Smart Screening: Identifying Restricted Items in CLCXray Images with YOLO Models

Nagasarapu Sarayu Krishna
Dept. of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4cse22037@bl.students.amrita.edu

Trisha Vijayekkumaran
Dept. of Computer Science and Engineering
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
bl.en.u4cse22065@bl.students.amrita.edu

Dr. Rimjhim Singh
Dept. of Computer Science and Engineering.
Amrita School of Computing, Bengaluru
Amrita Vishwa Vidyapeetham, India
ps$_{r}imjhim@blr.amrita.edu$

***Abstract*—Object detection in X-ray images plays an important role in enhancing security systems in airports and other secure facilities. The Cutters and Liquid Containers X-ray Dataset (CLCXray) is a standard for detecting potential dangers concealed in check-in luggage. To offer high accuracy and real-time processing, all major YOLO (You Only Look Once) models, from YOLOv5 to YOLOv12, were tested and used systematically. Insights from these models guided the development of a better YOLOv9 model with new attention structures for better detection capabilities. The advancements enable the new model to detect better using higher precision and recall even from occluded or complicated X-ray images. This paper is a critical analysis of the performance of the current YOLO models and the comparison of the new model with the baseline models. .**

***Index Terms*—Object Detection, X-ray Security Screening, YOLO, Automated Threat Detection, Computer Vision**

I. Introduction

Object detection in X-ray images is a key application in many fields, such as security screening, medical diagnosis, and industrial inspection. Manual inspection is slow and subject to human error, making the use of automated methods inevitable. The latest advances in deep learning and computer vision have greatly improved the accuracy and efficiency of object detection in X-ray images. Convolutional Neural Networks (CNNs) and current top-of-the-line structures, including You Only Look Once (YOLO), have proven excellent at identifying and categorizing objects inside intricate X-ray scans. They can perform real-time analysis and are thus appropriate for highthroughput settings such as airport security checkpoints and cargo scanning systems. YOLO models have been continuously upgraded over time, with updates like YOLO v9, v10, and v11 releasing enhancements in terms of speed, accuracy, and computational efficacy. The models are especially efficient for object detection under difficult situations, including occlusion, overlap, and varying contrast that are characteristic of X-ray images. The use of deep learning-based object detection methods improves automation, minimizes reliance on human screening, and increases overall detection accuracy. This research examines the use of object detection using YOLO for X-ray images, investigating the possibility of applying it for real-time security checks and other inspections.

II. Literature Review

Object detection in X-ray security scans is of prime importance for the detection of forbidden objects, weapons, and threats in baggage inspection systems. Deep learning models, especially YOLO-based architectures, have progressed in a way that object detection has become highly accurate. The most recent versions, from YOLOv9 onwards, incorporate better spatial attention, transformer-based feature extraction, and fast inference mechanisms to counter the limitations of occlusion, cluttered background, and real-time processing. This paper offers a review of recent progress in object detection on X-ray images, with a focus on YOLOv9-based techniques and their implementations on the CLC X-ray dataset. Since its release, YOLO (You Only Look Once) has seen massive improvements in its performance on detection tasks. YOLOv7 and YOLOv8 added transformer layers for enhanced feature fusion, and YOLOv9 has attention-based refinement methods included in it, so it performs well in detecting highly occluded objects [1,2,3]. Recent studies have investigated YOLO model adjustments for use with X-ray images. PIXDet introduced a new wholeprocess feature fusion method to greatly improve detection in baggage cluttered environments [4]. Overlapping image disentanglement methods for precise detection of illegal items in X-ray images were also introduced by ForkNet [5,6]. The CLC X-ray dataset is commonly used for object detection performance assessment in security scenarios. It has very occluded and messy baggage images, which demand strong feature extraction techniques. The foremost problem in this dataset is overlapped objects, which decrease detection accuracy. Previous research indicated that label-aware mechanisms can effectively cope with occlusion [7,8]. In addition, GANbased data augmentation methods have been effective in improving training performance [9,10]. Some improvements in YOLOv9 and later versions have been made to overcome the shortcomings of earlier YOLO architectures. The addition of multi-scale feature attention transformers enhances the model's capacity to differentiate between overlapping objects [11]. A task-decoupling YOLO architecture has been introduced to decouple localization and classification tasks, minimizing false positives [1]. Experiments indicate that suspicious object detection algorithms can be enhanced

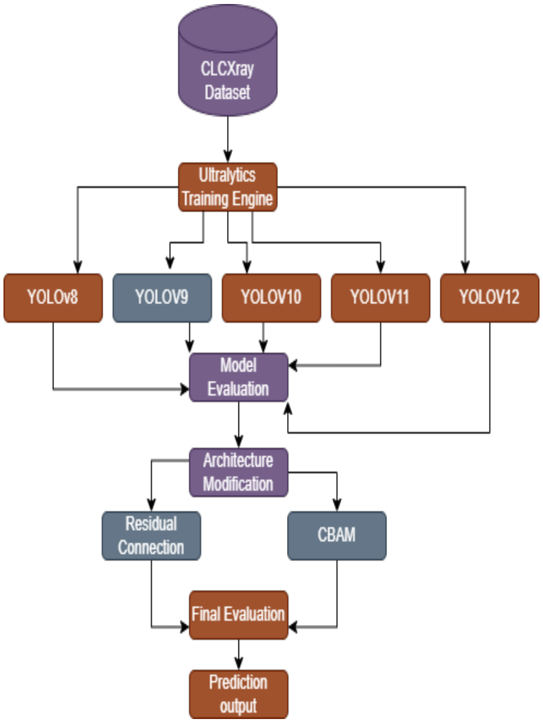

Fig. 1. System Architecture

using adaptive learning rates and hyperparameter optimization [12,13]. Bottom-up attention mechanisms and meta-learning have been introduced to enhance feature selection and classification in security imagery [14,15]. The efficiency of multi-scale hierarchical contour frameworks and multi-label classification mechanisms has been exhibited in enhancing object detection in security X-ray images [16,17]. Though YOLOv9 and the following architectures have demonstrated great improvements, problems like adversarial robustness and generalizability improvement across varied Xray datasets still exist. Combining self-supervised learning and meta-learning methods has been suggested as an upcoming improvement to enhance accuracy in real-world applications [18,19,20]. In summary, YOLOv9 object detection has shown promising performance on the CLCX-ray dataset through improvements in feature extraction, attention, and augmentation. Future work should aim to enhance interpretability, domain adaptation, and robustness to further improve performance in security-related tasks.

III. Methodology

This section presents the methodology, including the system architecture, the architecture of model variant with the attention mechanism, and detailed descriptions of the datasets used. The selection of model variants is driven by their ability to enhance feature extraction, improve multi-scale detection, and effectively capture details in X-ray imagery, which are crucial for identifying prohibited items with high accuracy. Figure. 1. illustrates the overall system architecture used in this study.

A. Dataset

The CLCXray dataset is a specialized X-ray image dataset designed for the detection of various objects commonly found in security scanning scenarios. It includes a diverse collection of X-ray images featuring different types of cutters, knives, liquid containers, and other related items. It corresponds to thirteen distinct object categories. These categories are:

- gun-knife-wrench-pilers-scissors (combined class)
- Cans
- CartonDrinks
- GlassBottle
- PlasticBottle
- SprayCans
- SwissArmyKnife
- Tin
- VacuumCup
- blade
- dagger
- knife
- scissors

The dataset is divided into three subsets: training, validation, and testing. The statistics of the dataset are summarized in Table I.

TABLE I
Dataset Statistics

Dataset	Total Files
Train Images	3187
Train Labels	3186
Test Images	957
Test Labels	956
Valid Images	957
Valid Labels	956

The distribution of object instances across classes in the entire dataset is shown in Table II. It can be observed that the most frequent class is *PlasticBottle* with 3196 instances, followed by *blade* (1597 instances), and *VacuumCup* (1213 instances). Some classes, such as *knife* and *GlassBottle*, have fewer instances, indicating potential class imbalance which should be considered during model training.

TABLE II
Label Distribution per Class

Class	Count
PlasticBottle	3196
blade	1597
VacuumCup	1213
scissors	1201
CartonDrinks	954
SprayCans	571
SwissArmyKnife	496
dagger	494
Tin	466
Cans	409
GlassBottle	351
knife	347

This dataset provides a challenging benchmark for object detection models due to the diversity of object types and the variability of their appearances in X-ray images.

B. Sample Images

Figure. 2. shows a few sample images from the CLCXray dataset, showcasing different object classes and their respective bounding boxes. These samples highlight the variability and complexity of objects present in X-ray scans.

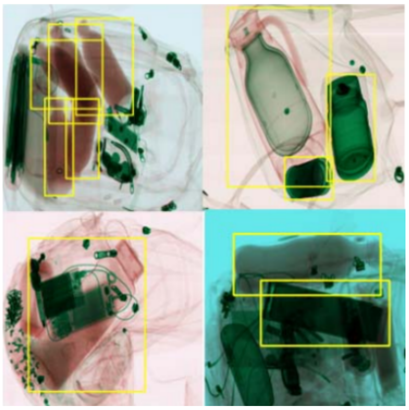

Fig. 2. Sample images from the CLCXray dataset with annotated objects

The CLCXray dataset [21], which includes diverse X-ray images of objects commonly found in security checks, is publicly available through the Roboflow Universe platform.

C. Data Preprocessing

The CLCXray dataset is well-structured and comes with clear annotations, requiring minimal preprocessing for object detection tasks. Each image is already labeled with bounding boxes and class identifiers in YOLO format, which is compatible with most modern object detection models. Since the dataset contains a substantial number of samples per class, additional techniques are not necessary. Furthermore, the images are of good quality, and the annotation consistency is well-maintained. Basic preprocessing steps include resizing the images to the required input dimensions of the model and normalizing pixel values.

D. Model Implementation

This section describes in length the different YOLO models used in this study, together with their setups, strategies, and particular modifications.

1) Baseline YOLO Models: To establish a strong baseline, five YOLO variants were implemented and evaluated: YOLOv8s, YOLOv9m, YOLOv10s, YOLOv11n, and YOLOv12s. The *s*, *m*, and *n* variants were selected and compared for their fast inference speeds and low resource consumption, making them ideal for practical deployment in constrained environments.

All models were trained on the CLCXray dataset, which consists of thirteen object categories. A uniform training and evaluation setup was maintained to ensure a fair comparison, utilizing the Ultralytics framework for streamlined implementation. Performance was measured using standard object detection metrics, including Precision, Recall, F1-score, mAP@50, and mAP@50–95.

Among all the variants, YOLOv9m consistently outperformed the others across most metrics, demonstrating its robustness and efficiency. Consequently, YOLOv9m was selected as the foundation for the architectural improvements of incorporating the attention mechanisms.

2) Training Configuration and Parameter Settings: The training process for the YOLOv9m-based model was configured with specific parameter settings to optimize detection performance on the CLCXray dataset. The training configuration was executed using the Ultralytics framework, and the primary parameters are outlined as follows:

- **data**:Dataset configuration file that defines the train, validation, and test splits, as well as the class labels.
- **epochs**: The model was trained for 100 epochs to ensure sufficient learning across the dataset.
- **batch**: A batch size of 16 was used for stable gradient updates without exhausting memory resources.
- **imgsz**: The input image size was set to 640 × 640 pixels, which is optimal for object detection tasks while balancing detail and processing speed.
- **save**: The `save=True` flag ensures that checkpoints and final model weights are saved during training.
- **cache**: Caching of datasets was enabled (`cache=True`) to accelerate data loading times by storing images and labels in memory.
- **device**: The training was executed on a GPU (`device=0`), enabling faster computations and better parallel processing.
- **optimizer**: AdamW was chosen as the optimization algorithm for its adaptive learning rate capabilities and weight decay regularization, leading to better convergence.

3) Hardware Requirements: To efficiently train the YOLOv9m-based model, a robust hardware configuration is recommended. The model was trained on an Kaggle's GPU (Tesla T4 or higher is recommended) with at least 16GB of VRAM to handle the 640 × 640 image resolutions. A system with a minimum of 32GB RAM is ideal to manage caching and batch processing effectively. Additionally, running the experiments on cloud platforms such as Kaggle or Google Colab Pro with GPU acceleration significantly reduces training time.

4) Architectural Modifications with CBAM Integration: To enhance feature extraction and attention-based learning, the YOLOv9m architecture was modified by integrating Convolutional Block Attention Modules (CBAM) at strategic points in the network.This is created by cloning the original YOLOv9m and performing the following modifications:

- **Backbone Modifications**: The backbone of the model was restructured to improve hierarchical feature learning. Specifically, CBAM blocks were inserted after key feature extraction layers:

  - A CBAM block was added after the third pooling layer (P3) to focus on low-level features that are crucial for edge detection and fine-grained object features.
  - Another CBAM block was introduced after the fifth pooling layer (P5) to capture high-level semantic information, enhancing the model's ability to recognize complex object structures.

  The integration of CBAM at these points enables the model to implement both spatial and channel-wise attention, allowing it to selectively amplify relevant features while suppressing noise, thereby improving object localization and classification accuracy.
- **Head Modifications**: The head architecture was redesigned to integrate CBAM before final feature aggregation at different scales:

  - A CBAM block was placed before the final P3 output, optimizing the attention on fine details for small objects like *blade*, *knife*, and *SwissArmyKnife*.
  - Another CBAM block was inserted before the P5 output, ensuring enhanced focus on larger objects such as *CartonDrinks* and *VacuumCup*.

  These strategic placements allow the model to refine the feature maps right before detection, boosting both precision and recall rates for objects of varying sizes.

The rationale behind positioning CBAM blocks specifically at P3 and P5 is grounded in multi-scale feature representation:

- **P3 (Low-Level Features)**: CBAM at this stage allows the network to enhance edge and texture details crucial for small objects that are typically challenging to detect.
- **P5 (High-Level Features)**: Adding CBAM here emphasizes the semantic understanding of object shapes and larger contextual information, which improves object distinction in cluttered scenes.

The modifications were seamlessly integrated into the model's configuration, ensuring backward compatibility and optimized detection performance. Figure. 3 . illustrates the architecture of the redesigned YOLOv9m model with CBAM enhancements.

5) YOLOv9m Architecture Overview: As shown in Figure 4, the YOLOv9m architecture is designed for efficient multi-scale object detection with outputs at three scales: P3 (small objects), P4 (medium objects), and P5 (large objects). It consists of three main components: the backbone, the head, and the detection layer.

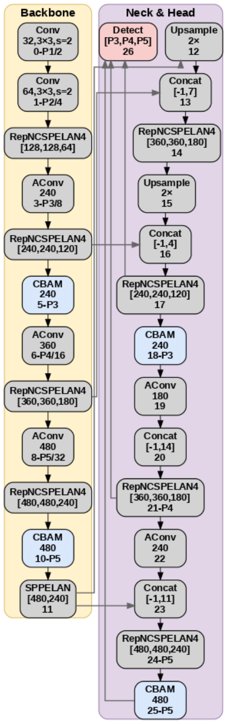

Fig. 3. YOLOV9m with CBAM module

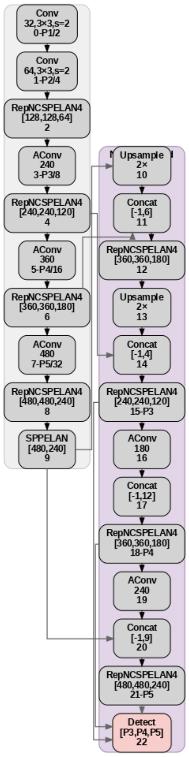

Fig. 4. Standard YOLOv9m Architecture

a) Backbone:: The backbone is responsible for feature extraction from the input image. It uses:

- **Conv layers** for basic spatial feature extraction.

- **RepNCSPELAN4** residual blocks, which apply repetitive convolutional operations to capture deeper semantic information.
- **AConv** (Attention Convolution) modules that enhance important feature regions.
- **SPPELAN** blocks to spatially pool and enrich the feature maps.

The backbone progressively reduces spatial resolution from P1/2 to P5/32, increasing feature depth and semantic richness.

b) Head:: The head performs upsampling and concatenation of features from deeper layers with shallower layers to preserve spatial details, which is crucial for detecting objects at different scales. Specifically:

- Features from P5 are upsampled and concatenated with P4 features.
- The combined features are further upsampled and concatenated with P3 features.
- Residual blocks (RepNCSPELAN4) are applied after concatenations to refine the multi-scale features.

c) Detection Layer:: The final detection layer aggregates the processed features from P3, P4, and P5 scales to output bounding boxes and class predictions. This multi-scale detection improves accuracy across object sizes.

d) Role of Residual Blocks (RepNCSPELAN4):: Residual blocks are placed at key stages in both the backbone and head for the following reasons:

- **Improved gradient flow:** They help avoid vanishing gradients by learning residual mappings, which allows training of deeper networks.
- **Multi-scale feature refinement:** Positioned at scale transition points, they enhance the representation of spatial and semantic features crucial for detecting objects of varying sizes.
- **Preservation of low-level features:** Skip connections inside residual blocks retain essential fine details, especially important for small object detection.

These properties make residual blocks critical to the YOLOv9m architecture's effectiveness in accurate and efficient object detection.

IV. Results

TABLE III
Comparison of YOLO Models

S.No	Model	Precision	Recall	mAP@50	mAP@50–95	F1-Score
1	YOLOv9	0.805	0.664	0.711	0.612	0.728
2	YOLOv11	0.777	0.627	0.665	0.555	0.694
3	YOLOv12	0.856	0.648	0.703	0.600	0.738
4	YOLOv5s	0.858	0.643	0.689	0.574	0.735
5	YOLOv8s	0.793	0.666	0.692	0.594	0.724
6	YOLOv10n	0.819	0.648	0.690	0.584	0.723
7	YOLOv9m (CBAM)	0.781	0.621	0.659	0.548	0.672

The loss plots for YOLOv5, as shown in (Fig. 5), demonstrate a consistent downward trend across all key loss components—box loss, objectness loss, and classification loss—for both training and validation sets. This indicates that the model is effectively learning spatial localization, object confidence, and class predictions over the course of 100 epochs. The training and validation curves closely align, suggesting minimal overfitting and good generalization to unseen data. Notably, the smooth convergence of `val/cls_loss` and `val/obj_loss` highlights the model's robust performance in maintaining classification and detection quality across validation samples .

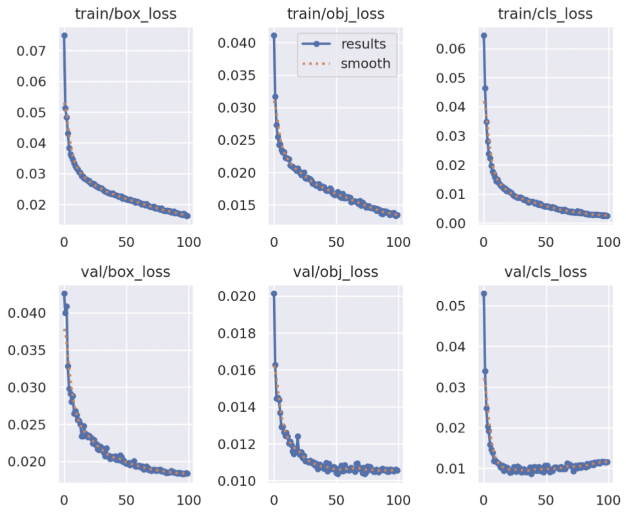

Fig. 5. Yolov5 Loss Plots

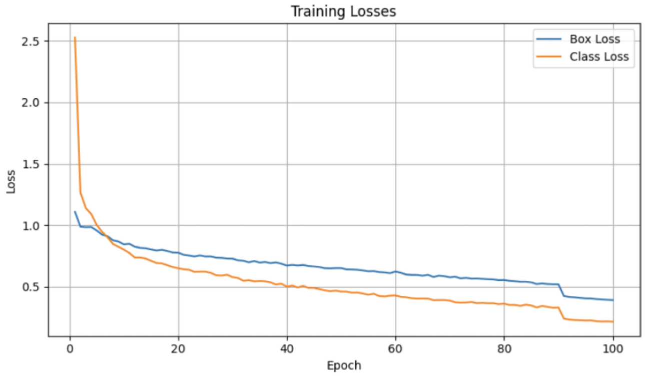

Fig. 6. Yolov8 Loss Plot

The training loss curves for YOLOv8, shown in (Fig. 6), depict a steady and smooth decline in both box loss and class loss over the course of 100 epochs. The box loss starts off higher than the class loss but decreases gradually, indicating consistent improvement in spatial localization accuracy. The class loss also shows a stable downward trend, suggesting effective learning of classification features. The absence of sharp oscillations or divergence between the two curves implies that the training process is stable and converging well. This performance, captured in Fig 7, supports the model's robustness in learning both object localization and classification simultaneously. The loss plots for the standard YOLOv9 model, shown in (Fig. 7), illustrate smooth convergence across all key

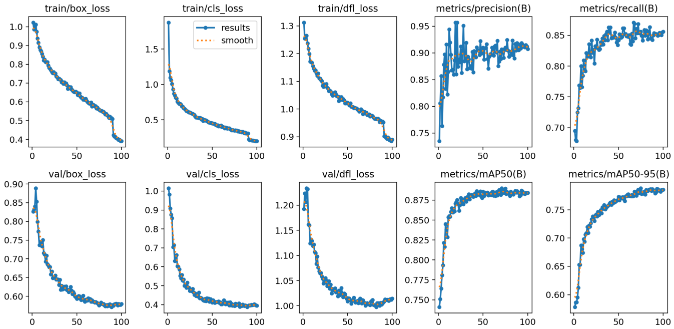

Fig. 7. Yolov9 Loss Plots

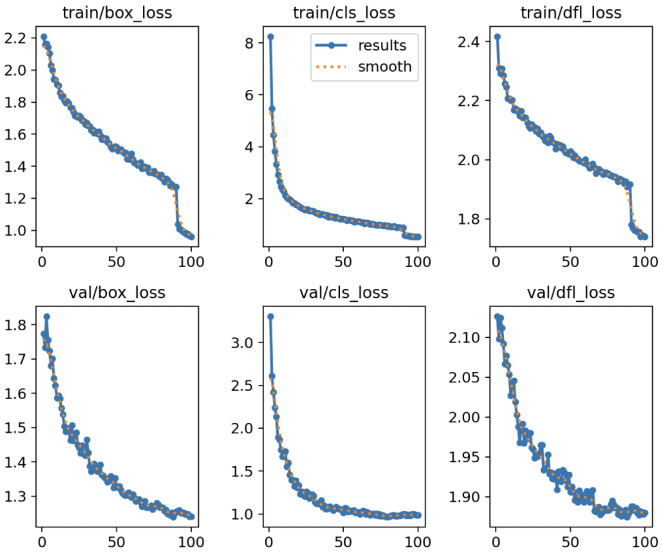

Fig. 8. Yolov10 Loss Plots

metrics: box loss, classification loss, and distribution focal loss (DFL) for both training and validation datasets. The model also demonstrates strong improvements in precision, recall, and mAP values throughout training, indicating a stable and well-generalized learning process. The performance curve of mAP@50–95 continues to rise steadily, confirming YOLOv9's capacity for fine-grained detection accuracy.

As illustrated in (Fig. 8), the YOLOv10 model exhibits a consistent decrease in training and validation losses. The model's distribution focal loss (DFL) steadily reduces, reinforcing effective bounding box regression. Moreover, the convergence patterns of precision and recall curves affirm that the model progressively improves in both classification and localization tasks. The smooth trajectory of the mAP@50–95 metric supports the model's reliability in multi-object settings.

(Fig. 9) presents the loss and performance curves for YOLOv11. The losses—box, classification, and DFL—all show a stable reduction, with minimal oscillations across both training and validation sets. The precision and recall metrics exhibit a sharp rise within the first 20 epochs, after which they plateau, suggesting the model reaches optimal performance relatively early. mAP@50 and mAP@50–95 metrics further confirm YOLOv11's generalization capability, making it a robust option for deployment.

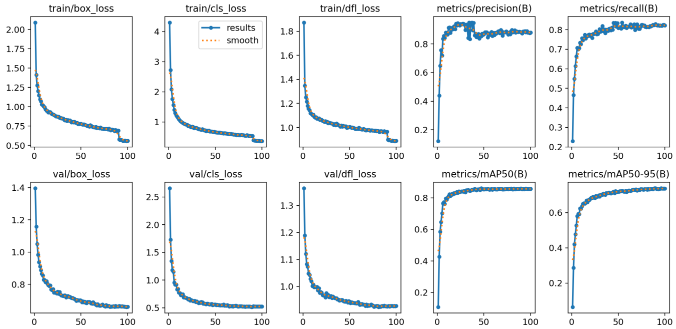

Fig. 9. Yolov11 Loss Plots

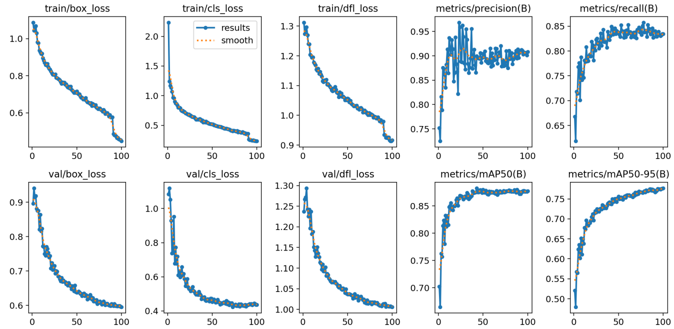

Fig. 10. Yolov12 Loss Plots

The performance of YOLOv12, as depicted in (Fig. 10), reveals a highly stable training regime with low variance between training and validation losses. The classification loss, DFL, and box loss converge steadily, while the evaluation metrics (precision, recall, and mAP scores) maintain a consistent upward trajectory. Notably, mAP@50–95 shows a smooth curve with minimal jitter, indicating that YOLOv12 generalizes exceptionally well to unseen samples and exhibits strong fine-grained prediction capacity.

The integration of CBAM attention modules into YOLOv9 (Fig. 11) shows improved convergence smoothness compared to the ResBlock variant. The validation losses drop rapidly and then plateau with minimal variance, suggesting that CBAM

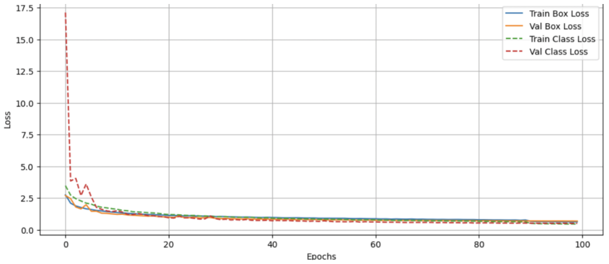

Fig. 11. Yolov9 Loss Plots with CBAM

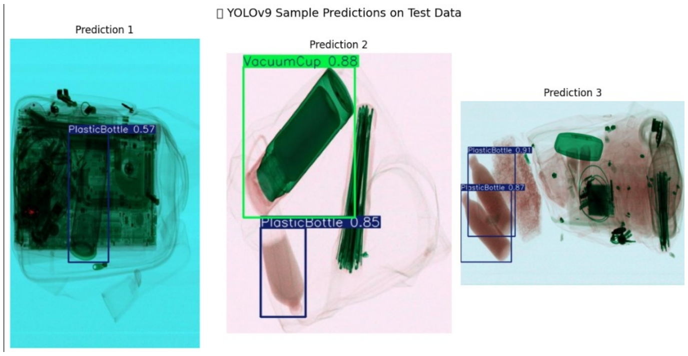

Fig. 12. Yolov9 with CBAM results

helps the network focus on relevant features more efficiently. This is especially evident in the alignment of train and val class loss curves, highlighting improved consistency between training and validation behavior.

V. Conclusion and Future Scope

The comparative evaluation of various YOLO models underscores the evolving trade-offs between model architecture, precision-recall dynamics, and detection efficacy across scales. Pre-trained models such as YOLOv5s and YOLOv12 demonstrated superior performance across several metrics, with YOLOv5s achieving the highest precision (0.858) and a robust F1-score of 0.735, while YOLOv12 balanced high precision (0.856) with commendable mAP@50 (0.703) and mAP@50–95 (0.600) values. These models benefit from extensive pre-training, which enhances their generalization and robustness across detection scenarios.

In contrast, our experiment introduced an untrained YOLOv9m variant augmented with Convolutional Block Attention Module (CBAM) to evaluate the impact of spatial and channel-wise attention mechanisms on raw learning capacity. Remarkably, the YOLOv9m (CBAM) model—despite not undergoing any form of pre-training—achieved a precision of 0.781 and an F1-score of 0.672. While it lags slightly in recall (0.621) and mAP@50–95 (0.548) compared to its pre-trained counterparts, the margins remain narrow, and its baseline performance is notable given the absence of prior training.

The integration of CBAM modules appears to enhance the model's ability to prioritize salient spatial and channel-level features, thereby compensating—at least partially—for the lack of pre-training. This suggests that attention-based architectural modifications can significantly amplify the learning potential of lightweight models when introduced at the design stage. Given these findings, the YOLOv9m (CBAM) variant represents a promising direction for real-time object detection tasks, particularly in scenarios with limited computational resources or where model initialization from scratch is required. With continued training and fine-tuning, it is plausible that the performance of this CBAM-enhanced model could converge with or even surpass that of standard pre-trained YOLO models.

Future work can explore integrating transformer-based modules to better capture long-range dependencies in images, potentially improving feature representation over CNN-only backbones. Incorporating advanced attention mechanisms beyond CBAM, such as dual attention or deformable attention, could further enhance focus on critical regions. Experimenting with hybrid architectures that combine CNNs and transformers may yield more robust detection and classification. Additionally, replacing or augmenting residual blocks with efficient convolutional variants like depthwise separable convolutions or bottleneck layers could reduce model complexity. Finally, exploring multi-scale feature fusion and improved loss functions will likely boost detection accuracy and model generalization.

References

[1] K. Wang, H. Du, and M. Xie, "Prohibited Items Detection in Xray Images Based on Task Decoupling YOLOv5," in *Proc. IEEE Int. Conf. Commun. China (ICCC)*, 2023, pp. 1733–1737, doi: 10.1109/ICCC59590.2023.10507633.

[2] A. Wang, P. Yuan, H. Wu, Y. Iwahori, and Y. Liu, "Improved YOLOv8 for Dangerous Goods Detection in X-ray Security Images," *Electronics*, vol. 13, no. 16, p. 3238, 2024, doi: 10.3390/electronics13163238.

[3] "Prohibited Object Detection using YOLOv7 on X-Ray Images of Airport Luggages (OPIXray)," *IEEE Conf. Publ.*, Jul. 6, 2023. Available: https://ieeexplore.ieee.org/document/10306920.

[4] "PIXDET: Prohibited item detection in X-Ray image based on Whole-Process feature fusion and Local–Global semantic dependency interaction," *IEEE J. Mag.*, 2023. Available: https://ieeexplore.ieee.org/document/10309173.

[5] "ForkNet: Overlapping image disentanglement for accurate prohibited item detection," *IEEE J. Mag.*, 2024. Available: https://ieeexplore.ieee.org/document/10510166.

[6] M. Alansari et al., "Multi-Scale Hierarchical Contour Framework for Detecting Cluttered Threats in Baggage Security," *IEEE Access*, pp. 1–1, 2024, doi: 10.1109/ACCESS.2024.3407720.

[7] C. Zhao, L. Zhu, S. Dou, W. Deng, and L. Wang, "Detecting Overlapped Objects in X-Ray Security Imagery by a Label-Aware Mechanism," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 998–1009, 2022, doi: 10.1109/TIFS.2022.3154287.

[8] D. Gupta et al., "Security threats on Data-Driven Approaches for Luggage Screening," in *Proc. IEEE Comput. Commun. Workshop Conf. (CCWC)*, 2024, pp. 474–483, doi: 10.1109/CCWC60891.2024.10427869.

[9] J. K. Dumagpi and Y. Jeong, "Evaluating GAN-Based Image Augmentation for Threat Detection in Large-Scale X-ray Security Images," *Appl. Sci.*, vol. 11, no. 1, p. 36, 2020, doi: 10.3390/app11010036.

[10] A. Wang, P. Yuan, H. Wu, Y. Iwahori, and Y. Liu, "Improved YOLOv8 for Dangerous Goods Detection in X-ray Security Images," *Electronics*, vol. 13, no. 16, p. 3238, 2024, doi: 10.3390/electronics13163238.

[11] H. Sima, B. Chen, C. Tang, Y. Zhang, and J. Sun, "Multi-Scale Feature Attention-DEtection TRansformer: Multi-Scale Feature Attention for Security Check Object Detection," *IET Comput. Vis.*, vol. 18, 2024, doi: 10.1049/cvi2.12267.

[12] A. Wang, P. Yuan, H. Wu, Y. Iwahori, and Y. Liu, "Improved YOLOv8 for Dangerous Goods Detection in X-ray Security Images," *Electronics*, vol. 13, no. 16, p. 3238, 2024, doi: 10.3390/electronics13163238.

[13] "Detection of Sharp Objects using Deep Neural Network-Based Object Detection Algorithm," *IEEE Conf. Publ.*, Sep. 28, 2020. Available: https://ieeexplore.ieee.org/document/9315272.

[14] B. Hu, C. Zhang, L. Wang, Q. Zhang, and Y. Liu, "Multi-Label X-ray Imagery Classification via Bottom-Up Attention and Meta Fusion," in *Proc. Asian Conf. Comput. Vis. (ACCV)*, 2020. Available: https://openaccess.thecvf.com/content/ACCV2020/html/HuMulti-labelX-rayImageryClassificationviaBottom-upAttentionandMetaFusionACCV2020paper.html.

[15] N. Gan et al., "YOLO-CID: Improved YOLOv7 for X-ray Contraband Image Detection," *Electronics*, vol. 12, p. 3636, 2023, doi: 10.3390/electronics12173636.

[16] Y. Liu, E. Zhang, X. Yu, and A. Wang, “Efficient X-ray Security Images for Dangerous Goods Detection Based on Improved YOLOv7,” *Electronics*, vol. 13, no. 1530, 2024, doi: 10.3390/electronics13081530.

[17] T. Morris, T. Chien, and E. Goodman, “Convolutional Neural Networks for Automatic Threat Detection in Security X-Ray Images,” in *Proc. IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, 2018, pp. 285–292, doi: 10.1109/ICMLA.2018.00049.

[18] W. Liu et al., “ABTD-Net: Autonomous Baggage Threat Detection Networks for X-ray Images,” in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, 2023, pp. 1229–1234, doi: 10.1109/ICME55011.2023.00214.

[19] Y. Zhang, L. Zhuo, C. Ma, Y. Zhang, and J. Li, “CTA-FPN: Channel-Target Attention Feature Pyramid Network for Prohibited Object Detection in X-ray Images,” *Sensing Imaging*, vol. 24, no. 14, 2023, doi: 10.1007/s11220-023-00416-7.

[20] V. R. Ambati and A. Borthakur, “A Study on Tiny YOLO for Resource Constrained X-ray Threat Detection,” in *Proc. Int. Conf. Artif. Intell. Mach. Learn. Syst. (AIMLSystems)*, 2023, pp. 1–5, doi: 10.1145/3639856.3639899.

[21] CLCXRay Object Detection Dataset by Didenegar. (2024, August 28). Roboflow. https://universe.roboflow.com/didenegar-zrt8i/clcxray-vldcf