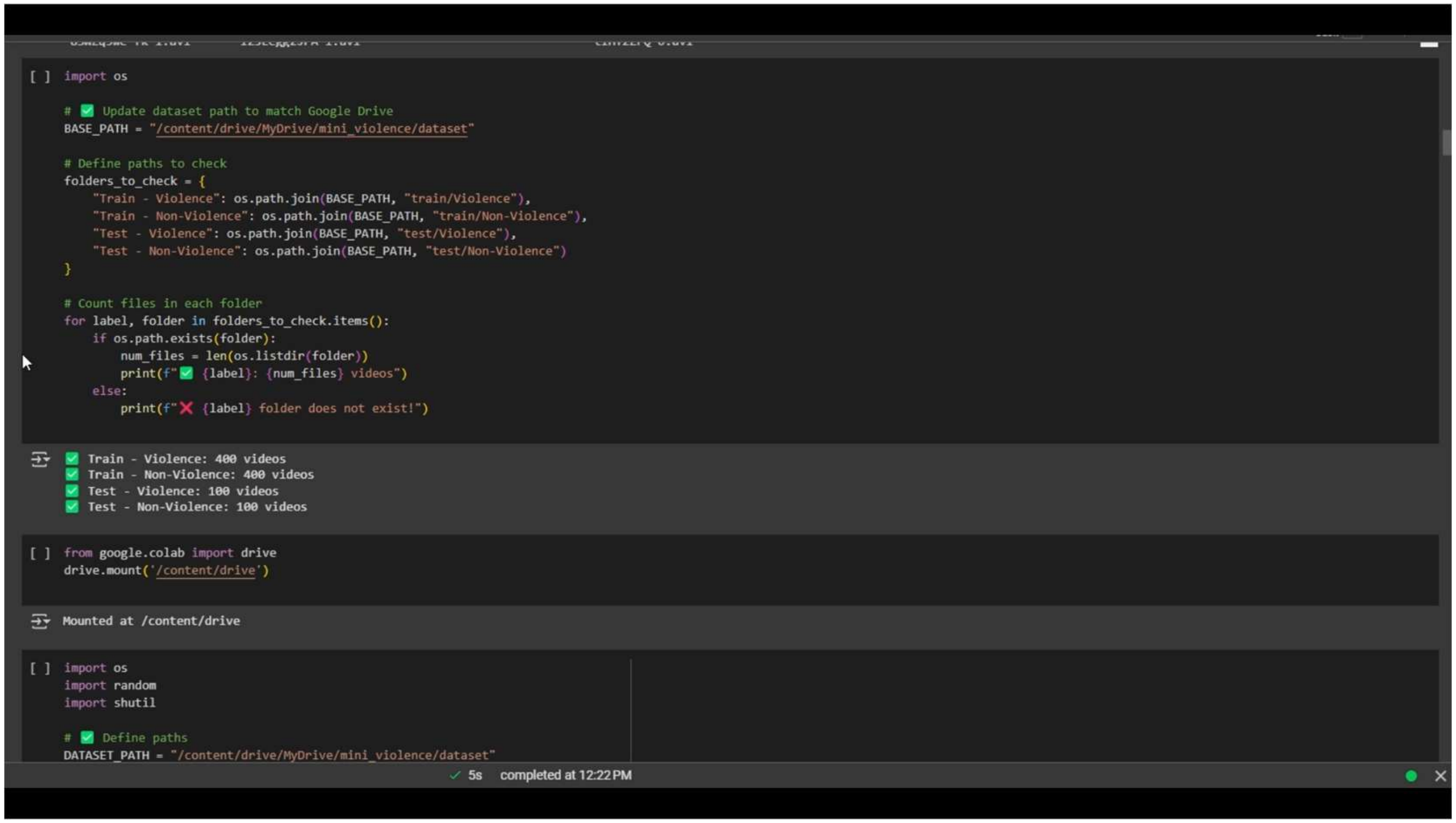
Results Visualization

- Highlights the detected violent frames for evidence.
- Displays a graph of predictions (training vs. validation accuracy)

Report Generation

- Generates a summary report with **timestamps of violent incidents**.
- Stores reports for **future reference and analysis**.

Model Workflow



```python
import os

# ✅ Update dataset path to match Google Drive
BASE_PATH = "/content/drive/MyDrive/mini_violence/dataset"

# Define paths to check
folders_to_check = {
    "Train - Violence": os.path.join(BASE_PATH, "train/Violence"),
    "Train - Non-Violence": os.path.join(BASE_PATH, "train/Non-Violence"),
    "Test - Violence": os.path.join(BASE_PATH, "test/Violence"),
    "Test - Non-Violence": os.path.join(BASE_PATH, "test/Non-Violence")
}

# Count files in each folder
for label, folder in folders_to_check.items():
    if os.path.exists(folder):
        num_files = len(os.listdir(folder))
        print(f"✅ {label}: {num_files} videos")
    else:
        print(f"❌ {label} folder does not exist!")
```

```
✅ Train - Violence: 400 videos
✅ Train - Non-Violence: 400 videos
✅ Test - Violence: 100 videos
✅ Test - Non-Violence: 100 videos
```

```python
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

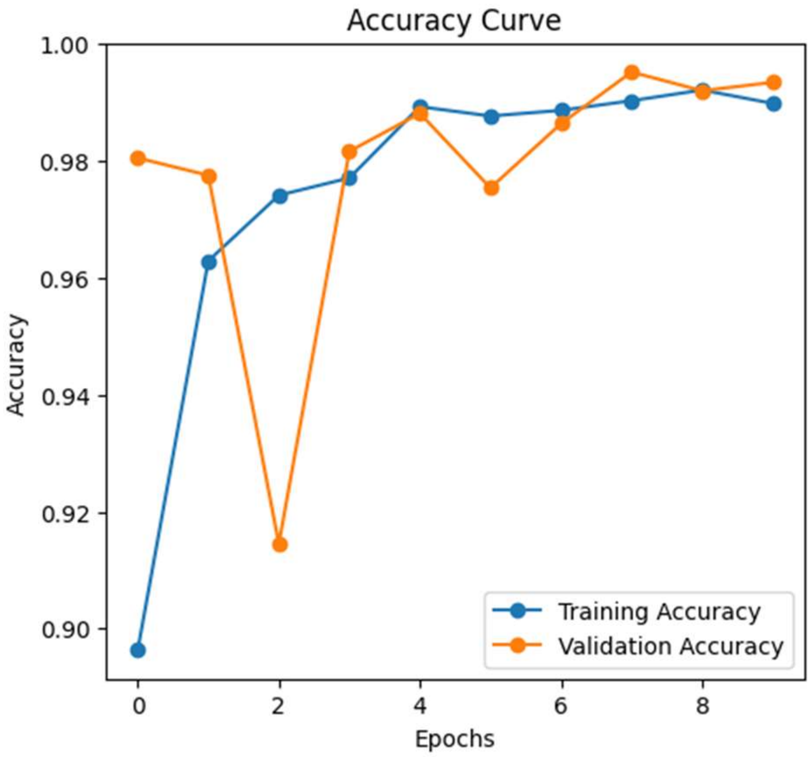
```python
import os
import random
import shutil

# ✅ Define paths
DATASET_PATH = "/content/drive/MyDrive/mini_violence/dataset"
```

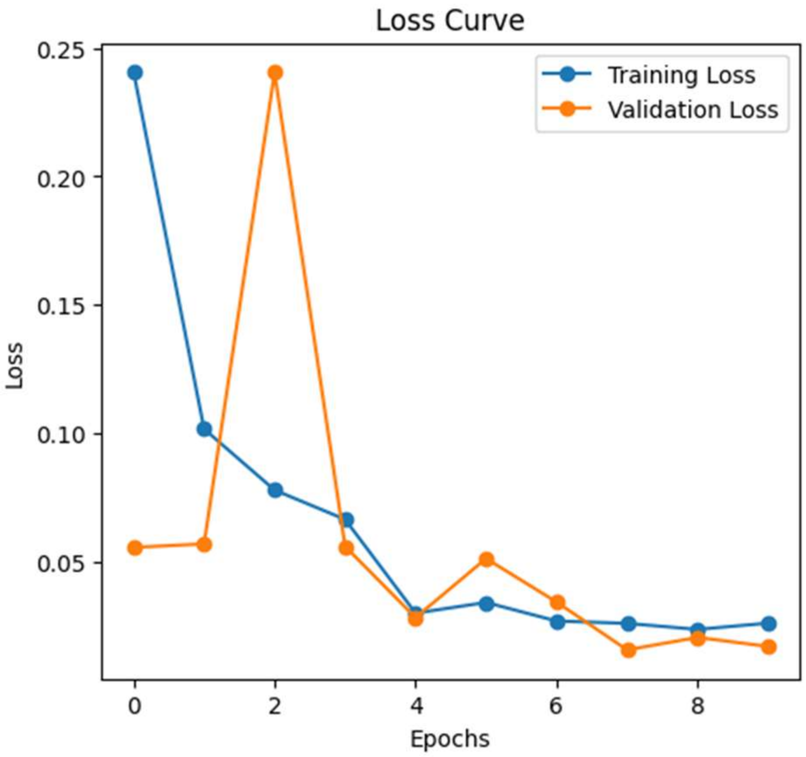
Results & Analysis

Accuracy: 96.71%

- Model correctly classifies **violence vs. non-violence**.
- **Real-time alerting system** reduces response time.
- **Graph comparing training vs. validation accuracy.**

Accuracy & Loss Graphs

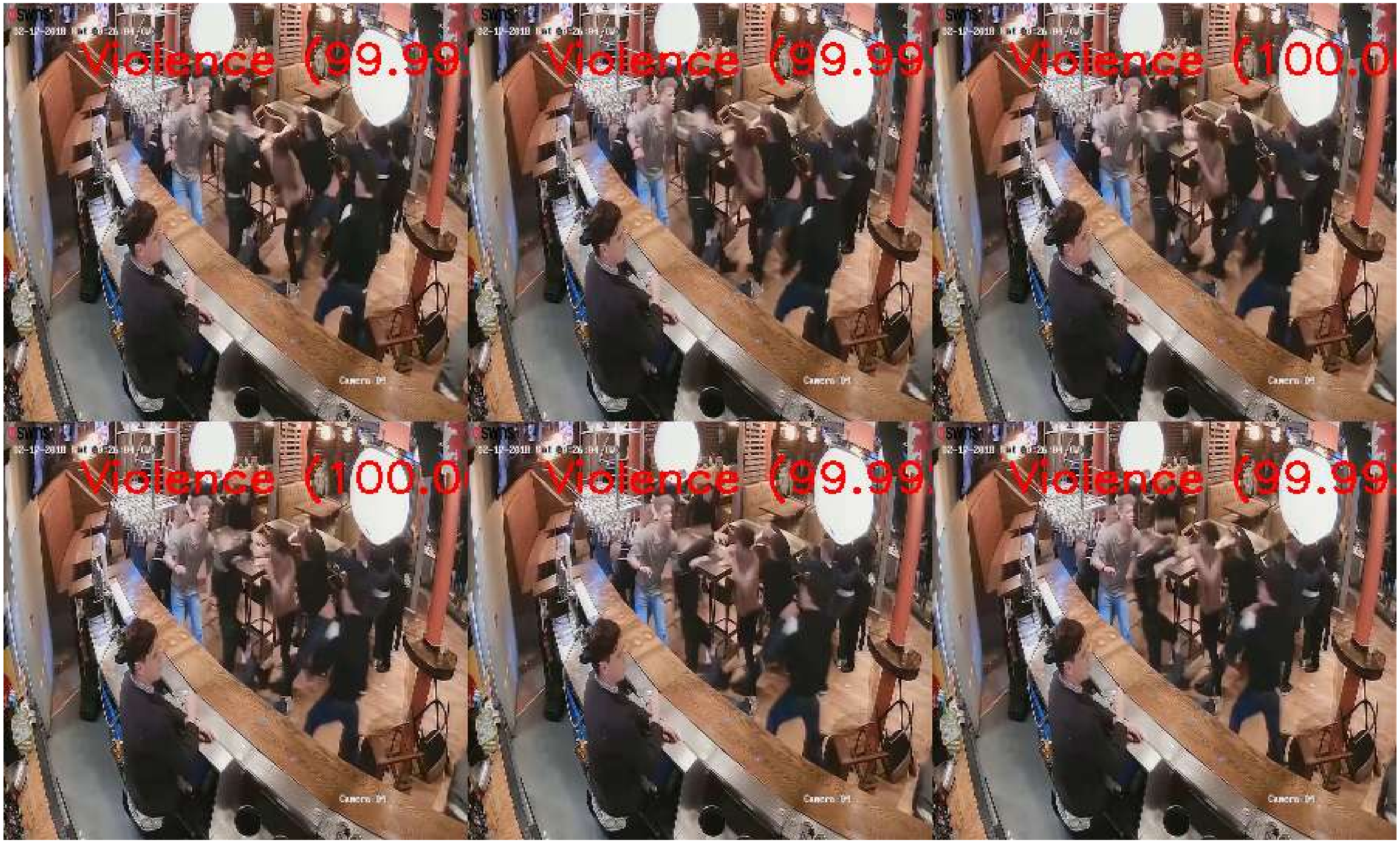




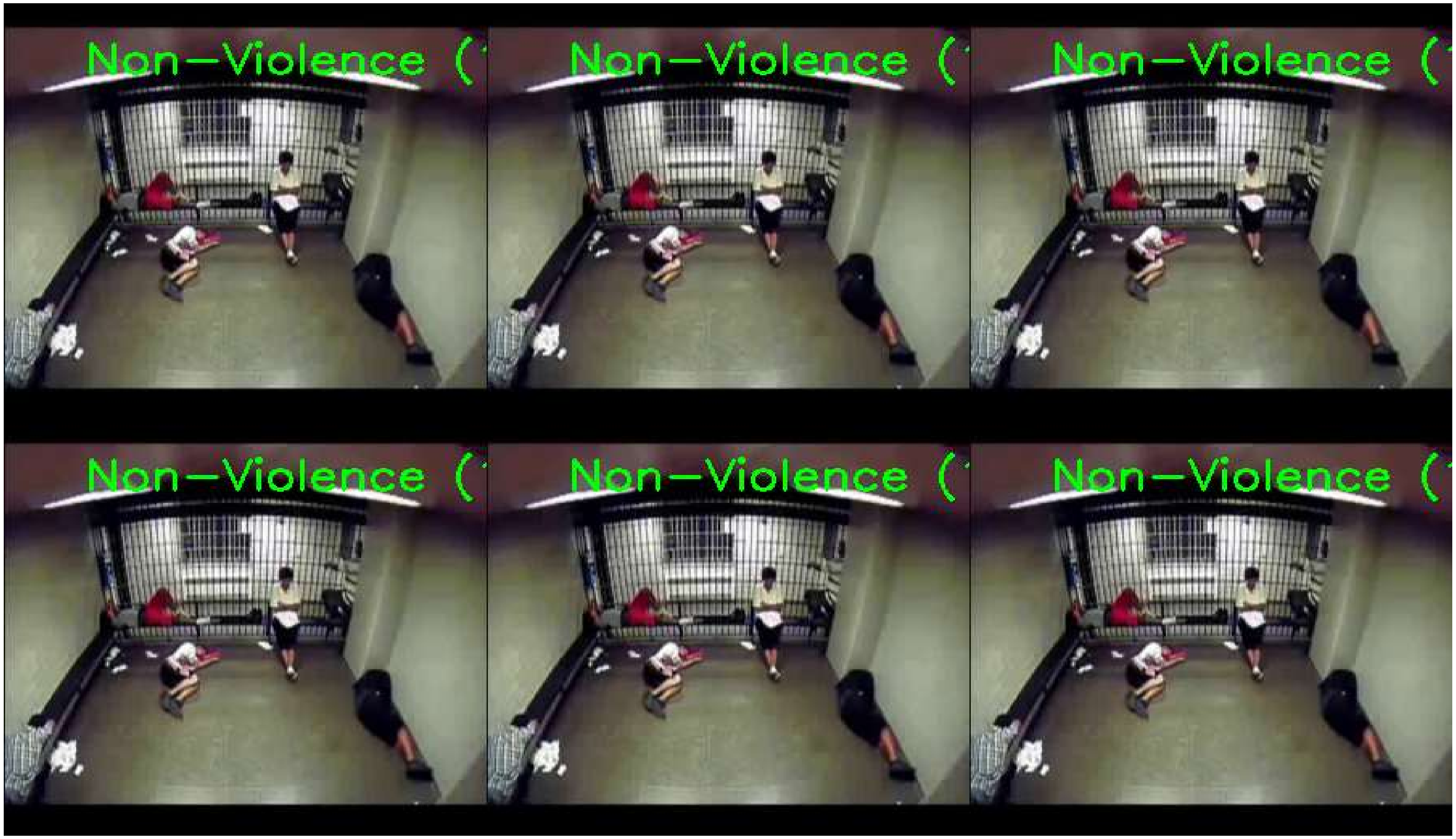
Training Performance Metrics Over Epochs

Epoch	Accuracy	Loss	Val Accuracy	Val Loss	Learning Rate	Time per Step
1	0.8194	0.3744	0.9805	0.0554	0.0010	445ms
2	0.9620	0.1071	0.9776	0.0568	0.0010	391ms
3	0.9740	0.0736	0.9145	0.2405	0.0010	398ms
4	0.9746	0.0724	0.9817	0.0559	0.0005	392ms
5	0.9899	0.0316	0.9881	0.0281	0.0005	391ms
6	0.9894	0.0306	0.9754	0.0511	0.0005	392ms
7	0.9881	0.0257	0.9864	0.0342	0.0005	392ms
8	0.9897	0.0253	0.9952	0.0155	0.0005	400ms
9	0.9934	0.0208	0.9920	0.0204	0.0005	391ms
10	0.9905	0.0281	0.9934	0.0169	0.0005	391ms

Predicted output



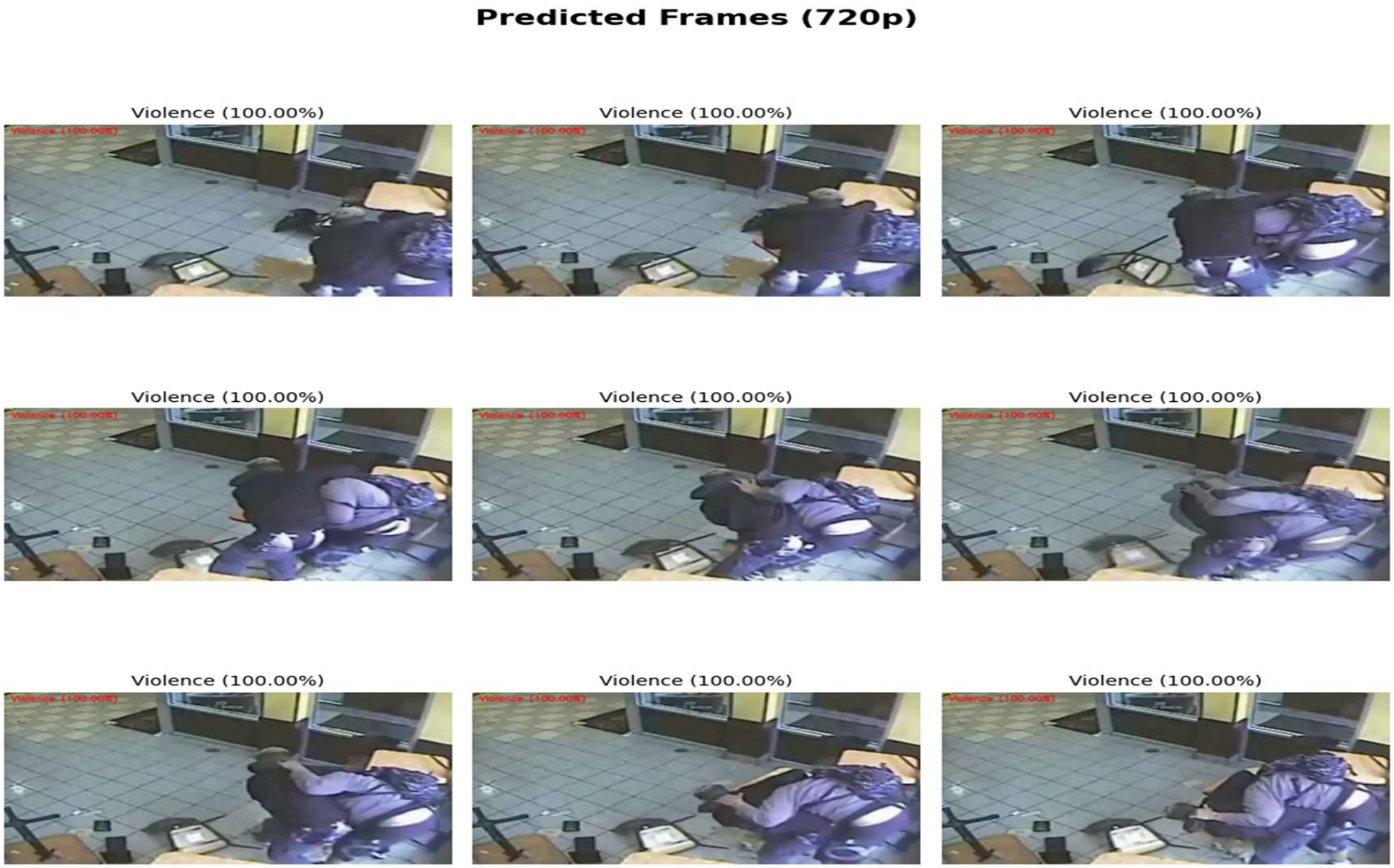
Predicted output



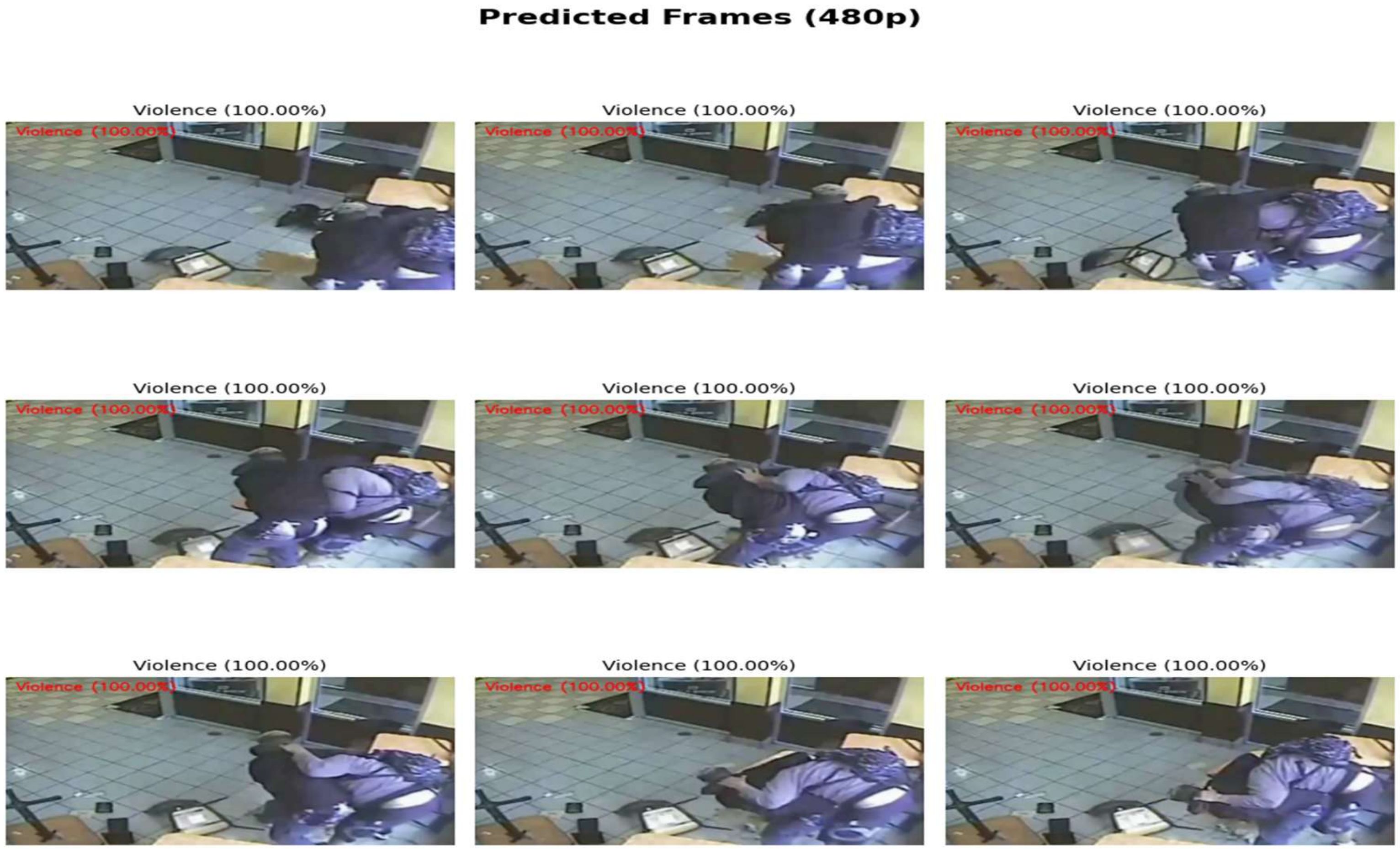
Experimental Setup in Different Frames

- Classified frames as **Violence/Non-Violence**.
- Resized to **720p, 480p, 360p**.
- Used **5-frame sequences** for prediction.
- Labeled frames **(Red/Green) with confidence**.
- Displayed **every 10th frame** in a **3x3 grid**.
- Used **Matplotlib for visualization**.

Experimental setup in different frames

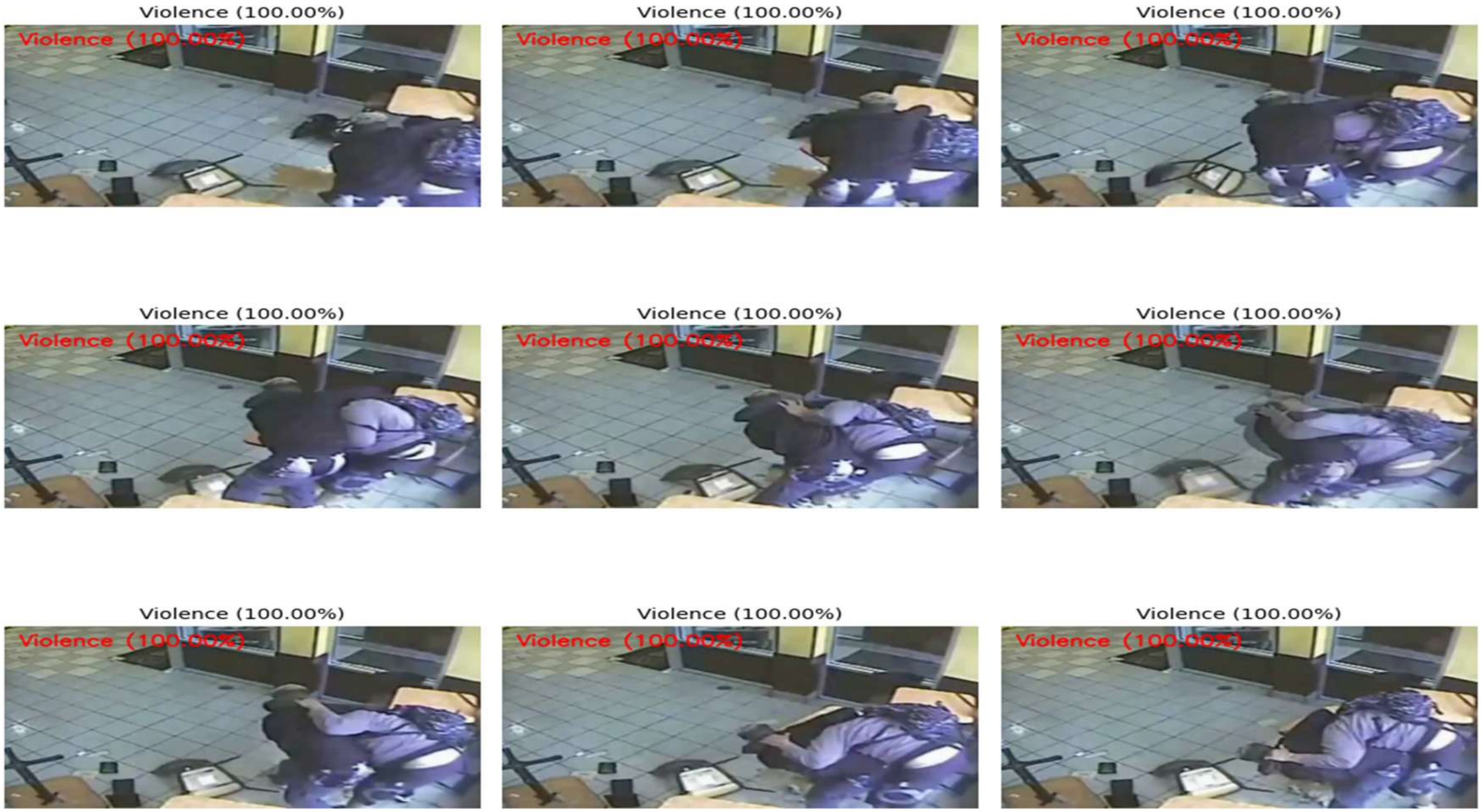


Experimental setup in different frames



Experimental setup in different frames

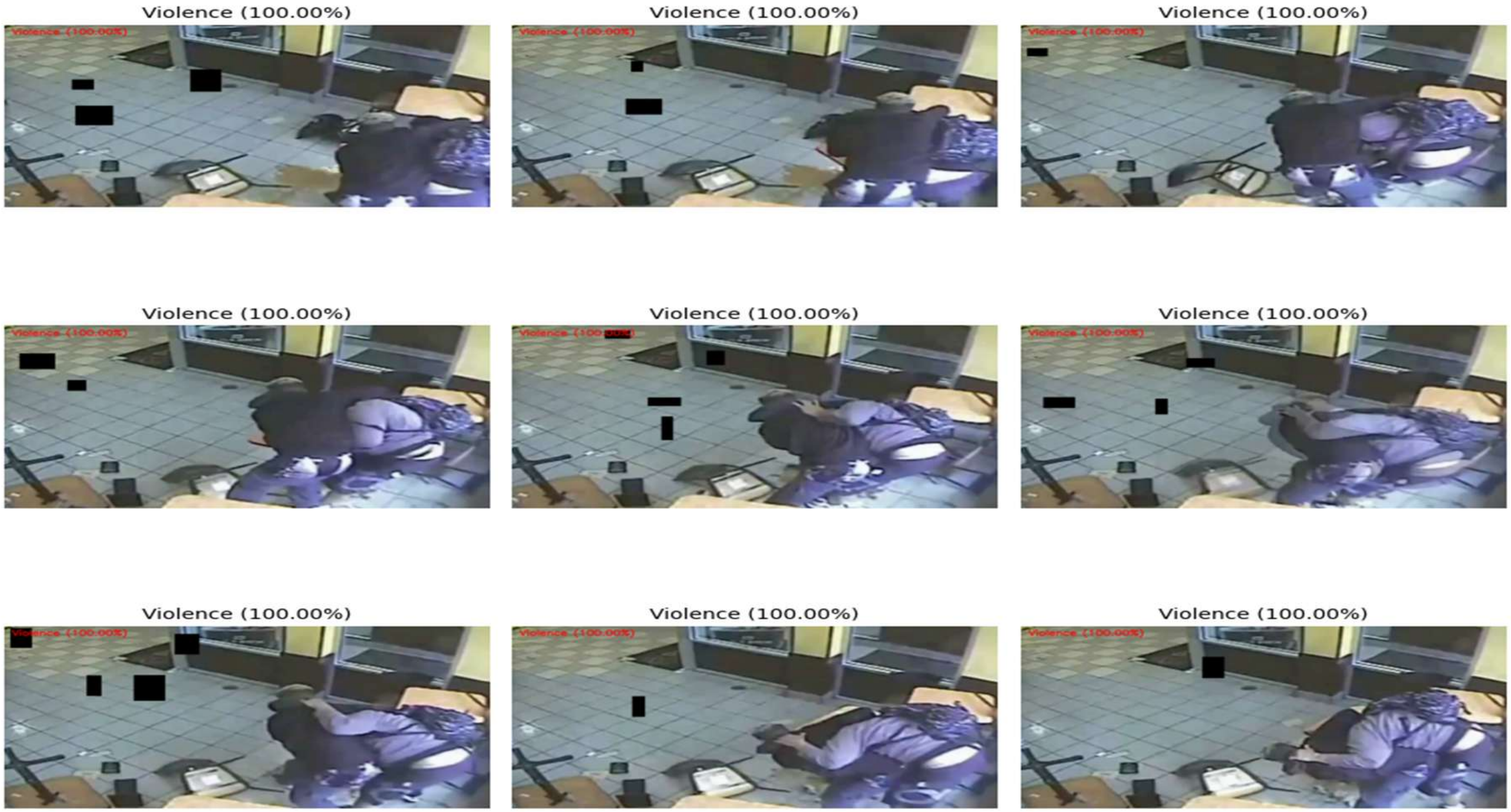
Predicted Frames (360p)



Occlusion Impact on Violence Detection

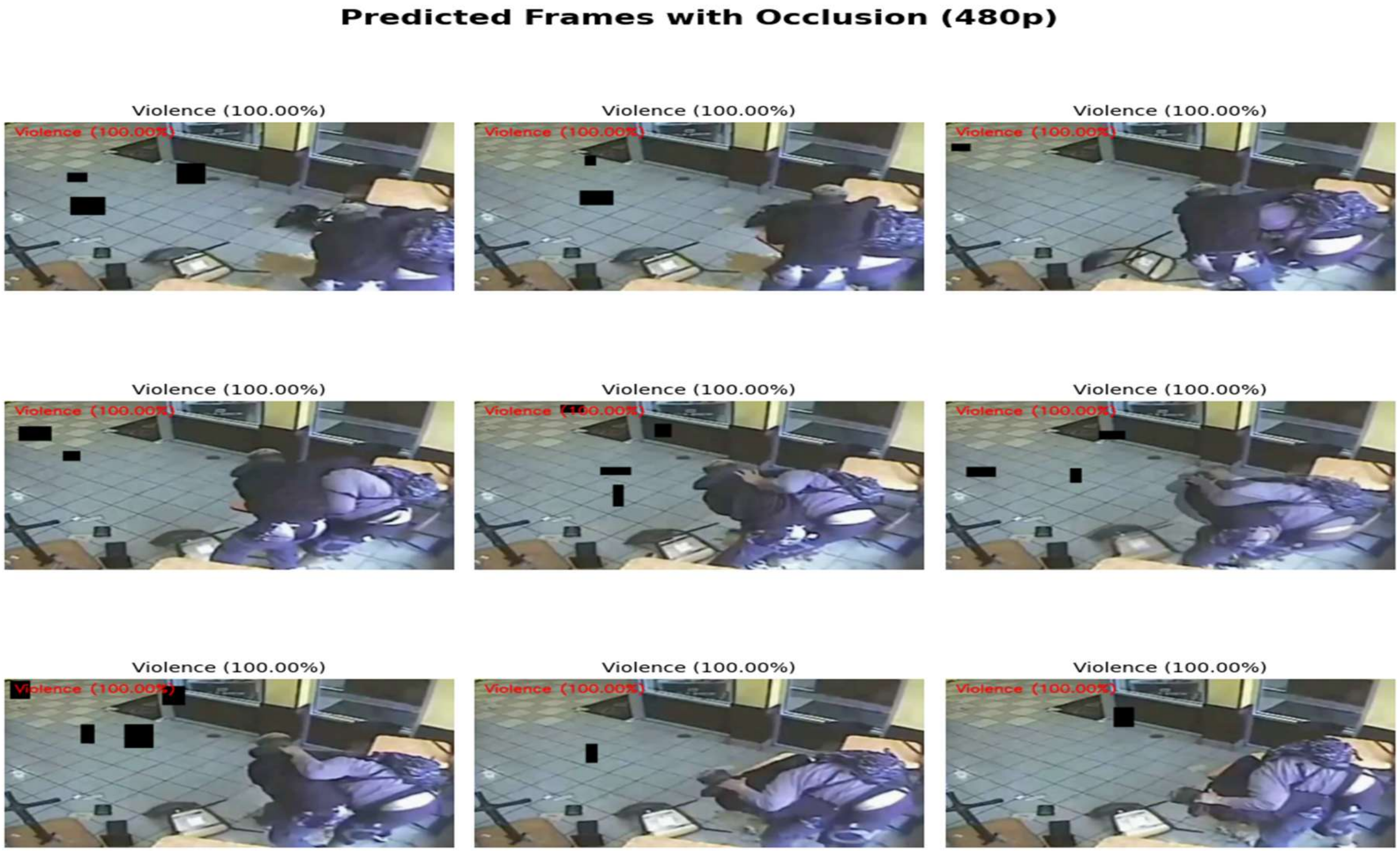
- **Violence detection** with occluded frames.
- **Random black patches** simulate blocked views.
- **Frames resized** to 720p, 480p, 360p.
- **5-frame sequences** used for prediction.
- **Labels with confidence scores** displayed.
- **3x3 grid visualization** for occlusion effects.

Occlusion Impact on Violence Detection

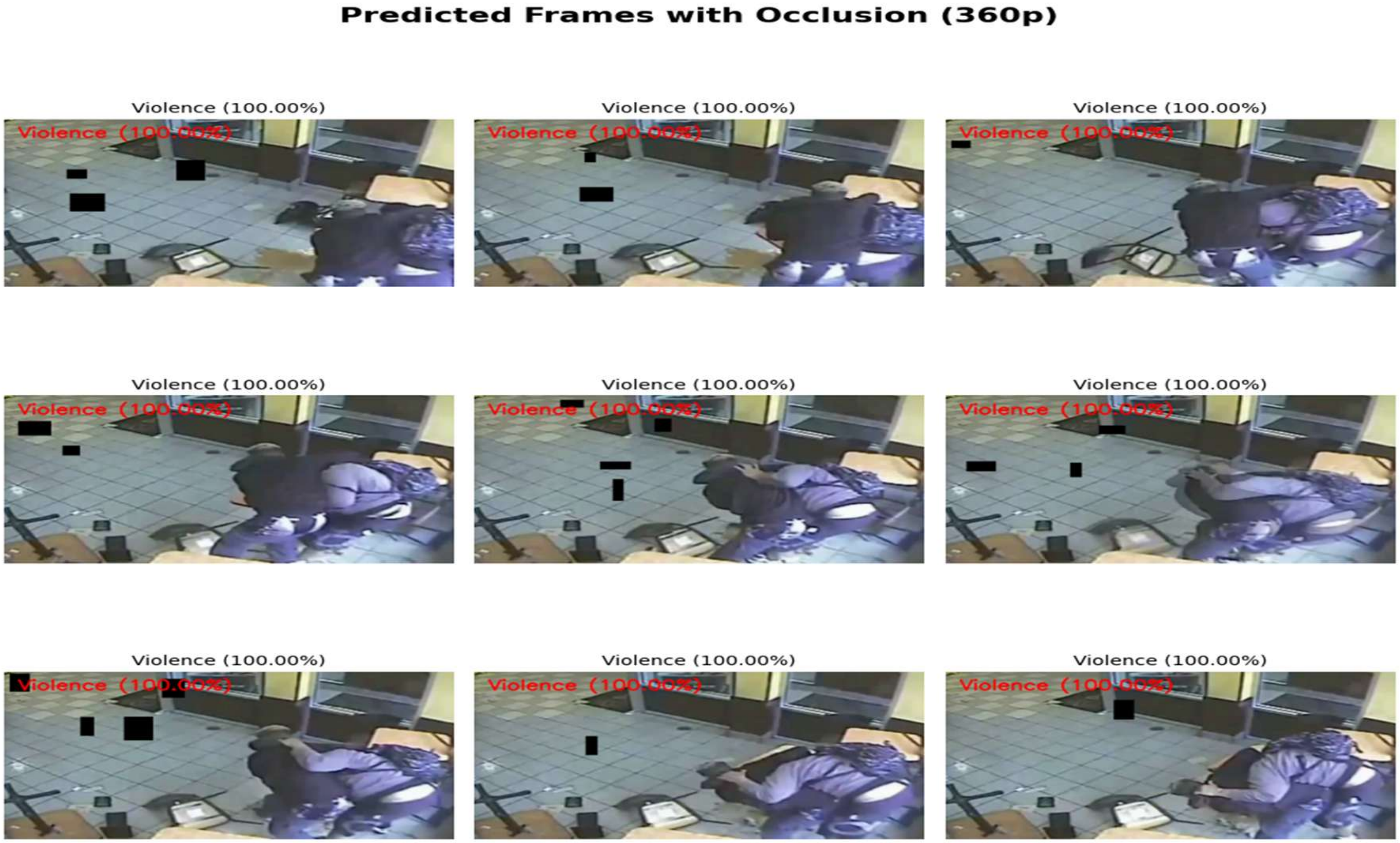


Predicted Frames with Occlusion (720p)

Occlusion Impact on Violence Detection



Occlusion Impact on Violence Detection

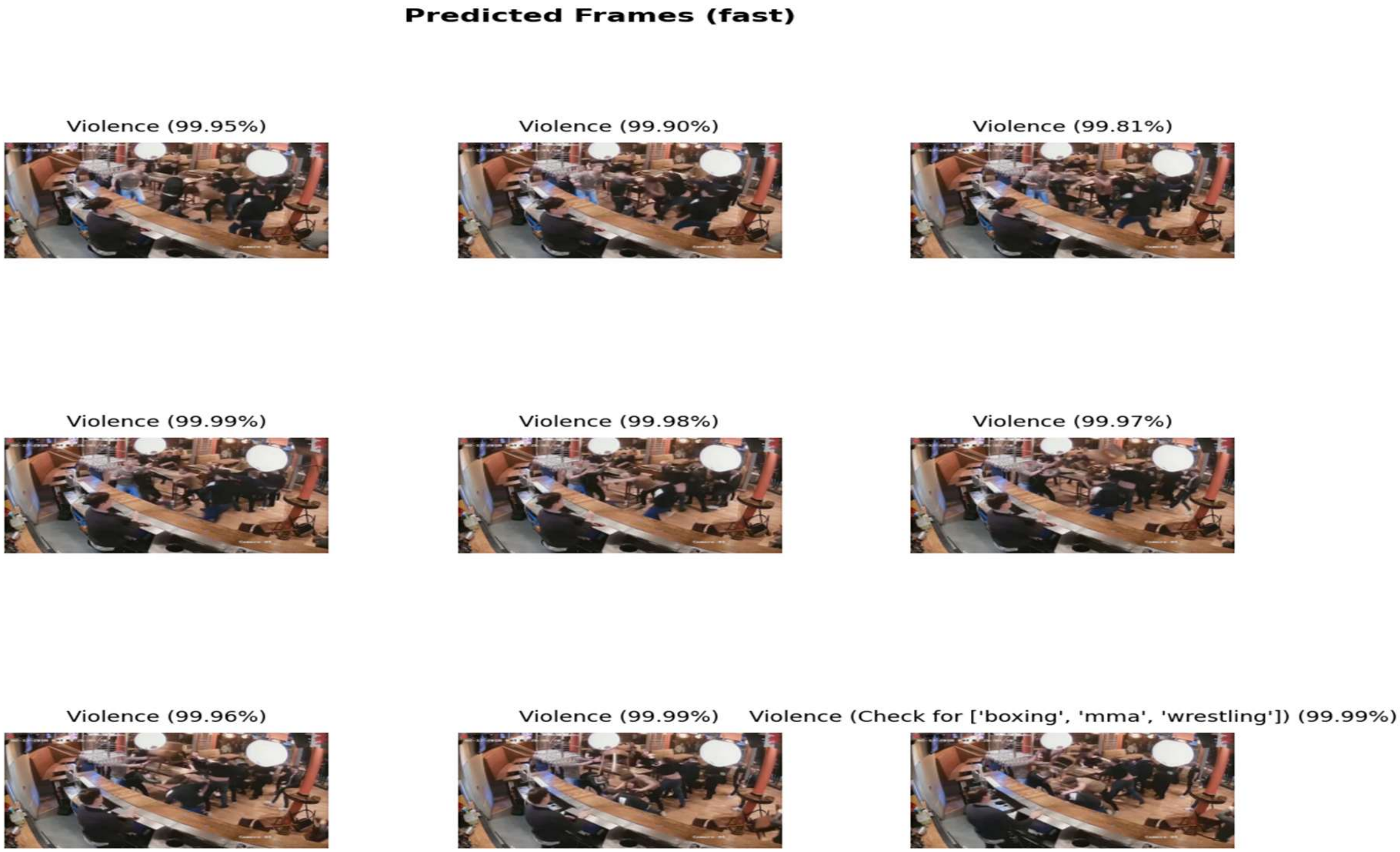


Predicted Frames with Occlusion (360p)

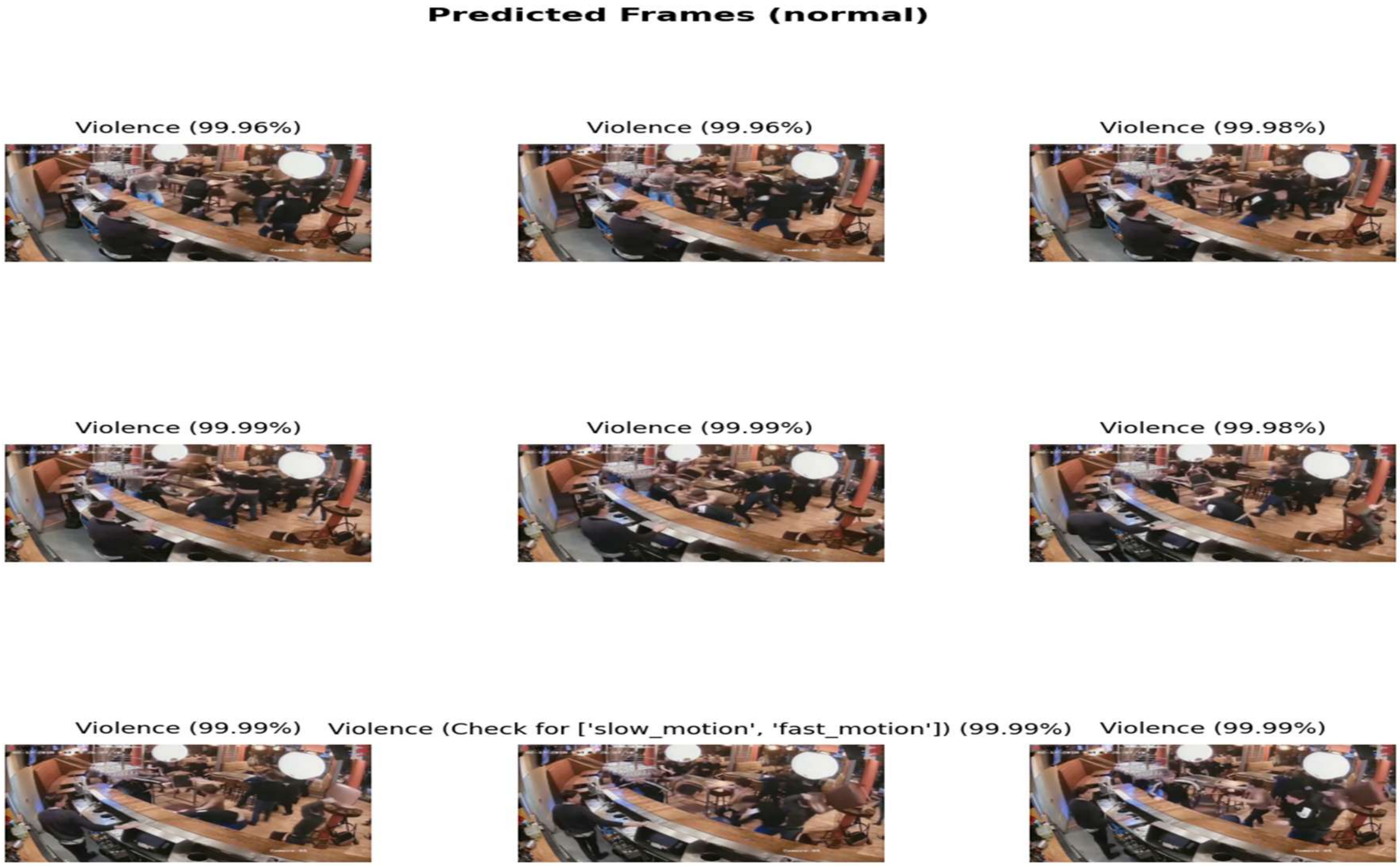
Adversarial Motion-Aware Violence Detection in Videos

- **Violence detection** with adversarial testing.
- **Frame rate adaptation** for different motion speeds.
- **5-frame sequences** used for prediction.
- **Random adversarial variations** (e.g., staged fights, sports).
- **Frames resized** to 32x32 for model compatibility.
- **Confidence scores displayed** with softmax predictions.
- **3x3 grid visualization** for better readability.

Adversarial Motion-Aware Violence Detection in Videos



Adversarial Motion-Aware Violence Detection in Videos



Predicted Frames (normal)

Violence (99.96%)

Violence (99.96%)

Violence (99.98%)

Violence (99.99%)

Violence (99.99%)

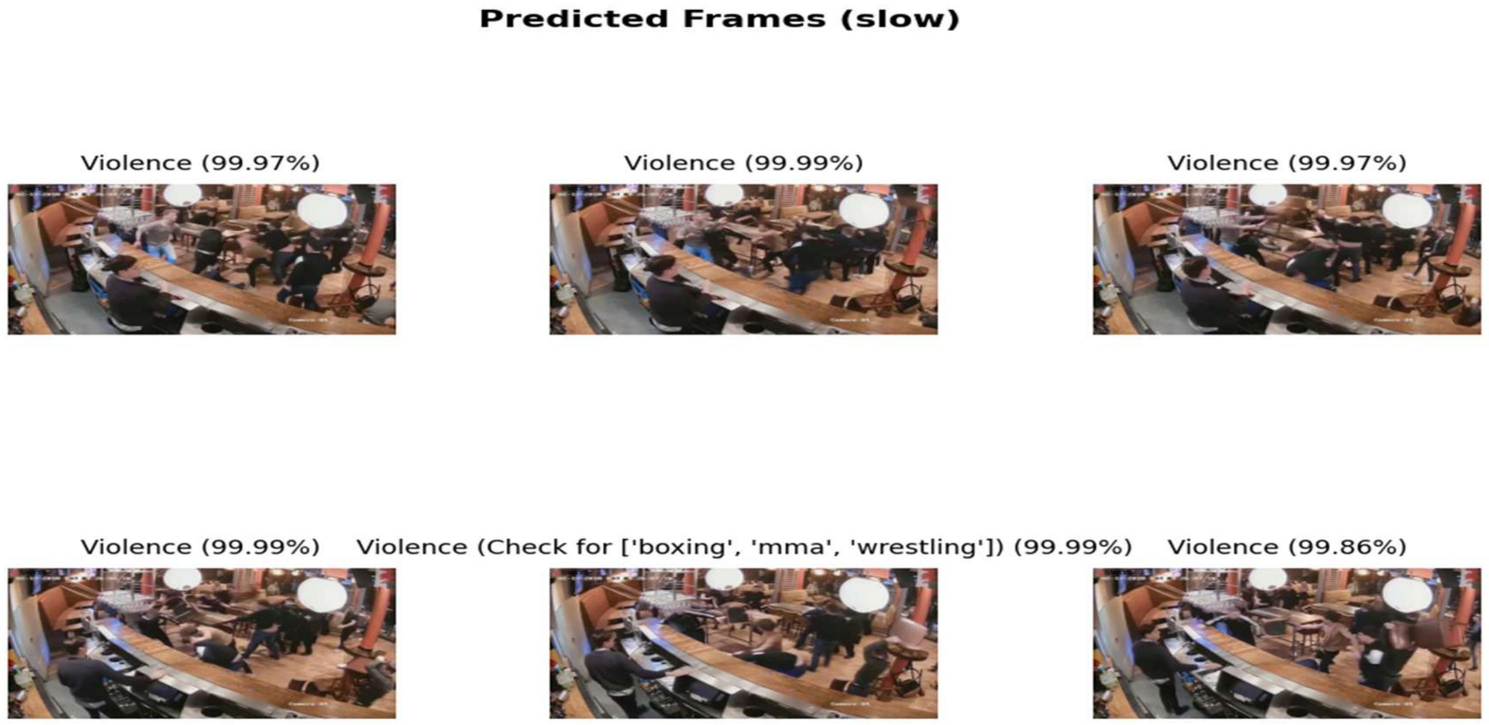
Violence (99.98%)

Violence (99.99%)

Violence (Check for ['slow_motion', 'fast_motion']) (99.99%)

Violence (99.99%)

Adversarial Motion-Aware Violence Detection in Videos



Predicted Frames (slow)

Violence (99.97%)

Violence (99.99%)

Violence (99.97%)

Violence (99.99%)

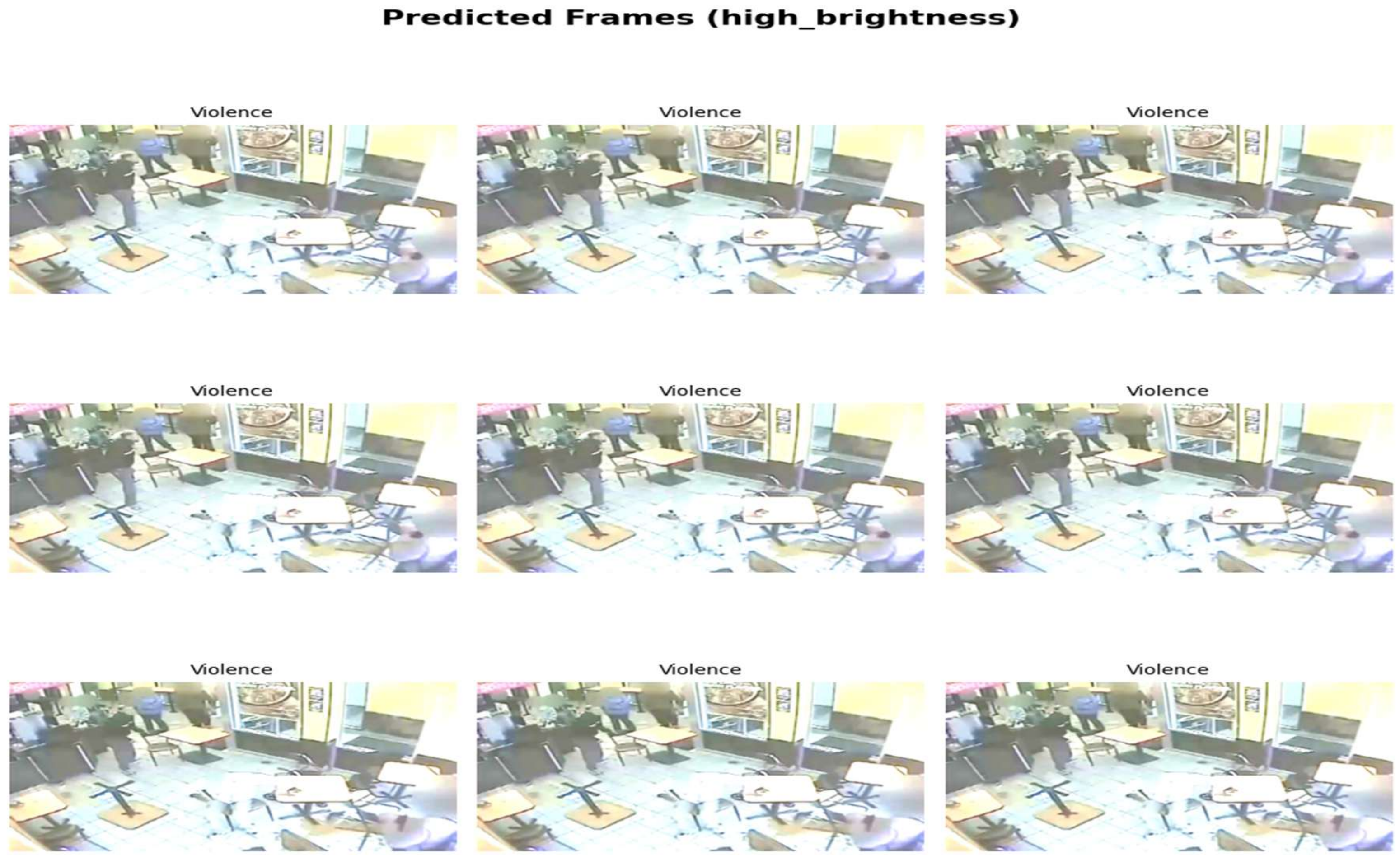
Violence (Check for ['boxing', 'mma', 'wrestling']) (99.99%)

Violence (99.86%)

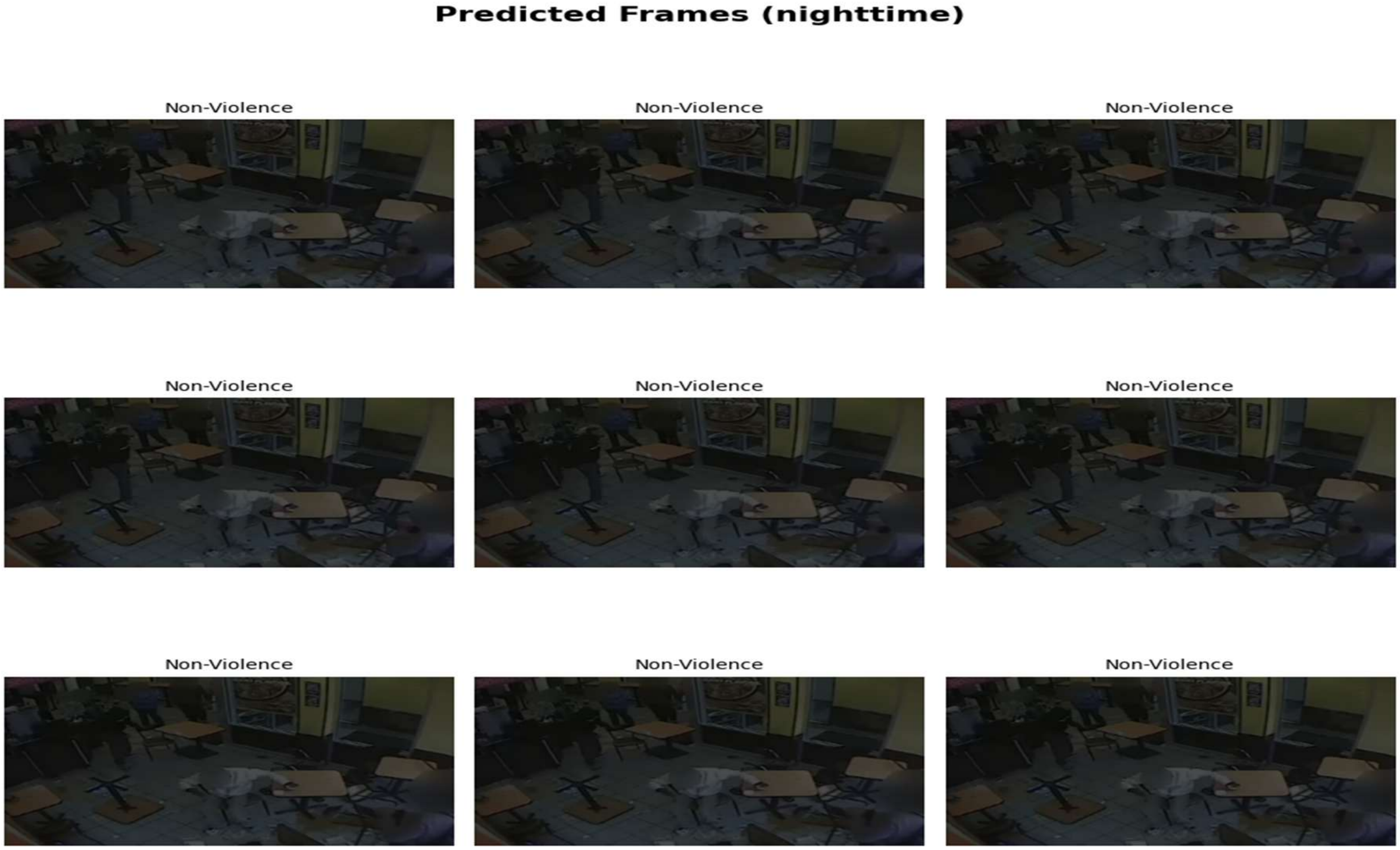
Lighting-Adaptive Violence Detection in Videos

- **Violence detection under lighting variations.**
- **Conditions tested**: brightness, contrast, blur, nighttime.
- **Frames resized to 1280x720.**
- **5-frame sequences for prediction.**
- **Deep learning model for classification.**
- **Confidence-based labels: Violence/Non-Violence.**
- **3x3 grid for visual comparison.**

Lighting-Adaptive Violence Detection in Videos

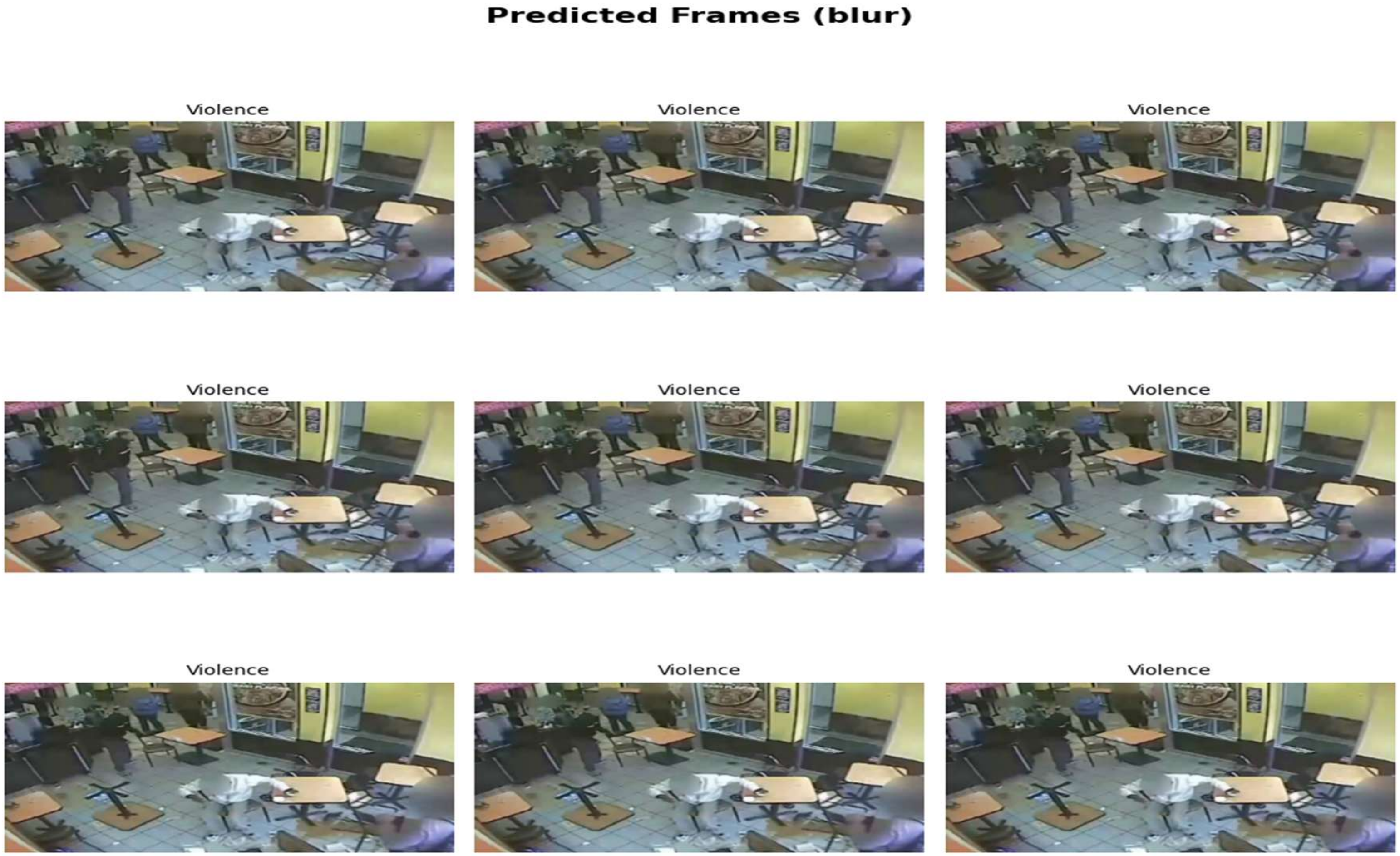


Lighting-Adaptive Violence Detection in Videos



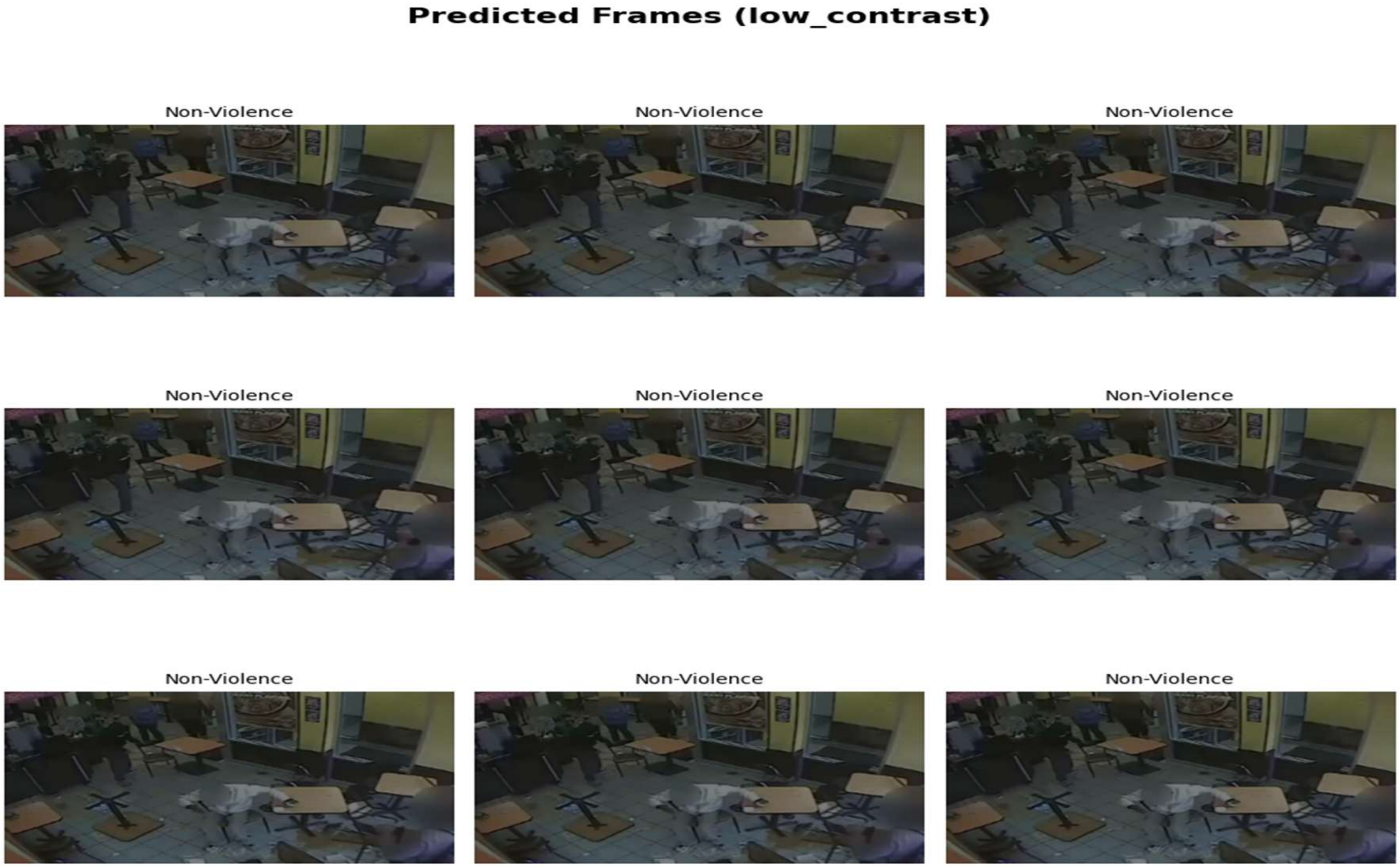
Predicted Frames (nighttime)

Lighting-Adaptive Violence Detection in Videos



Predicted Frames (blur)

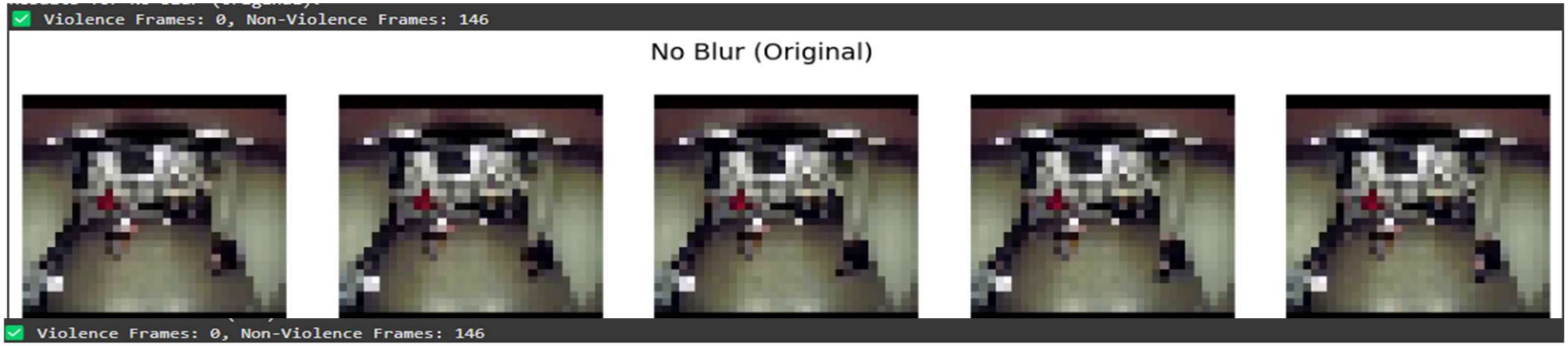
Lighting-Adaptive Violence Detection in Videos



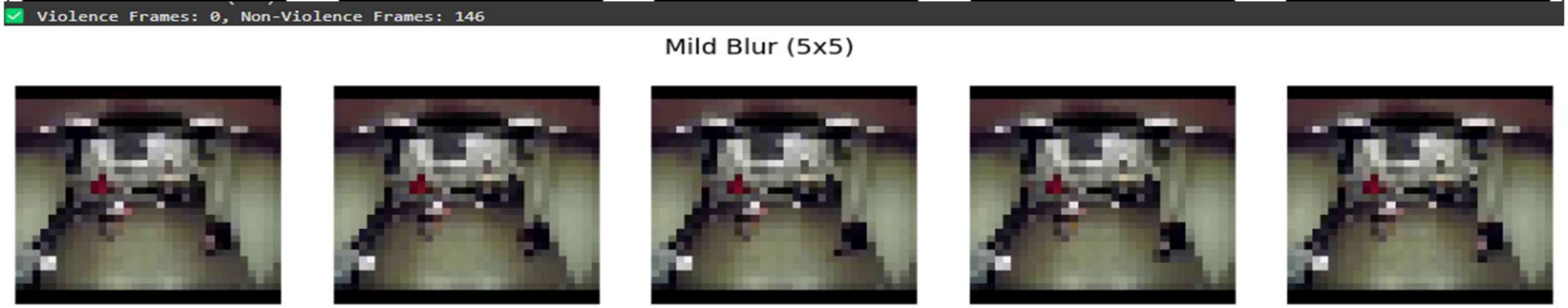
Violence Detection with Blur Effects

- **Violence detection with blur effects.**
- **Gaussian blur applied at 5x5 and 15x15 levels.**
- **Frames resized and normalized for model input.**
- **5-frame sequences used for prediction.**
- **Model classifies frames as Violence/Non-Violence.**
- **Counts of each class recorded for analysis.**
- **Sample frames displayed for visual comparison.**

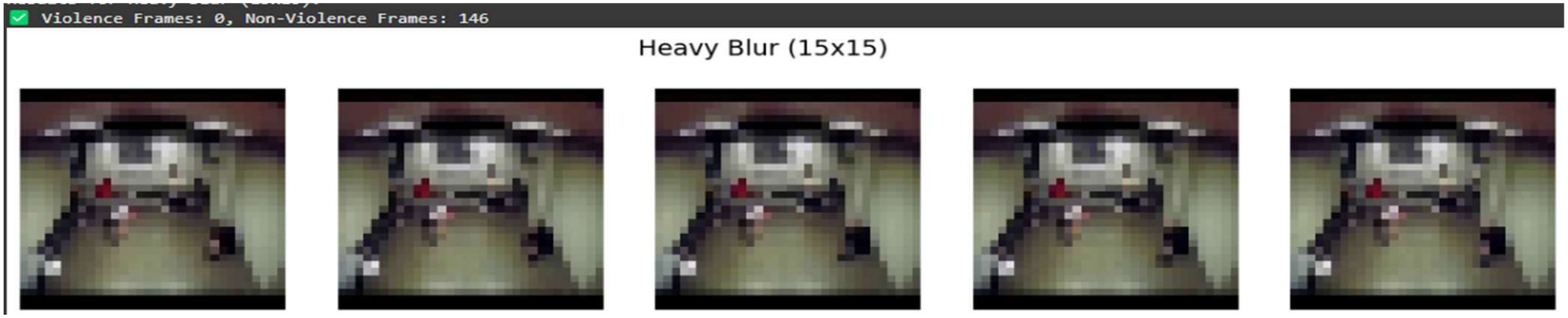
Violence Detection with Blur Effects



```
✅ Violence Frames: 0, Non-Violence Frames: 146
```



```
✅ Violence Frames: 0, Non-Violence Frames: 146
```



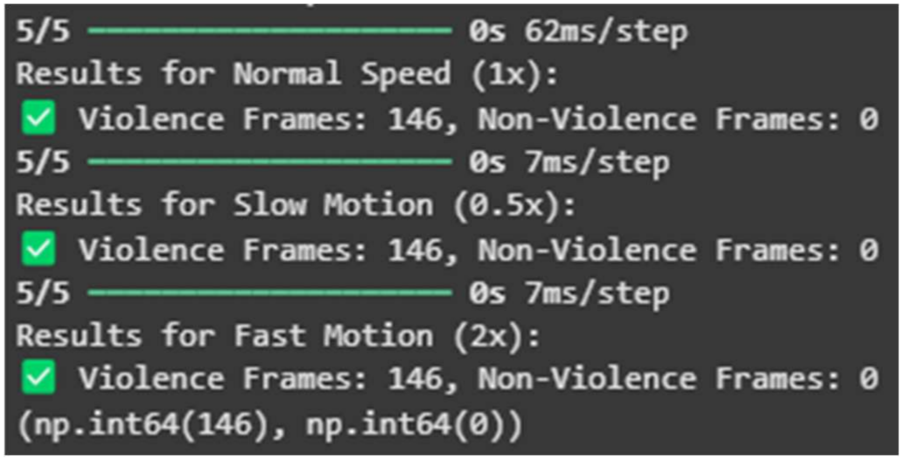
```
✅ Violence Frames: 0, Non-Violence Frames: 146
```

Speed-Adaptive Violence Detection in Videos

- **Violence detection with blur effects.**
- **Gaussian blur applied at 5×5 and 15×15 levels.**
- **Frames resized and normalized for model input.**
- **5-frame sequences used for prediction.**
- **Model classifies frames as Violence/Non-Violence.**
- **Counts of each class recorded for analysis.**
- **Predictions compared across different blur levels.**

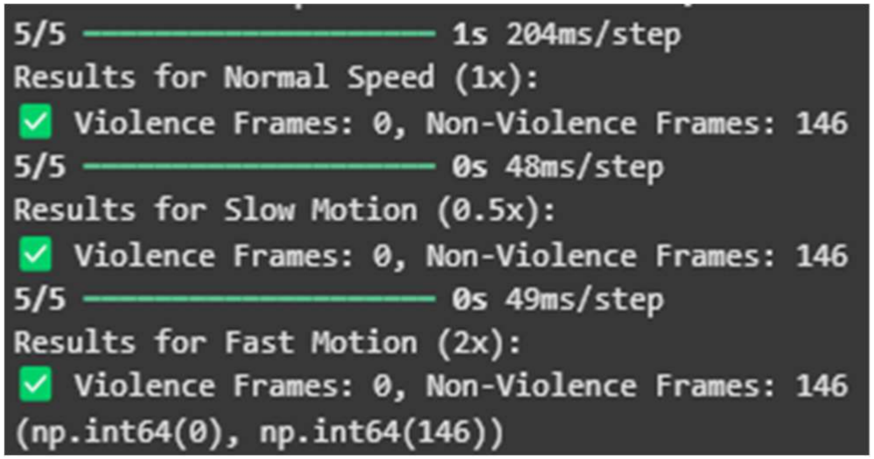
Speed-Adaptive Violence Detection in Videos

Violence



```
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 62ms/step
Results for Normal Speed (1x):
✅ Violence Frames: 146, Non-Violence Frames: 0
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step
Results for Slow Motion (0.5x):
✅ Violence Frames: 146, Non-Violence Frames: 0
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step
Results for Fast Motion (2x):
✅ Violence Frames: 146, Non-Violence Frames: 0
(np.int64(146), np.int64(0))
```

Non Violence

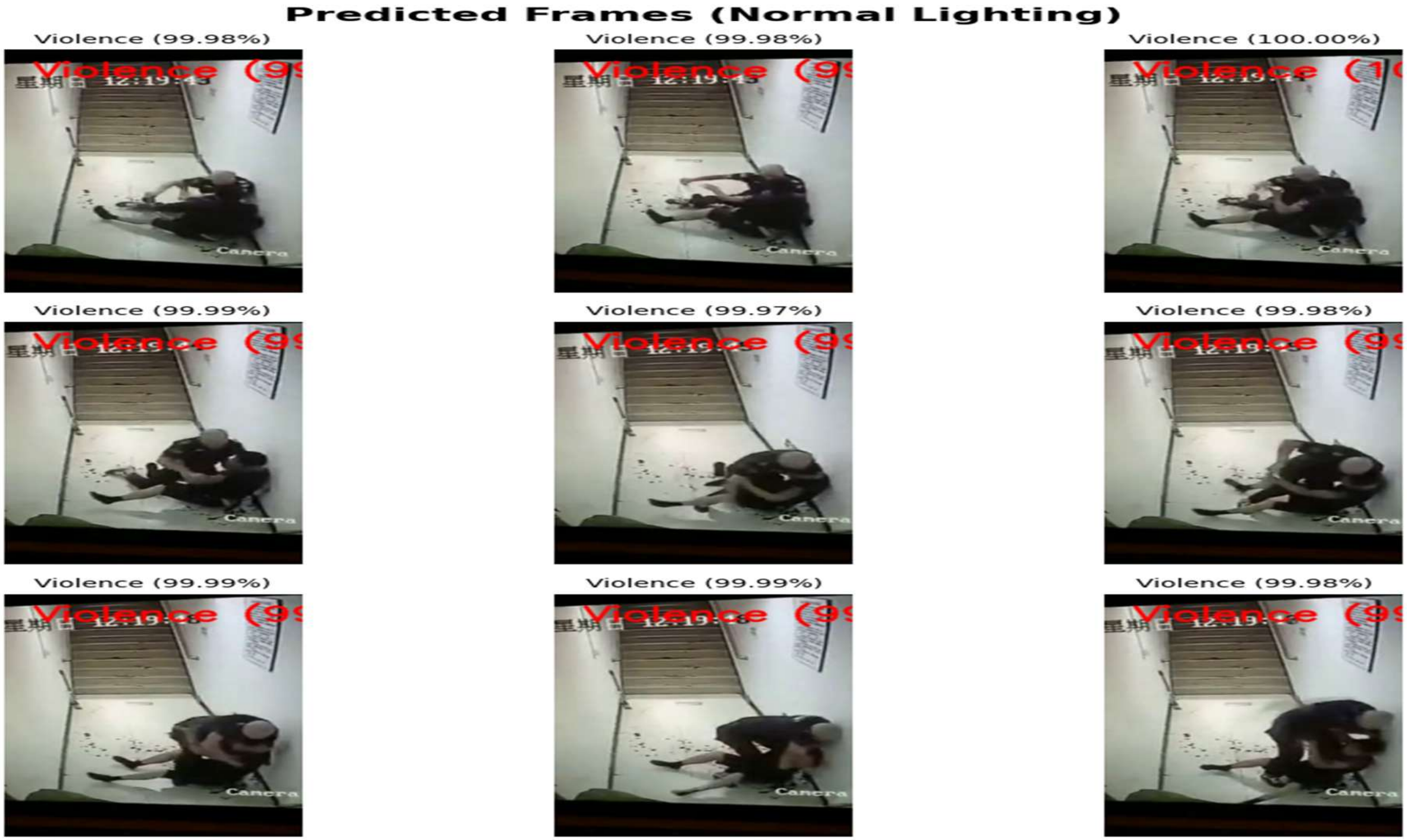


```
5/5 ━━━━━━━━━━━━━━━━━━━━ 1s 204ms/step
Results for Normal Speed (1x):
✅ Violence Frames: 0, Non-Violence Frames: 146
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 48ms/step
Results for Slow Motion (0.5x):
✅ Violence Frames: 0, Non-Violence Frames: 146
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 49ms/step
Results for Fast Motion (2x):
✅ Violence Frames: 0, Non-Violence Frames: 146
(np.int64(0), np.int64(146))
```

Violence Detection in Different Lighting

- **Violence detection under different lighting conditions.**
- **Lighting variations: Normal, Bright, and Dim.**
- **Frames resized, normalized, and processed in 5-frame sequences.**
- **Model predicts Violence/Non-Violence for each frame sequence.**
- **Overall prediction is based on averaged confidence scores.**
- **Selected frames displayed with labels for visual comparison.**
- **Analysis helps assess model robustness to lighting changes.**

Violence Detection in Different Lighting



Predicted Frames (Normal Lighting)

Violence (99.98%)

Violence (99.98%)

Violence (100.00%)

Violence (99.99%)

Violence (99.97%)

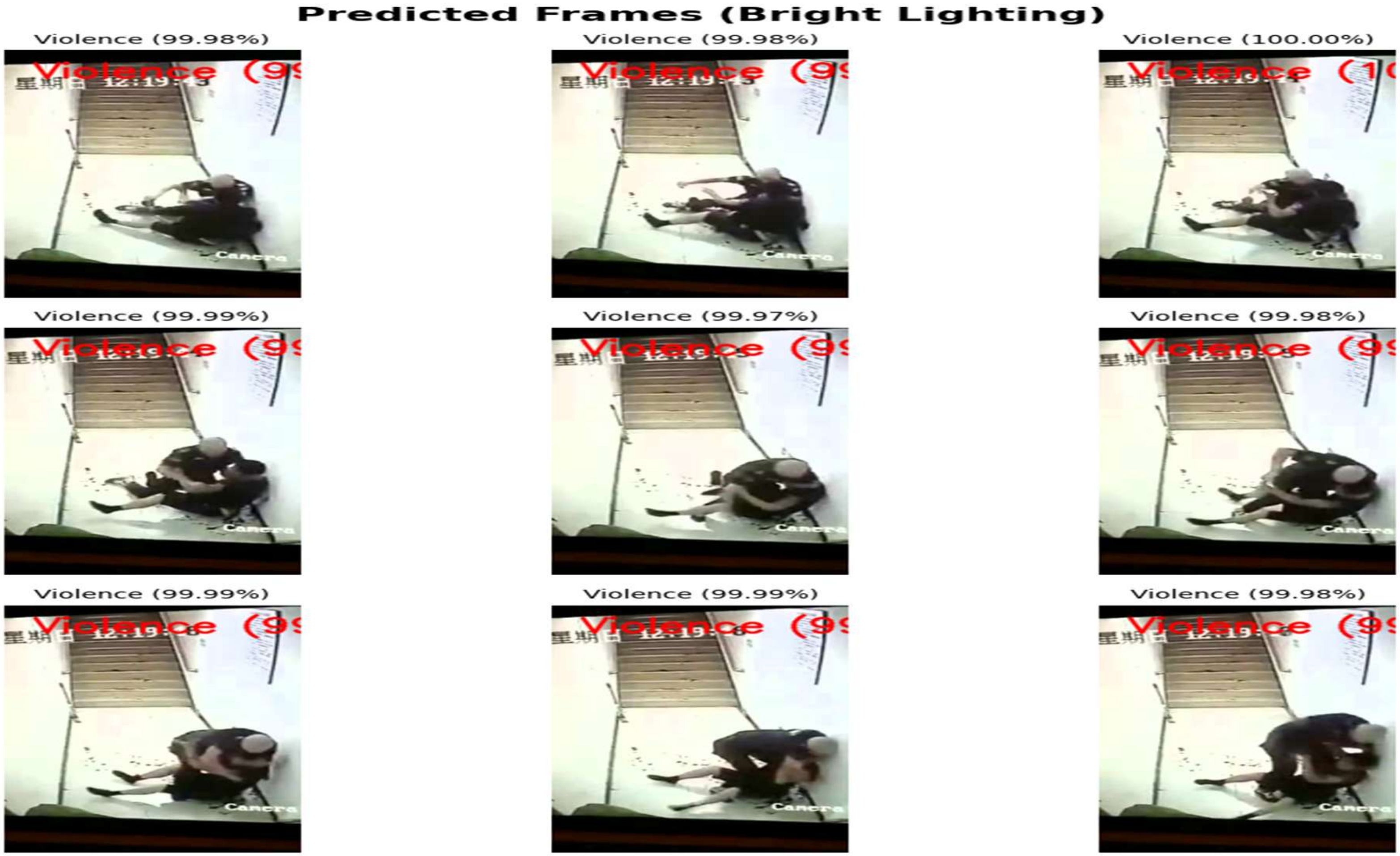
Violence (99.98%)

Violence (99.99%)

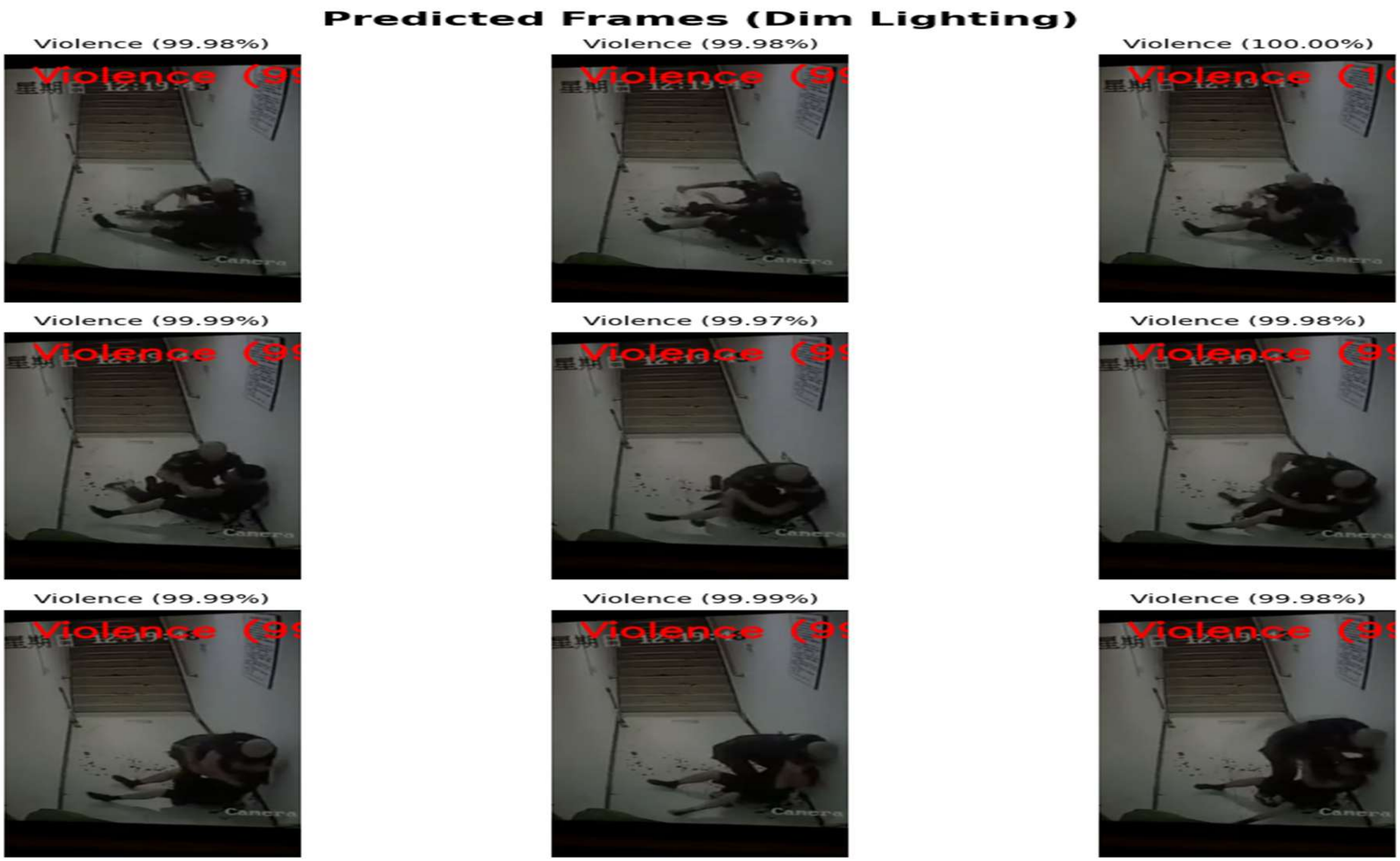
Violence (99.99%)

Violence (99.98%)

Violence Detection in Different Lighting



Violence Detection in Different Lighting



Predicted Frames (Dim Lighting)

Violence (99.98%)

Violence (99.98%)

Violence (100.00%)

Violence (99.99%)

Violence (99.97%)

Violence (99.98%)

Violence (99.99%)

Violence (99.99%)

Violence (99.98%)

Challenges & Limitations

- **Imbalanced dataset** → More "Non-Violence" than "Violence" samples.
- **Blurred & unclear frames** → Hard to classify aggressive actions.
- **Subtle movements** → Some violent actions are too quick.
- **Overfitting risk** → Model may memorize training data.

Conclusion

- **Improves security** and **reduces human error**.
- **Automates violence detection** for jails.
- **Provides real-time alerts** for quick response.
- **Can be integrated** with existing CCTV systems.

Future Scope

- **Improve dataset size** for better model generalization.
- **Enhance real-time processing** speed.
- **Expand to other security applications** (schools, public places).
- **Use multimodal AI** (audio + video detection).

References

- **Ashwin Saji Kumar et al.**, "Violence Detection in Jails and Mental Asylums," *IJISRT*, 2024.
- **Marjam V. Smeekens et al.**, "Risk Screener Violence (RS-V): A Predictive Model for Violent Behavior," *Frontiers in Psychology*, 2023.
- **Nancy G. La Vigne et al.**, "Crime Prevention in Jail: A Surveillance Monitoring Approach," *Urban Institute*, 2022.
- **RWF-2000 Dataset**: Real-world Fighting Dataset for Violence Detection.
- **UCF-Crime Dataset**: A Large-Scale Benchmark for Anomaly Detection in Surveillance Videos.
- **Hockey Fight Dataset**: A Video-Based Dataset for Violence Recognition in Sports.

References

- **TensorFlow Documentation**: https://www.tensorflow.org/
- **OpenCV Documentation**: https://docs.opencv.org/
- **Google Colab**: https://colab.research.google.com/ (Used for model training and evaluation).

Thank You & Queries

- Thank you for your attention.
- Open to any questions.