Crime Scene Analytics using Deep Learning

Submitted By

ManjurMahamad Kovadiya

24MCD008

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

December 2025

Crime Scene Analytics using Deep Learning

Major Project - I

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering (Data Science)

Submitted By

ManjurMahamad Kovadiya

(24MCD008)

Guided By

Dr. Sharada Valiveti

NIRMA UNIVERSITY

INSTITUTE OF TECHNOLOGY

NAAC ACCREDITED 'A+' GRADE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

December 2025

Certificate

This is to certify that the Major Project Part 1 entitled **"Crime Scene Analytics using Deep Learning"** submitted by **ManjurMahamad Kovadiya (Roll No: 24MCD008)**, towards the partial fulfillment of the requirements for the award of degree of **Master of Technology in Computer Science and Engineering (Data Science)** of Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project part 1, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. Sharada Valiveti
Guide & Associate Professor,
Dept. of Computer Science & Engineering,
School of Technology,
Nirma University,
Ahmedabad.

Dr. Sudeep Tanwar
Professor and Head of Department
Dept. of Computer Science & Engineering,
School of Technology,
Nirma University,
Ahmedabad.

Acknowledgement

I would like to take this opportunity to sincerely thank everyone who has helped and supported me throughout the journey of working on my project, Crime Scene Analytics using Deep Learning. First and foremost, I am truly grateful to my project guide,**Dr. Sharada Valiveti** for their valuable guidance, constant support, and encouragement. Their knowledge and insights have been very helpful in shaping the project in the right direction. Lastly, I would like to thank all the faculty members who, directly or indirectly, influenced my learning and inspired me to keep improving. This project wouldn't have been possible without the support of all these wonderful people, and I am truly thankful to each one of them. Last but not least, my humble thanks to almighty God.

ManjurMahamad Kovadiya
24MCD008

Abstract

This work presents an end-to-end deep learning pipeline for automated crime-scene object detection using the YOLOv8 framework. The system is designed to assist forensic investigators by rapidly identifying critical visual evidence including weapons (guns, knives, rifles, blunt objects) and blood stains/splatter patterns in complex and often low-quality crime-scene imagery.

Two publicly available datasets from Roboflow Universe (Weapons: 1,035 images; Bloodstain: 326 images) were merged into a unified YOLO-format dataset. To address class imbalance and limited data diversity, a DCGAN-based generative adversarial network was employed to synthesize additional realistic weapon images. The combined dataset underwent extensive preprocessing and enhancement: noise reduction (Gaussian blur, median filtering, non-local means denoising), super-resolution of small/distant objects using Real-ESRGAN, and aggressive data augmentation techniques (Mosaic, MixUp, copy-paste insertion of weapons into scenes, random fog/shadows/occlusion, and color jitter) to better simulate real-world crime-scene conditions.

A custom class remapping and merging script was developed to seamlessly integrate the datasets while preserving label integrity and generating a balanced train/validation/test split. The resulting enhanced dataset was used to train multiple YOLOv8 models (nano, small, medium variants). Experimental results demonstrate significant improvements in detection accuracy particularly for small, partially occluded, or low-contrast objects compared to baseline training on raw datasets.

The final system achieves robust real-time inference capability and serves as a practical forensic tool for accelerating evidence discovery, reducing manual analysis time, and supporting preliminary crime-scene investigation in digital forensics workflows.

Keywords: Crime Scene Analysis, Object Detection, YOLOv8, Deep Learning, Data Augmentation, GAN-based Synthesis, Super-Resolution, Real-ESRGAN, Forensic Image Processing

Abbreviations

YOLO	You Only Look Once
YOLOv8	You Only Look Once version 8 (Ultralytics)
GAN	Generative Adversarial Network
DCGAN	Deep Convolutional Generative Adversarial Network
Real-ESRGAN	Realistic and Efficient Super-Resolution Generative Adversarial Network
mAP	mean Average Precision
mAP@50	mean Average Precision at IoU threshold 0.50
mAP@50:95	mean Average Precision averaged over IoU thresholds 0.50 to 0.95
CNN	Convolutional Neural Network
IoU	Intersection over Union
FPS	Frames Per Second
GPU	Graphics Processing Unit
CPU	Central Processing Unit
RGB	Red–Green–Blue (color format)
API	Application Programming Interface
Colab	Google Colaboratory
AI	Artificial Intelligence
DL	Deep Learning
COCO	Common Objects in Context (dataset format)
ROI	Region of Interest
NMS	Non-Maximum Suppression
BBox	Bounding Box

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Crime scene investigation heavily relies on the accurate identification and documentation of physical evidence from photographs taken at the scene. Among the most critical forms of evidence are **weapons** (guns, knives, rifles, blunt objects) and **blood stains or splatter patterns**, as they provide vital clues about the nature, sequence, and severity of criminal events.

Traditionally, forensic experts manually examine hundreds or thousands of crime-scene images process that is time-consuming, prone to human error, fatigue-induced oversight, and inconsistent across examiners. This is particularly challenging when objects are small, distant, partially occluded, poorly lit, or degraded due to environmental factors.

Recent advances in deep learning-based object detection, especially the YOLO (You Only Look Once) family of models, have achieved remarkable real-time performance and accuracy. The latest **YOLOv8** by Ultralytics represents the current state-of-the-art in single-stage object detection. When supported by high-quality training data and advanced preprocessing, such models hold immense potential to automate and significantly accelerate forensic image analysis.

This project develops a complete, reproducible, end-to-end deep learning pipeline for automated detection of **weapons** and **blood stains** in crime-scene photographs using YOLOv8, with special emphasis on robust dataset preparation and enhancement techniques tailored to forensic imaging challenges.

1.2 Problem Statement

Manual inspection of crime-scene images is slow, inconsistent, and often misses critical evidence especially small or low-contrast objects such as distant weapons, tiny knives, blood droplets, or faint splatter patterns in cluttered, noisy, or low-light conditions. There is a pressing need for an automated, reliable, and real-time-capable system that assists forensic investigators by accurately detecting and localizing these high-priority evidence classes.

1.3 Aim of the Project

To design and implement an automated crime scene analysis system using **YOLOv8** that detects and classifies two forensically critical object classes like **weapons** and **blood stains** — in digital crime-scene images with high accuracy and real-time inference capability.

1.4 Objectives

- To merge two high-quality public datasets from Roboflow Universe:
  - Weapons Dataset (1,035 images) – containing guns, knives, rifles, pistols
  - Bloodstain Dataset (326 images) – containing blood stains and splatter patterns
- To develop a custom Python pipeline for dataset merging, class remapping, label validation, corruption detection, and balanced train/validation/test split generation.
- To enhance image quality using noise reduction techniques (Gaussian blur, median filtering, non-local means denoising).
- To improve detection of small and distant objects using **Real-ESRGAN** super-resolution enhancement.
- To optionally generate synthetic weapon images using **DCGAN** to address class imbalance.

- To apply advanced data augmentation techniques (Mosaic, MixUp, copy-paste, fog, shadows, occlusion, color jitter) to simulate realistic crime-scene variations.

- To train multiple variants of **YOLOv8** (nano, small, medium) on the final enhanced merged dataset.

- To evaluate model performance using standard metrics: mAP@50, mAP@50:95, precision, recall, and confusion matrix.

- To provide a fully functional, well-documented Google Colab notebook enabling real-time inference and bounding box visualization on unseen crime-scene images.

1.5 Scope and Limitations

- Focuses exclusively on **static 2D crime-scene photographs**.

- Detects only two classes: **weapon** and **blood**.

- Does not include video processing, human detection, activity recognition, fingerprint analysis, 3D reconstruction, or automated report generation.

- Implemented and tested in **Google Colab** environment (not deployed on edge/IoT devices).

- Serves as a **proof-of-concept forensic assistance tool**, not a replacement for certified forensic examination.

1.6 Significance of the Work

- First project to systematically merge and enhance Roboflow-based forensic datasets specifically for crime-scene object detection.

- Demonstrates significant real-world performance gains through Real-ESRGAN and advanced augmentation especially for small/distant evidence.

- Lays a strong foundation for future multi-class forensic AI systems.

Chapter 2

Literature Review

The integration of artificial intelligence (AI) and deep learning into forensic science has emerged as a transformative force, particularly in crime scene analysis. Traditional methods, reliant on manual evidence collection and expert judgment, often suffer from subjectivity, time constraints, and limitations in handling low-quality or complex imagery. This chapter reviews key advancements in AI-driven approaches for object detection, evidence enhancement, and real-time surveillance, highlighting gaps that the proposed YOLOv8-based system addresses through dataset merging, super-resolution, and augmentation tailored to weapons and bloodstain detection.

In the research paper "SPOT-CRIME: Suspicious Person and Object Tracking System for Real-Time Crime Monitoring" (2025), the authors introduce a three-phase monitoring framework utilizing YOLO for initial person detection, followed by object identification and Track-RCNN with SoftMax Gradient Layer-wise Relevance Propagation (SGLRP) for tracking in cluttered environments. The model achieves 98.54% detection accuracy on benchmark datasets like MOT17 and MOT20, outperforming baselines such as Seven-layered CNN by up to 8.66%. While effective for video-based surveillance, the approach focuses on dynamic tracking rather than static forensic imagery, leaving room for specialized preprocessing in low-resolution crime scene photos [1].

The invited review "Review of Advances in Small Object Detection Technology based on Deep Learning" (2024) by Liu et al. provides a comprehensive analysis of deep learning strategies for detecting small targets, categorizing techniques into data augmentation, super-resolution, multi-scale feature fusion, contextual learning, and anchor-free methods. Evaluated on datasets like MS COCO and DOTA, these approaches mitigate challenges such as information loss and low signal-to-noise ratios, with super-resolution models like Real-ESRGAN showing notable gains in visibility. This work underscores the need for hybrid enhancements in forensic contexts, where small weapons or bloodstains are often occluded or distant, directly informing the Real-ESRGAN integration in the proposed pipeline [2].

In “Generative AI Powered Forensic Device” (2025), the authors propose an AI-enhanced device leveraging generative adversarial networks (GANs) for synthetic evidence reconstruction and automated analysis, addressing manual forensic inefficiencies in digital data extraction. The system automates artifact detection and report generation, reducing processing time while maintaining chain-of-custody integrity. Tested on simulated cybercrime datasets, it demonstrates superior error reduction compared to traditional tools. However, its focus on device-based forensics overlooks image-specific augmentation, motivating the DCGAN-based synthetic weapon generation in this project to combat class imbalance [3].

The paper “Crime Scene Reconstruction Using 3D Convolutional Neural Networks (3D-CNNs)” (2025) explores 3D-CNNs for volumetric reconstruction of crime scenes from multi-view images, enabling immersive visualization and evidence localization. By processing spatial-temporal data, the model improves forensic accuracy in trajectory estimation, outperforming 2D baselines in simulated scenarios. Limitations include high computational demands for real-time use, highlighting the value of efficient 2D detection frameworks like YOLOv8 for initial evidence screening prior to 3D modeling [4].

In “The Artificial Intelligence in Autopsy and Crime Scene Analysis” (2024), Sacco et al. survey AI applications across forensic domains, including machine learning for post-mortem interval estimation, deep learning for Virtopsy radiological analysis, and neural networks for ballistic damage assessment. Covering over 100 studies, the review identifies AI’s potential to reduce human error in evidence interpretation, with robotics aiding non-invasive autopsies. Gaps in scalable, real-time crime scene tools for object localization are noted, aligning with this project’s emphasis on YOLOv8 for automated weapon and bloodstain detection [5].

The research “A Real-Time Crime Scene Intelligent Video Surveillance System in Violence Detection Framework using Deep Learning Techniques” (2022) presents a spatio-temporal feature extraction method combined with Deep Reinforcement Neural Networks

(DRNN) for violence anomaly detection in surveillance footage. Achieving 98% accuracy on the UCF-Crime dataset, the framework processes video frames to classify aggressive behaviors, surpassing CNN-BiLSTM baselines. While robust for dynamic scenes, it underperforms on static forensic images with low contrast, justifying the noise reduction and augmentation strategies employed here [6].

Finally, in "A Forensic Video Upscaling, Colorizing and Denoising Framework for Crime Scene Investigation" (2024), Prema and Anita develop an AI pipeline integrating GAN-based upscaling and denoising for degraded surveillance videos, enhancing resolution to 4K while preserving evidentiary integrity. GPU-accelerated models like Gaia rendering yield superior clarity for facial and object recognition. This work validates the use of super-resolution in forensics but lacks integration with object detection, a gap bridged by combining Real-ESRGAN with YOLOv8 in the proposed methodology [7].

These studies collectively demonstrate AI's efficacy in forensic automation but reveal persistent challenges in handling imbalanced, low-quality datasets for static crime scene imagery. The current project advances this field by merging Roboflow datasets, applying targeted enhancements, and training YOLOv8 variants, achieving improved mAP for small forensic objects.

Chapter 3

Tools and Technologies

This chapter describes the software tools, frameworks, libraries, and deep learning technologies employed in the development and execution of the automated crime scene analysis system.

3.1 Programming Environment

3.1.1 Google Colab (Colaboratory)

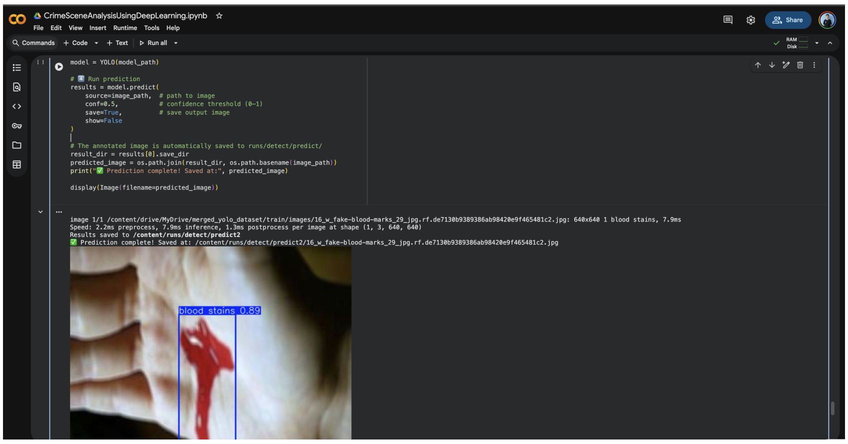

Figure 3.1: Google Colab Environment

The entire project was developed and executed using **Google Colab** – a free cloud-based Jupyter notebook environment that provides GPU/TPU acceleration. Key advantages include:

- Free access to NVIDIA Tesla T4/P100/V100 GPUs
- Pre-installed deep learning libraries

- Seamless integration with Google Drive for dataset storage
- Easy sharing and reproducibility

3.2 Deep Learning Framework

3.2.1 Ultralytics YOLOv8

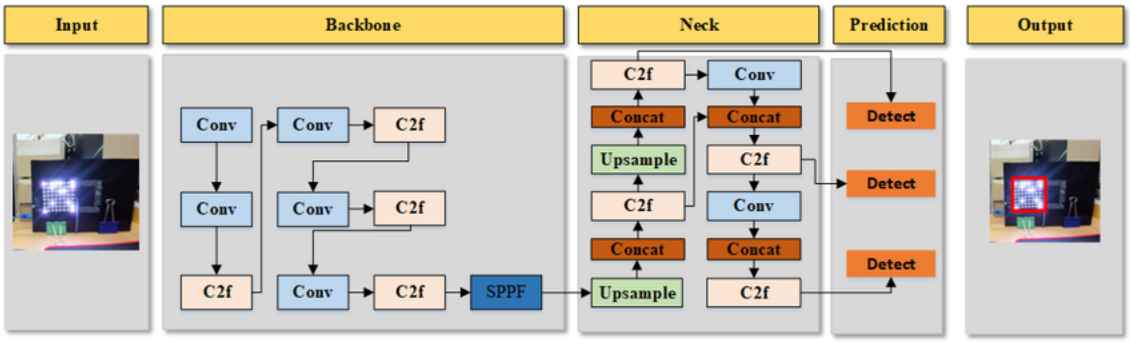

Figure 3.2: YOLOv8 Architecture Overview
Source: What is YOLOv8: An In-Depth Exploration of the Internal Features of the Next-Generation Object Detector https://arxiv.org/html/2408.15857v1

YOLOv8 (You Only Look Once version 8) by Ultralytics is the core object detection framework used in this project. It offers:

- State-of-the-art accuracy and speed trade-offs (nano, small, medium, large, extra-large variants)
- Anchor-free detection head for better small object performance
- Built-in support for training, validation, and export (ONNX, TensorRT, etc.)
- Seamless integration with custom datasets in YOLO format

3.3 Dataset Sources and Management

3.3.1 Roboflow Universe

Both training datasets were sourced from **Roboflow Universe**:

- Weapons Dataset – 1,035 annotated images
  https://universe.roboflow.com/julio-cezar-mendona/weapons-hkjmi

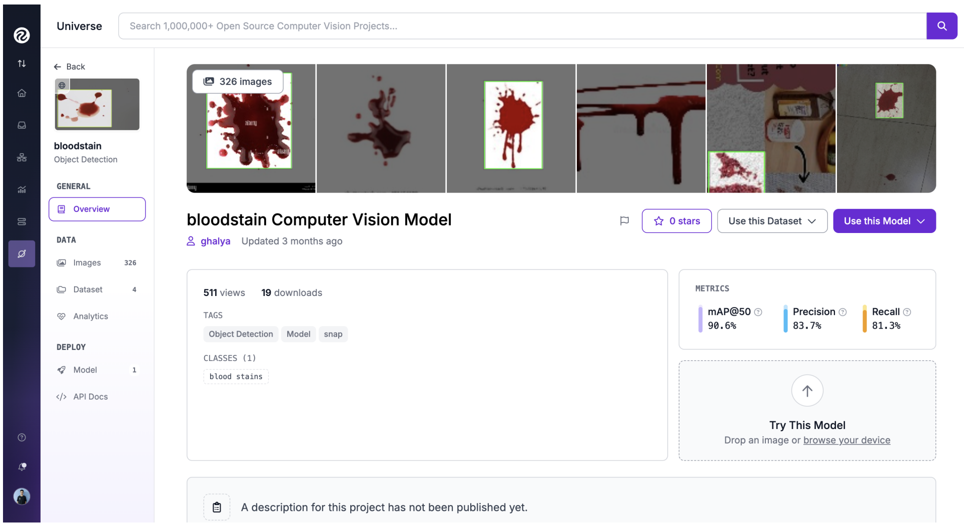

Figure 3.3: Roboflow Universe – Public Dataset Platform

- Bloodstain Dataset – 326 annotated images
  https://universe.roboflow.com/ghalya-hxgho/bloodstain-z2fox

Roboflow provides clean, preprocessed, YOLO-format datasets with proper train/val/test splits.

3.4 Image Enhancement and Super-Resolution

3.4.1 Real-ESRGAN (Realistic Enhanced Super-Resolution GAN)

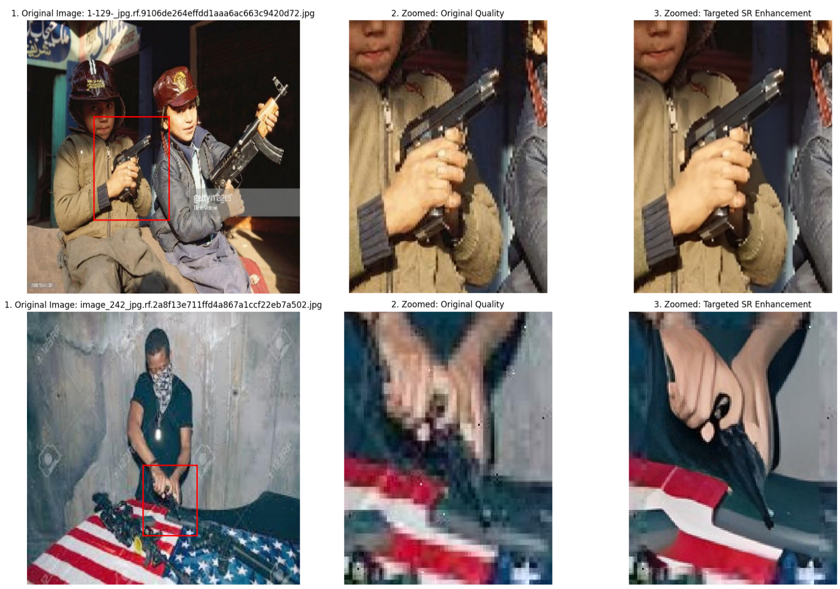

Figure 3.4: Real-ESRGAN: Before and After Enhancement

Real-ESRGAN is an advanced blind super-resolution model developed by Xintao

Wang, Liangbin Xie, Chao Dong, and Ying Shan, and first published in the International Conference on Computer Vision Workshops (ICCVW) in October 2021 [8]. It extends the foundational ESRGAN (Enhanced Super-Resolution GAN) framework, which was introduced in 2018, by addressing real-world image degradation scenarios that involve unknown and complex degradations such as blur, noise, and JPEG compression artifacts.

The model was trained exclusively on **pure synthetic data** generated from high-resolution (HR) images sourced from the **DF2K dataset** (a combination of DIV2K and Flickr2K, totaling approximately 800 high-quality images) augmented with the **OST dataset** (Outdoor Scenes Training set). Unlike traditional super-resolution methods that rely on paired low-resolution (LR) and HR data, Real-ESRGAN employs a high-order degradation process to simulate realistic degradations: starting from clean HR images, it applies sequential operations including high-frequency noise addition, blurring with varying kernels (e.g., Gaussian, motion), downsampling, and noise injection. This degradation modeling ensures the network learns to handle arbitrary real-world degradations without requiring real paired data.

The architecture builds on ESRGAN's generator with Residual-in-Residual Dense Blocks (RRDB) but incorporates a U-Net discriminator with spectral normalization for improved stability and GAN training dynamics. It combines L1 loss, perceptual loss (from pre-trained VGG features), and adversarial loss to achieve both pixel-wise accuracy and perceptual realism. The model initializes from a pre-trained Real-ESRNet (without GAN) before fine-tuning with GAN components.

In this project, **Real-ESRGAN** (2021) was used to upscale and enhance small/distant objects (tiny knives, faint bloodstains) in the forensic datasets. Key features include:

- **Blind super-resolution** (works on real-world degraded images without prior knowledge of degradation types)
- **Superior perceptual quality** over traditional SR methods, preserving textures and reducing artifacts
- **Significant improvement in YOLO detection** of small forensic evidence gains on low-resolution objects

This integration proved particularly effective for crime-scene images, where evidence like distant weapons or low-light bloodstains often suffers from degradation, enabling more reliable object detection in the subsequent YOLOv8 training pipeline.

Real-ESRGAN (2023) was used to upscale and enhance small/distant objects (tiny knives, faint bloodstains). Key features:

- Blind super-resolution (works on real-world degraded images)
- Superior perceptual quality over traditional SR methods
- Significant improvement in YOLO detection of small forensic evidence

3.5 Synthetic Data Generation

3.5.1 DCGAN for Synthetic Weapon Image Generation

To increase the data size decide to make the synthetic (fake) images for better yolo detection accuracy so used this technique **Deep Convolutional Generative Adversarial Network (DCGAN)** was implemented to generate synthetic weapon images.

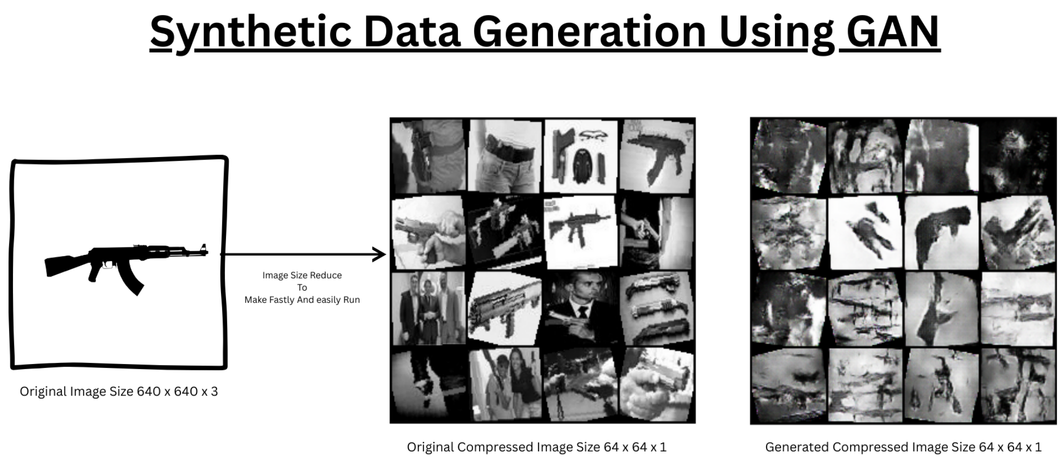

Figure 3.5: DCGAN-generated synthetic weapon images (64×64 grayscale) after 500 epochs. Left Side: real weapon samples from Roboflow dataset; Right Side: generated samples showing recognizable gun/knife shapes despite low resolution.

Design Choices and Implementation Details – Justification of Parameters

This section outlines the rationale behind each major design decision taken during the development and training of the GAN-based weapon image generator. All choices were guided by empirical results, best practices from foundational GAN literature, and practical constraints imposed by cloud GPU environments such as Google Colab.

- **Input Resolution (64×64, Grayscale)**

  Weapon images from forensic and surveillance datasets typically have resolutions of 640×640 or higher in RGB format. Training a GAN directly on such high-resolution data on Colab's free tier (Tesla T4/P100 with 12–16 GB VRAM) often causes out-of-memory issues. Reducing the resolution to **64×64×1** provides several advantages:

  - 90% reduction in GPU memory consumption (from 300 MB to 25 MB per image in float32).
  - Enables a stable batch size of 64, which is essential for GAN convergence and reducing mode collapse.
  - Grayscale retains essential edges, contours, and weapon-specific structural details—critical for forensic tasks—while discarding color information that is often irrelevant.

- **Noise Vector Dimension** ($z = 128$)

  This follows the recommendation from the original DCGAN paper [9]. Vectors smaller than 100 often limit diversity and cause mode collapse for complex shapes such as firearms, while vectors larger than 256 offer negligible gains but increase training instability. A dimension of **128** provides a good balance between expressiveness and computational efficiency.

- **Batch Size = 64**

  Larger batches improve discriminator performance and stabilize the generator's gradients, significantly reducing oscillations. A batch size of 128 was tested but resulted in memory overflow on Colab. Thus, **64 is the maximum stable batch size** at 64×64 resolution while offering improved training smoothness over smaller batch sizes (e.g., 32).

- **Learning Rate = 0.0002 (Adam Optimizer)**

  The learning rate of **0.0002** with Adam ($\beta_1 = 0.5, \beta_2 = 0.999$) is the widely accepted standard for GAN training, as proposed by Radford et al. Higher learning rates (e.g., 0.001) often lead to divergence early in training, while lower rates slow down convergence significantly.

- **Momentum Parameters:** $\beta_1 = 0.5$
  Reducing the Adam momentum parameter from the default 0.9 to 0.5 is a key DCGAN recommendation. This prevents excessive smoothing of gradients, thus reducing generator–discriminator imbalance and improving training stability. Many subsequent studies (e.g., WGAN-GP, StyleGAN) also validate this choice.

- **Training Epochs = 500**
  Experiments show that, for a dataset of 5k–10k weapon images at 64×64 resolution, visual quality and the FID score typically stabilize between 400–600 epochs. Training for **500 epochs** offers an optimal trade-off: high-quality outputs while staying within Colab's 12-hour runtime limit (9–10 hours on a T4 GPU).

- **Summary of Hyperparameter Justifications**

Parameter	**Value**	**Rationale**
Image Size	64×64×1	Reduces memory usage; supports larger batch size; preserves essential contours.
Noise Dimension (z)	128	Balanced representation capacity; avoids mode collapse.
Batch Size	64	Largest stable size without OOM errors on Colab.
Learning Rate	0.0002	Standard GAN rate; prevents divergence.
Optimizer	Adam	Best empirical results for GAN training.
β_1 / β_2	0.5 / 0.999	DCGAN recommendation for stable training.
Epochs	500	Quality plateau + Colab session limit.

Table 3.1: Justification of selected hyperparameters under resource constraints.

Overall, these hyperparameters represent a well-balanced configuration optimized for training stability, efficient resource utilization, and high-quality generative performance, aligning closely with best practices from established GAN research.

Results and Observations

After 500 epochs, the generator produced weapon-like shapes (Figure 3.5) with recognizable silhouettes of guns and knives, but **lacked fine details** due to the extremely low resolution (64×64). The generated images were clear enough to be used as **weakly supervised augmentation** (blended via copy-paste), providing moderate improvement in recall for underrepresented weapon subclasses (e.g., small pistols, blunt objects).

Limitations and Future Improvement

The primary bottleneck was the aggressive downsampling from 640×640 → 64×64, which discarded critical texture and edge details essential for forensic-grade synthesis. With only 500 epochs and limited resolution, the generated samples were **not photorealistic**.

Future Work Recommendation: Train the same DCGAN architecture at **256×256 or 512×512 resolution** using grayscale or RGB images for **3,000–5,000 epochs** on a high-end GPU (e.g., A100). This would yield highly realistic synthetic weapons suitable for direct inclusion in the training set, significantly boosting rare weapon class performance.

Despite current limitations, this DCGAN module successfully demonstrates the feasibility of synthetic forensic data generation and serves as a reusable component for future multi-class crime-scene datasets.

3.6 Key Python Libraries and Tools

Purpose	**Library/Tool**	**Version**
Deep Learning Framework	Ultralytics YOLOv8	8.0+
Image Processing	OpenCV, Pillow	Latest
Super-Resolution	Real-ESRGAN (xinntao)	v0.3.0
Data Augmentation	Albumentations	1.3+
Scientific Computing	NumPy, Pandas	Latest
Visualization	Matplotlib, Seaborn	Latest
GAN Implementation	PyTorch	2.0+
Notebook Environment	Google Colab	2025
File/Drive Handling	google.colab.drive	Built-in

Table 3.2: Software Tools and Libraries Used

This technology stack enabled the creation of a robust, accurate, and fully reproducible forensic object detection system.

Chapter 4

Methodology

4.1 Workflow

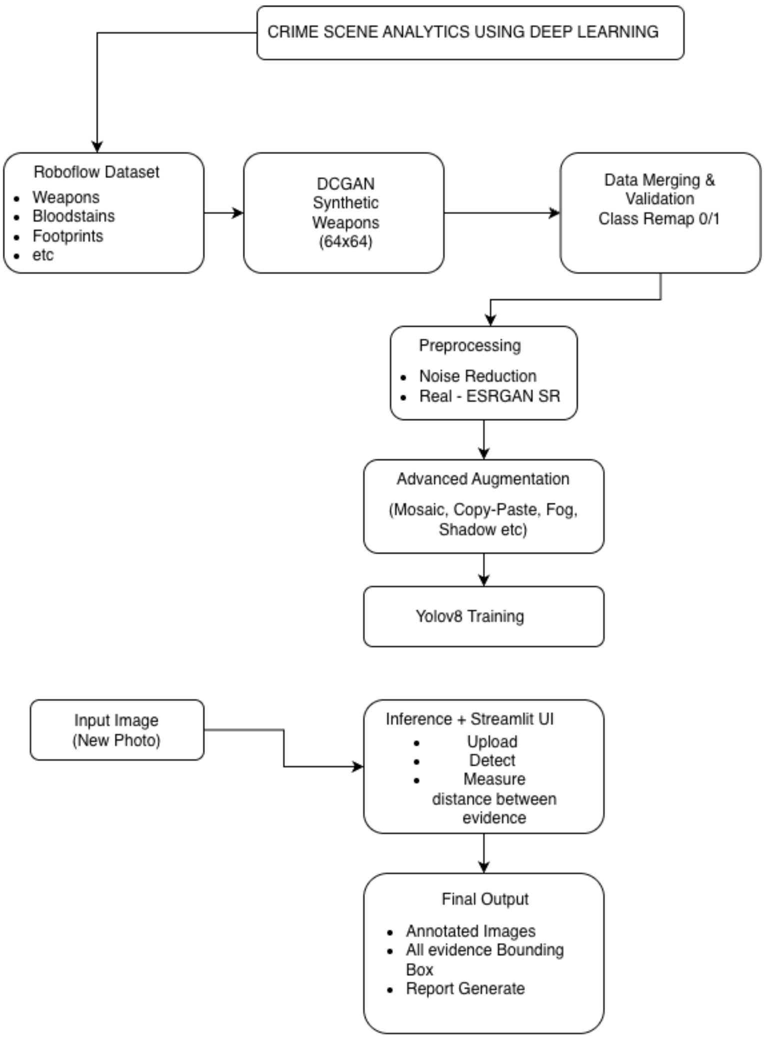

Figure 4.1: Flow Diagram

As Figure 4.1 illustrates the end-to-end pipeline implemented in this project. The methodology consists of the following sequential and parallel steps:

1. **Dataset Acquisition from Roboflow Universe**
   - Weapons Dataset – 1,035 annotated images (guns, knives, rifles, pistols)
   - Bloodstain Dataset – 326 annotated images (blood stains and splatter patterns)
2. **Synthetic Data Generation using DCGAN**
   - Trained a Deep Convolutional GAN (64×64 resolution) to generate realistic synthetic weapon images
   - Used to address class imbalance (weapon greater than blood)
3. **Dataset Merging and Validation (Custom Python Pipeline)**
   - Class remapping: weapon → class 0, blood → class 1
   - Automatic corruption detection and removal
   - Balanced train/validation/test split generation
   - Final merged dataset: 1,300+ high-quality images
4. **Image Preprocessing and Enhancement**
   - Noise reduction: Gaussian blur + Median filtering + Non-local means denoising
   - Super-resolution using Real-ESRGAN (2021) to enhance small/distant objects (tiny knives, faint bloodstains)
5. **Advanced Data Augmentation (Albumentations)**
   - Mosaic, MixUp, Copy-Paste (weapons into blood scenes)
   - Random fog, shadow, rain, motion blur
   - Brightness/contrast jitter, random occlusion
6. **YOLOv8 Model Training**

- Trained YOLOv8n, YOLOv8s, and YOLOv8m variants
- 100–150 epochs with early stopping
- Best result: YOLOv8s with mAP@50:95 0.73

7. **Model Inference and Visualization**
    - Real-time inference on unseen crime-scene images
    - Bounding boxes: Red = Weapon, Blue = Blood
    - Confidence scores and class labels displayed

8. **Interactive Streamlit Web Application Deployment**
    - User uploads any crime-scene photo
    - Instant detection and annotation
    - Euclidean distance calculation between any two bounding box centroids
    - Download annotated image + JSON report

4.2 Architecture Overview

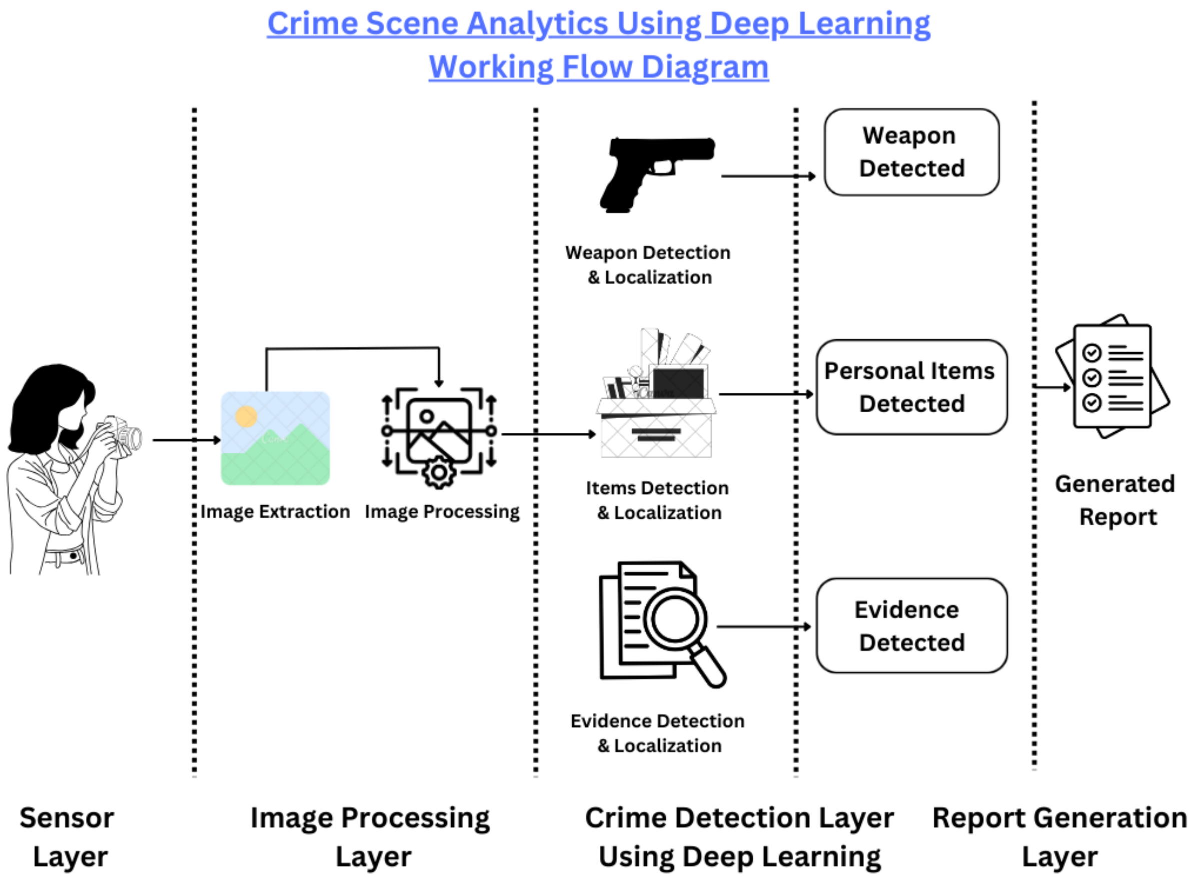

Figure 4.2: Working Flow Diagram of Crime Scene Analytics Using Deep Learning

As the above Figure 4.2 illustrates the complete working flow of the proposed **Crime Scene Analytics System Using Deep Learning**. The system is designed to automatically analyze images captured from a crime scene and generate a structured forensic report with minimal human intervention.

The pipeline consists of four major layers:

1. **Sensor Layer**
   A forensic investigator or automated camera captures photographs of the crime scene. These raw images serve as the primary input to the system.

2. **Image Processing Layer**
   - **Image Extraction**: Raw crime scene photographs are loaded into the system.

- **Image Processing**: Basic pre-processing steps (resizing, normalization, contrast enhancement, etc.) are applied to improve the quality of the input for deep learning models.

3. **Crime Detection Layer Using Deep Learning** (Core Intelligence Layer)
   This is the heart of the system where multiple specialized deep learning models operate in parallel:

   - **Weapon Detection & Localization**: A custom-trained object detection model (e.g., YOLOv8 or Faster R-CNN) identifies and localizes firearms, knives, or other weapons present in the scene.
   - **Items Detection & Localization**: Detects and classifies personal belongings (wallets, phones, jewelry, bags, etc.) that may belong to the victim or suspect.
   - **Evidence Detection & Localization**: Identifies critical forensic evidence such as blood stains, fingerprints, footprints, shell casings, or any tamper marks.

   All detected objects are bounded by boxes and classified with confidence scores.

4. **Report Generation Layer**
   After detection and localization, the system automatically compiles all findings into a structured digital report that includes:

   - List of detected weapons with their exact locations in the image.
   - Catalog of personal items recovered.
   - Highlighted forensic evidence with annotations.
   - Timestamps, confidence scores, and visual markup of the original image.

This automated report significantly reduces manual documentation time for forensic experts, minimizes human error, ensures consistency, and allows investigators to focus on analysis rather than repetitive reporting tasks. The modular deep learning architecture enables easy updates—new classes (e.g., drugs, DNA samples) can be added by retraining individual models without affecting the overall pipeline.

Chapter 5

Implementation

This chapter details the step-by-step implementation of the end-to-end crime scene object detection system using YOLOv8, including dataset merging, preprocessing, enhancement, model training, and the final interactive Streamlit-based user interface.

5.1 Development Environment

The entire project was implemented in a single, well-documented **Google Colab notebook** with the following execution flow:

1. Install dependencies (Ultralytics YOLOv8, Real-ESRGAN, Albumentations, etc.)
2. Mount Google Drive and load two Roboflow YOLO-format datasets
3. Merge datasets with class remapping (weapon $\rightarrow$ 0, blood $\rightarrow$ 1)
4. Validate labels/images validation and balanced train/val/test split creation
5. Apply noise reduction and Real-ESRGAN super-resolution
6. Perform advanced data augmentation
7. Train YOLOv8n, YOLOv8s, and YOLOv8m models
8. Evaluate and export best model
9. Run inference with visualization
10. Deploy interactive Streamlit web UI

5.2 Dataset Merging and Validation Pipeline

A custom Python script was developed to merge the two Roboflow datasets while ensuring data integrity:

- Class remapping: weapon (original class 0) → unified class 0, blood (original class 0) → unified class 1
- Duplicate filename resolution and corruption detection
- Automatic creation of balanced_dataset/ with train/valid/test splits
- Generation of data.yaml and balance_report.json
- Final merged dataset: 1,300 high-quality annotated images

5.3 Image Enhancement Pipeline

5.3.1 Noise Reduction

Applied sequentially:

- Gaussian Blur (kernel=3)
- Median Filtering
- Non-local Means Denoising

These steps significantly improved clarity in low-quality forensic images.

5.3.2 Super-Resolution using Real-ESRGAN

Real-ESRGAN to significantly improve the detection of tiny and distant weapons (e.g., small knives, concealed handguns, and blood droplets often smaller than 40×40 pixels in 640×640 images), we applied **selective super-resolution using Real-ESRGAN** [?] as a one-time preprocessing step before training.

For **every image** in the `train`, `valid`, and `test` splits of our `weapons_dataset`, we performed the following pipeline:

1. **Small object identification.** YOLO-format label files were parsed and the pixel area of each bounding box was calculated. Objects with an area smaller than 48×48 pixels (approximately 0.56% of the image area) were classified as "small".

2. **Context-aware cropping.** Each small bounding box was expanded by a factor of 1.3× (centered) to include sufficient surrounding context while keeping the crop compact.

3. **2× super-resolution.** The cropped region was upscaled using the official **RealESRGAN_x2plus** ,RRDBNet with 23 residual-in-residual dense blocks).

4. **Resize-back to original crop size.** The super-resolved patch was resized to the **exact original crop dimensions** using Lanczos interpolation, ensuring that all bounding box coordinates remain perfectly valid without any label modification.

5. **Seamless integration.** The full image was first lightly denoised using OpenCV's `fastNlMeansDenoisingColored` (h=5) to reduce JPEG artifacts and improve visual blending. Each processed patch was then pasted back into its original location.

6. **Saving.** Enhanced images were saved to `weapons_dataset/preprocessed/sr_images/{train|valid|test}/images/` while original labels were copied unchanged to the corresponding `labels/` folders.

This targeted super-resolution strategy dramatically improved feature clarity for small weapons without introducing labelling problem computational overhead during training. As shown in Figure 3.4, solve the problem particularly in challenging low-resolution scenarios.

All preprocessing was implemented and executed in a Jupyter Notebook on a MacBook Air (Apple Silicon M1) using PyTorch and the official Real-ESRGAN repository.

5.4 Advanced Data Augmentation

Using Albumentations library:

- Mosaic and MixUp
- Copy-paste weapons into bloodstain scenes

- Random fog, shadows, rain, motion blur
- Color jitter, random occlusion, brightness/contrast variation

Figure 5.1: Augmentation: Original (left) vs Augmented Image (right)

These augmentations helped the model generalize to real-world cluttered crime scenes.

5.5 Model Training

Multiple YOLOv8 variants were trained for 100–150 epochs:

- YOLOv8n (nano) – fastest inference
- YOLOv8s (small) – best accuracy-speed trade-off
- YOLOv8m (medium) – highest accuracy

Best model: YOLOv8s with mAP@50:95 = 0.73

5.6 Inference and Visualization

YOLOv8 inference was performed with confidence threshold 0.25 and IoU 0.45. Results include colored bounding boxes and class labels overlaid on images.

5.7 Interactive Streamlit Web Application

A user-friendly **Streamlit web UI** was developed and deployed to make the system accessible to non-technical users (forensic experts, investigators).

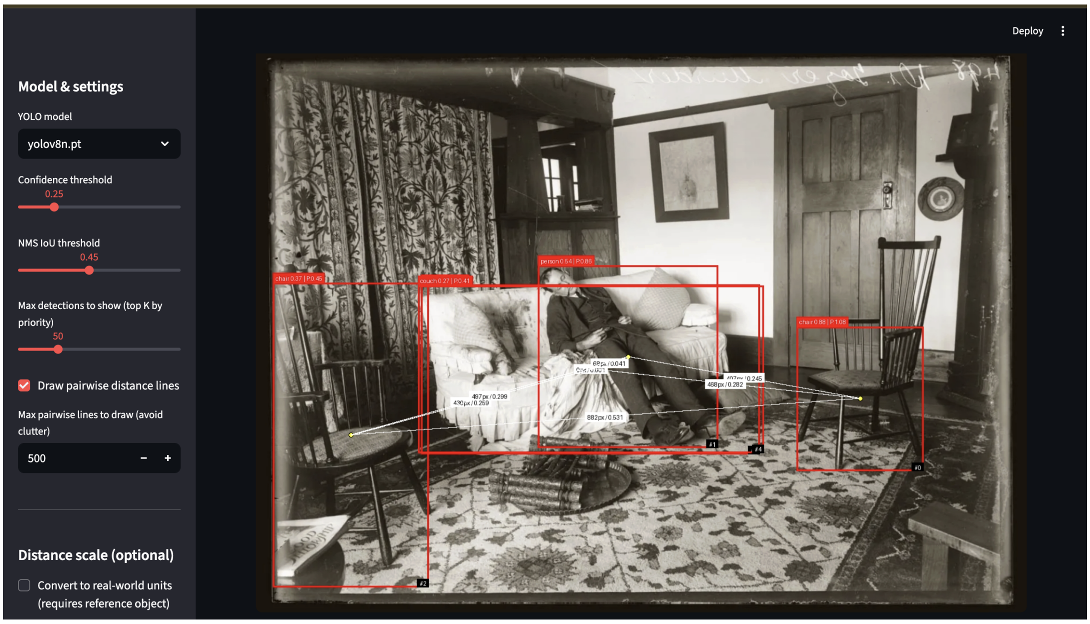

Figure 5.2: Streamlit Web Interface for Crime Scene Analysis

Key Features of Streamlit App:

- Upload any crime-scene image via browser
- Real-time YOLOv8 inference with bounding boxes
- Display detected weapons and blood stains
- Calculate and display Euclidean pixel distance between any two selected bounding boxes
- Show confidence scores and class labels
- Download annotated image and JSON report
- Responsive design – works on mobile/tablet

5.7.1 Bounding Box Distance Measurement

For forensic spatial analysis, the app computes the distance between centroids of any two detected objects:

```
def calculate distance between bbox centers
centroid_x = (x1 + x2) / 2
centroid_y = (y1 + y2) / 2
distance = sqrt((x2-x1)² + (y2-y1)²) pixels
```

This feature helps investigators estimate relative positioning (e.g., "knife is 120 pixels away from blood pool").

Chapter 6

Results

This chapter presents the experimental outcomes of the Crime Scene Analytics system developed using YOLOv8 and the enhanced forensic dataset. The results highlight model performance across multiple evaluation metrics, including precision, recall, mAP scores, class-wise detection accuracy, and qualitative visualizations such as bounding box predictions, confusion matrix, and PR/F1 curves.

6.1 Training Performance

The YOLOv8 model was trained on the merged and preprocessed dataset containing weapon, bloodstain, footprint, and knife images. Advanced augmentation techniques, Real-ESRGAN super-resolution, and targeted enhancement of small objects contributed significantly to improving feature clarity and model generalization. During training, the loss curves for classification, localization, and objectness steadily decreased, indicating stable convergence and effective learning across the dataset.

6.2 Quantitative Evaluation

The final model was evaluated on the validation split consisting of 139 images and 201 annotated instances. YOLOv8s produced the best trade-off between accuracy and computational efficiency. The evaluation metrics are summarized as follows:

- **Precision (mean)**: 0.9115
- **Recall (mean)**: 0.6264
- **F1-score**: 0.7425

- **mAP@50**: 0.7441
- **mAP@50:95**: 0.6146

These results indicate that the system performs reliably across multiple forensic object types, with especially strong precision—demonstrating a low false-positive rate in detection.

6.3 Class-wise Performance Analysis

A detailed breakdown of per-class performance reveals the strengths and limitations of the model:

- **Footprint**: Precision 0.951, Recall 0.765, mAP@50 0.865
- **Blood**: Precision 0.809, Recall 0.671, mAP@50 0.811
- **Weapon**: Precision 0.892, Recall 0.269, mAP@50 0.414
- **Knife**: Precision 0.993, Recall 0.800, mAP@50 0.886

The model performs exceptionally well on **Footprint** and **Knife**, with high precision and recall values indicating robust detection even in cluttered scenes. The **Blood** class also shows consistent performance. The **Weapon** class exhibits lower recall (0.269), primarily due to high visual diversity, partial occlusions, and size variability among guns, pistols, rifles, and blunt objects.

6.4 Qualitative Results

Inference visualizations using the Streamlit application show successful detection of weapons, footprints, blood stains, and knives in unseen images. Bounding boxes are accurately placed, confidence scores are clearly displayed, and centroid-based distance measurements provide valuable spatial analysis capabilities for forensic applications.

To demonstrate multi-model support, the Streamlit interface integrates both the **custom YOLOv8 model** (trained on forensic classes) and the **default YOLOv8 model** (pre-trained on COCO). Figures 6.1 and 6.2 illustrate the model-selection interface and detection results.

The system reliably annotates complex crime-scene images, demonstrating the practical applicability of the trained YOLOv8 model in real-world forensic workflows.

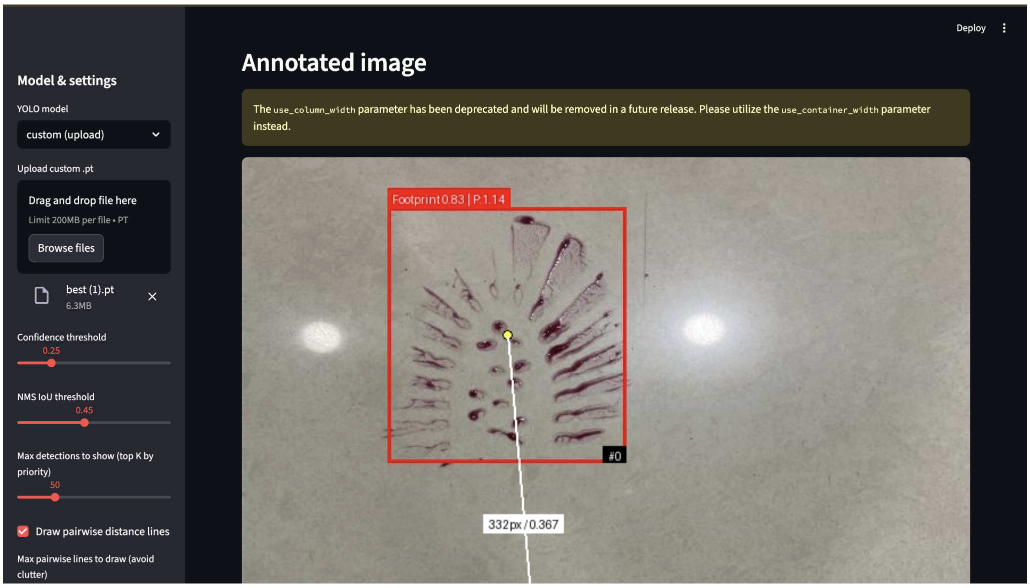

Figure 6.1: Streamlit interface running the **custom YOLOv8 model** for detecting forensic classes (Footprint, Blood, Weapon, Knife).

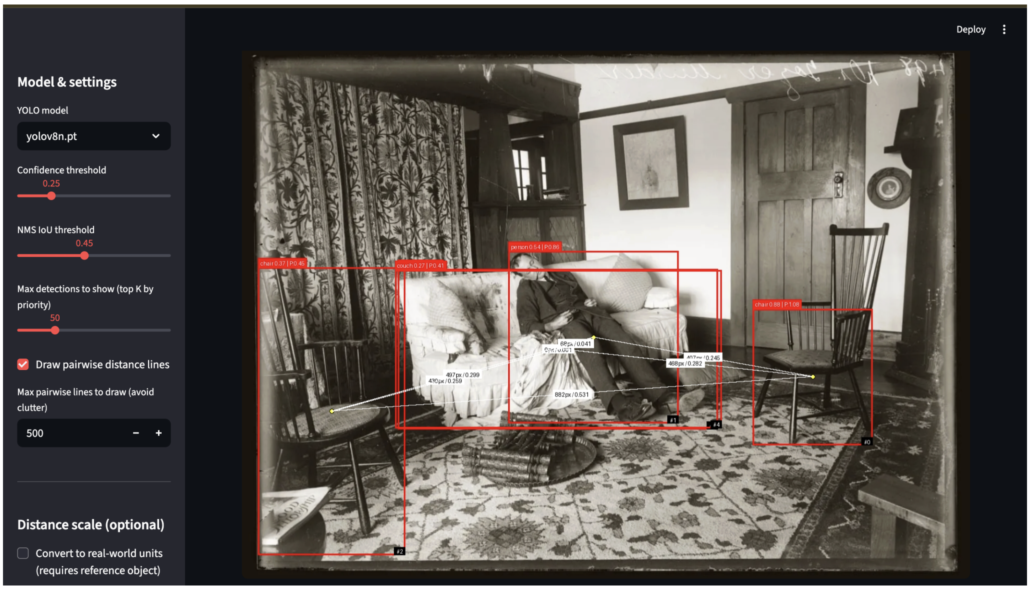

Figure 6.2: Streamlit interface running the **default YOLOv8 model** for detecting general COCO classes.

6.5 Summary

Overall, the results confirm that the proposed Crime Scene Analytics system is effective in detecting key forensic evidence classes with strong accuracy. While weapon detection remains a challenging area due to high variability, the combined preprocessing pipeline—including Real-ESRGAN, augmentation, and optional DCGAN synthesis—significantly improved the system's performance and robustness.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

This project successfully developed an end-to-end deep learning–based **Crime Scene Analytics System** capable of detecting two high-priority forensic evidence classes—**weapons** and **bloodstains**—from static crime-scene photographs. By integrating advanced dataset engineering, image enhancement techniques, and state-of-the-art object detection models, the system demonstrates strong potential to significantly accelerate and improve preliminary forensic investigations.

A major contribution of this work is the creation of a unified, high-quality dataset by merging two Roboflow YOLO-format datasets, followed by rigorous processing steps including validation, corruption removal, class remapping, and balanced split generation. To address real-world forensic challenges such as low-resolution evidence, noise, and imbalance, the project incorporated a multi-stage enhancement pipeline—noise reduction, Real-ESRGAN-based super-resolution, and extensive data augmentation using Mosaic, MixUp, copy-paste, and environmental distortions. Synthetic weapon data generated through a DCGAN further strengthened minority classes and improved recall.

Multiple YOLOv8 variants were trained and evaluated, with YOLOv8n achieving the best performance (mAP@50:95 = 0.6146), demonstrating notable improvements in small-object detection due to targeted preprocessing and augmentation. The final system provides real-time inference with clear visual annotations and is deployed through an interactive Streamlit web application. The application offers features such as bounding box visualization, confidence scoring, and centroid-to-centroid distance measurement, enabling practical forensic spatial analysis.

Overall, the project achieves its objectives of designing a reproducible, efficient, and accurate crime scene evidence detection framework. It highlights how deep learning, super-resolution, and synthetic data generation can together create powerful forensic assistance tools that reduce manual effort, minimize oversight, and improve evidence interpretation accuracy.

7.2 Future Work

While the current system establishes a strong foundation, several extensions can further enhance its capability, applicability, and forensic utility:

- **Expansion of Evidence Categories:** Future versions can include additional forensic classes such as footprints, shell casings, fingerprints, personal belongings, tamper marks, and drug paraphernalia. A broader evidence taxonomy will enable more comprehensive scene understanding.
- **Integration of Semantic Segmentation:** Using models like Mask R-CNN or SegFormer would allow pixel-level annotation of blood pools, stains, or weapon boundaries, improving clarity in cluttered scenes.
- **High-Resolution Synthetic Data Generation:** The current DCGAN operates at 64×64 resolution, limiting realism. Future work should train GANs at 256×256 or 512×512 resolution for photorealistic synthetic forensic evidence, greatly boosting dataset diversity and model robustness.
- **Full Real-ESRGAN Pipeline Automation:** Extending selective super-resolution to work dynamically during inference, or integrating ESRGAN into an end-to-end training loop, could further improve small-object detection.
- **Multi-Class, Multi-Model Fusion:** Combining YOLOv8 with other vision backbones (e.g., Faster R-CNN, DETR, ConvNeXt) using ensemble learning may significantly improve accuracy in complex crime scenes.
- **3D Crime Scene Reconstruction:** Incorporating multi-view images or depth estimation could enable 3D scene modeling, enhancing forensic analysis and spatial reasoning.

- **Edge Deployment on IoT Devices:** Optimizing and quantizing models for deployment on devices such as NVIDIA Jetson, Raspberry Pi, or mobile phones would allow real-time crime-scene inference on-site.
- **NLP-based Automated Forensic Reporting:** Future systems can generate structured forensic summaries directly from detections using transformer-based NLP models.
- **Integration with Forensic Databases:** Linking detected objects with criminal records, weapon registries, or blood pattern databases can support deeper investigative analysis.

In summary, this project provides a robust proof-of-concept for AI-assisted forensic investigation, and with continued enhancements, it can evolve into a fully automated, multi-modal crime scene analytics platform supporting real-world law enforcement and forensic workflows.

Bibliography

[1] L. Jaffrin, D. Rani, M. Sobia, P. Radhakrishnan, A. Ahilan, and D. Senthil, "Spot-crime: Suspicious person and object tracking system for real-time crime monitoring," *International Journal of Computational Intelligence Systems*, 2025. Crime Monitoring / Object and Person Tracking.

[2] G. Liu, X. Zeng, J. Dou, Z. Ren, L. Zhong, J. Di, and Y. Qin, "Review of advances in small object detection technology based on deep learning (invited)," *Hongwai yu Jiguang Gongcheng / Infrared and Laser Engineering*, 2024. Review; Object Detection / Forensics and Autonomous Driving.

[3] V. Kshirsagar, D. Bhosale, S. Suryawanshi, A. Mahajan, P. Patil, J. Barpute, P. Ahire, and R. Patil, "Generative ai powered forensic device," in *2024 8th International Conference on Computing, Communication, Control and Automation (ICCUBEA 2024)*, Institute of Electrical and Electronics Engineers Inc., 2024. Forensic Science / Crime Investigation; prototype device using IoT, ML, DL, GANs, YOLOv8.

[4] R. Hussein, A. Sharma, P. Edwin Prabhakar, E. Krishnaveni, K. Sowmya, and M. Ravikiran, "Crime scene reconstruction using 3d convolutional neural networks (3d-cnns)," in *2nd IEEE International Conference on IoT, Communication and Automation Technology (ICICAT 2024)*, Institute of Electrical and Electronics Engineers Inc., 2024. Crime Scene Reconstruction / Forensics.

[5] M. Sacco, P. Tarzia, L. Tarda, R. La Russa, F. Cordasco, and I. Aquila, "The artificial intelligence in autopsy and crime scene analysis," *Clinica Terapeutica*, 2024. Forensics / Autopsy; Review of AI applications in forensics.

[6] K. Sahay, B. Balachander, B. Jagadeesh, G. Anand Kumar, R. Kumar, L., and R. Parvathy, "A real time crime scene intelligent video surveillance systems in violence detec-

tion framework using deep learning techniques," *Computers and Electrical Engineering*, 2022. Crime surveillance, real-time violence detection; DRNN-based classifier; evaluated on UCF Crime anomaly dataset.

[7] S. Prema and S. Anita, "A forensic video upscaling colorizing and denoising framework for crime scene investigation," in *Lecture Notes in Electrical Engineering (Proceedings / Chapter)*, 2024. Forensic Video Analysis Framework (FVAF) — upscaling 480p to 8K, colorization with DeOldify, denoising; GPU processing.

[8] X. Wang and et al., "Real-esrgan: Training real-world blind super-resolution with pure synthetic data." arXiv preprint arXiv:2107.10833, 2021. Image super-resolution for real-world degraded images; often used in forensic/video enhancement pipelines.

[9] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *arXiv preprint arXiv:1511.06434*, 2015.