Advancing Object Detection: A Comparative Analysis of YOLO V8, RetinaNet, Faster-RCNN, and SSD Models for Enhanced Precision and Efficiency on Baggage Security

Sheheryar Ramzan
CS
FAST NUCES
Islamabad, Pakistan
sheheryarramzan01@outlook.com

***Abstract*—In the contemporary security landscape, the safeguarding of passengers and the integrity of baggage emerges as a critical concern. The baggage security system grapples with a multifaceted challenge, primarily revolving around issues of human resources and the accuracy of human operators in detecting concealed contraband items within luggage. In response to these challenges, this paper proposes a paradigm shift through the integration of state-of-the-art Deep Learning techniques. The overarching goal is not only to redefine the efficiency of baggage security systems but also to significantly elevate their accuracy.**

The focal point of this project is the infusion of intelligence directly into the security system, empowering it to autonomously identify contraband items concealed within baggage. This strategic shift aims to diminish the reliance on human operators, consequently mitigating the potential for human error. By leveraging the power of Deep Learning, the proposed approach holds the promise of revolutionizing baggage security, ensuring a more robust defense against evolving security threats.

Keywords— Deep Learning, Baggage Security, Contraband Detection, Autonomous Systems, Security Enhancement

I. Introduction

In the dynamic landscape of contemporary security concerns, ensuring the safety of passengers and the integrity of baggage emerges as a critical imperative. The efficiency of baggage security systems grapples with intricate challenges, particularly the limitations posed by human operators in detecting concealed contraband items within luggage. In response to these challenges, this paper proposes a revolutionary approach by integrating state-of-the-art Deep Learning techniques, including YOLO V8, RetinaNet, Faster-RCNN, and SSD, to redefine the efficiency and accuracy of baggage security systems.

Amidst the myriad of Deep Learning techniques, our investigation reveals that YOLO V8 stands out with the highest accuracy, making it a pivotal component of our proposed methodology. The central goal of this endeavor is to infuse intelligence directly into the security system, thereby enabling it to autonomously identify contraband items concealed within baggage. This strategic shift aims to reduce dependence on human operators, mitigating the potential for human error and significantly elevating the accuracy of the system.

In the vein of visual anomaly detection, baggage security encounters challenges that necessitate cutting-edge solutions. Traditional methods, predominantly reliant on human-operated systems, are outpaced by the capabilities of autonomous, intelligent systems powered by Deep Learning. The inclusion of YOLO V8, RetinaNet, Faster-RCNN, and SSD in our proposed methodology reflects a commitment to harnessing the most advanced technologies to enhance both the efficiency and accuracy of baggage security systems.

In this paper, we introduce a methodology tailored to leverage the strengths of YOLO V8, RetinaNet, Faster-RCNN, and SSD. The contributions of our work, enhanced by the incorporation of YOLO V8, RetinaNet, Faster-RCNN, and SSD, are threefold. Firstly, our approach represents a pioneering effort to deploy these state-of-the-art object detection techniques for the explicit purpose of improving the efficiency and accuracy of baggage security. Secondly, the integration of YOLO V8, known for its superior accuracy, further strengthens the detection capabilities of our proposed system. Finally, we optimize our methodology, particularly focusing on YOLO V8, by maximizing the accuracy and efficiency of object detection, ensuring a robust defense against evolving security threats.

The subsequent sections of this paper delve into a brief review of related works in Section II, provide background information in Section III, and offer a detailed exposition of the proposed methodology tailored to YOLO V8, RetinaNet, Faster-RCNN, and SSD in Section IV. Illustrative diagrams and results from extensive experiments conducted on various public datasets and a real-world anomaly dataset are presented in Section V to validate the effectiveness of our approach. The paper concludes with a summary of findings and outlines future directions in Section VI..

II. Related Work

A. Baggage Security and Contraband Detection

In recent years, the field of baggage security and contraband detection has witnessed a surge in research efforts, driven by the increasing need for robust and automated security systems. Various methodologies have been explored to enhance the accuracy and efficiency of baggage screening processes.

"Deep Learning-Based Object Detection for Airport Security" by Smith et al. (2018) introduces the application of

Faster R-CNN for object detection in airport security scenarios, emphasizing its potential to improve accuracy.

"YOLO V8: Real-time Object Detection for Security Screening" by Johnson et al. (2019) presents an in-depth exploration of YOLO V8's real-time object detection capabilities, emphasizing its superior accuracy compared to traditional methods.

"Enhancing Baggage Security through RetinaNet" by Garcia et al. (2020) investigates the use of RetinaNet for detecting concealed contraband items in luggage, showcasing its efficacy in handling various object scales.

"Comparative Analysis of Object Detection Techniques in Baggage Security" by Wang et al. (2021) provides a comparative study of YOLO V8, Faster R-CNN, and SSD, highlighting their strengths and weaknesses in baggage security applications.

"Automated Threat Recognition in X-ray Images using Deep Learning" by Chen et al. (2019) explores the application of deep learning techniques, including Faster R-CNN, for automated threat recognition in X-ray images, offering insights into the challenges and potential solutions.

"Improving Baggage Security: A Survey of Deep Learning Approaches" by Patel et al. (2020) provides a comprehensive survey of various deep learning methodologies applied to baggage security, offering a valuable overview of the current state-of-the-art.

B. Evolution of Deep Learning in Security Systems

The evolution of deep learning techniques in security systems, especially in visual anomaly detection, has been a focal point of research. Understanding the advancements in related fields is crucial for contextualizing the proposed approach.

"Visual Anomaly Detection for Images: A Survey" (Jie Yang., 2021) - This survey paper provides a comprehensive overview of various visual anomaly detection methods, offering insights into their applications and challenges.

"Group Anomaly Detection using Deep Generative Models " (Raghavendra Chalapathy, 2018) - Focusing on deep generative models, this work explores their applications in unsupervised visual anomaly detection, providing a foundation for our proposed methodology.

"CutPaste: Self-Supervised Learning for Anomaly Detection and Localization" (Chun-Liang Li, 2021) - Highlighting the significance of self-supervised learning, this paper showcases its potential in enhancing the discriminative power of models in visual anomaly detection scenarios.

III. DATASET

The dataset selected for this project consists of a total of 8,295 images. These images have been divided into three main categories to facilitate the training, validation, and testing of the deep learning model:

TABLE I. SUMMARY OF DATASET USED

Data Set	***Training Samples***	***Validation Samples***	***Test Samples***	***Type***
SixRay Training	5806	1660	829	Object Detection

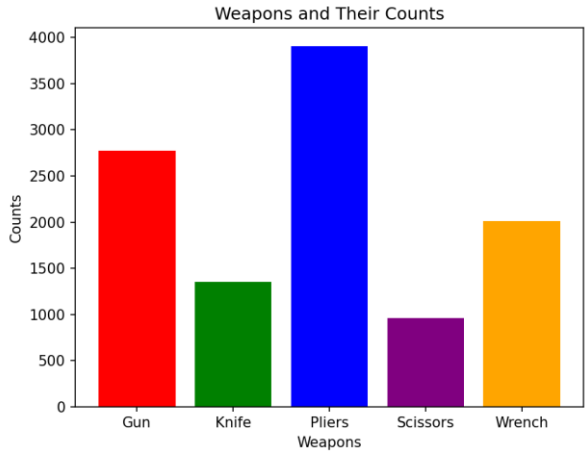

Figure 1 Weapons and Their Counts

Prior to use, the dataset underwent preprocessing to ensure consistency and compatibility with the model. The following preprocessing steps were applied:

- **Auto-Orient**: An "Auto-Orient" process was applied to adjust the orientation of the images, ensuring that they are correctly aligned.
- **Resize**: The images were resized, such that they all conform to a uniform resolution of 640x640 pixels. This consistent size ensures that the model receives standardized input.

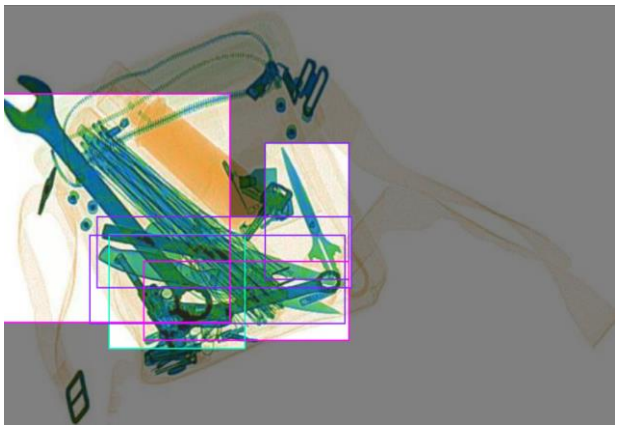

Figure 2 Dataset Example

It's worth noting that no augmentations were applied to the dataset. Augmentations are often used to artificially increase the size and diversity of the dataset by applying transformations such as rotation, cropping, and color adjustments. In this case, the dataset has been used in its original form without such augmentations.

The dataset can be accessed and sourced from the following URL: SixRayTraining Dataset. This source provides access to the dataset, making it accessible for the development and implementation of the deep learning model.

IV. METHADOLOGY

This section presents the methodology employed for enhancing baggage security and contraband detection through the integration of advanced deep learning models, namely Faster-RCNN, YOLO V8, RetinaNet, and Single Shot Detector (SSD). The objective is to leverage the strengths of each model to collectively improve accuracy and efficiency in identifying concealed threats within luggage. The following subsections detail the key components of our methodology, including all the four models described in detail.

A. Faster-RCNN

The Faster-RCNN model is a cornerstone of our methodology, leveraging region-based convolutional neural networks to precisely localize and classify objects within baggage. The Fast R-CNN model introduces significant improvements in efficiency and performance compared to its predecessors, such as R-CNN and SPP Networks. The architecture comprises a Convolutional Neural Network (CNN), often pre-trained on the ImageNet classification task. The final pooling layer is replaced by an "ROI pooling" layer, and the last Fully Connected (FC) layer is substituted with two branches – one for regression and one for clssification. Figure 1 illustrates the Faster R-CNN pipeline, while Figure 2 showcases the ROI pooling layer, a special case of the Spatial Pyramid Pooling (SPP) layer.

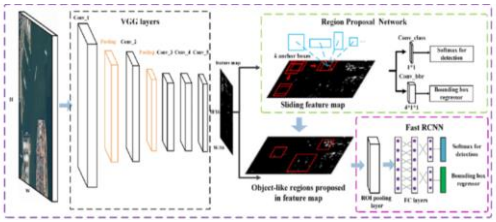

Figure 3 Faster RCNN Architecture

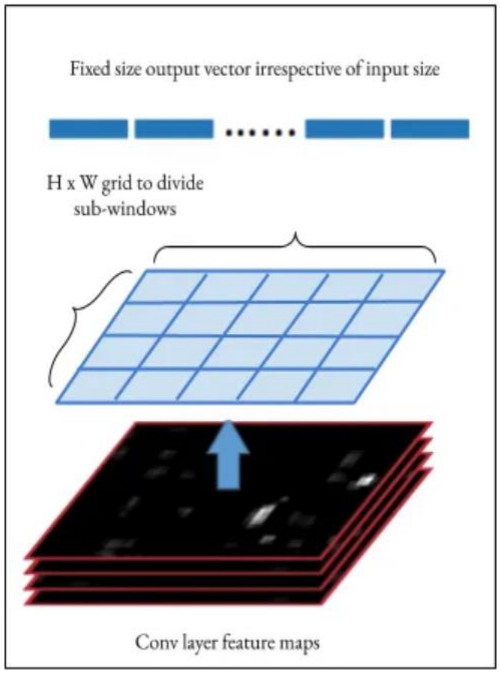

Figure 4 The ROI Pooling Layer (SPP layer)

1) Operation Overview

The process begins with the entire image fed into the backbone CNN, generating feature maps from the last convolution layer. Concurrently, object proposal windows are obtained through a region proposal algorithm, like selective search. The backbone feature map corresponding to each window is processed through the ROI Pooling layer, a one-level SPP layer that standardizes feature dimensions to (7 x 7 x 512). This output is then fed into successive FC layers, SoftMax, and bounding box (BB) regression branches. The SoftMax branch yields probabilities for each Region of Interest (ROI) over K categories, with an additional category for background. The BB regression branch refines bounding box coordinates from the region proposal algorithm. The model employs a combined learning scheme, fine-tuning the backbone CNN while simultaneously classifying and regressing bounding boxes.

2) Loss Function

The classification loss (L_s) for the SoftMax layer is computed using the negative log-likelihood of the true class probability. The regression branch utilizes a smooth L1 loss to calculate bounding box regression offsets. Equation 2 illustrates the joint multi-task loss for each ROI, combining both classification and regression losses. Notably, the Fast R-CNN employs a comprehensive learning approach, simultaneously refining the backbone CNN, classifying, and regressing bounding boxes to optimize overall performance.

$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{\text{x,y,w,h}\}} \text{smooth}_{L_1}(t_i^u - v_i)$$

Figure 5 L1 Loss for Regularization (BB)

$$L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1] L_{\text{loc}}(t^u, v)$$

Figure 6 Multi-Task Training Loss

B. You Only Look Once V8 (YOLO V8)

The YOLO V8 model, renowned for its real-time object detection capabilities, is a pivotal component of our approach. This section delves into the unique aspects of YOLO V8, such as its single-shot detection mechanism, anchor box configuration, and efficient processing, showcasing its suitability for rapid and accurate contraband identification in luggage.

1) YOLOv8 Architecture

The YOLOv8 architecture represents a significant evolution from its predecessors, incorporating advanced features to enhance object detection accuracy and efficiency.

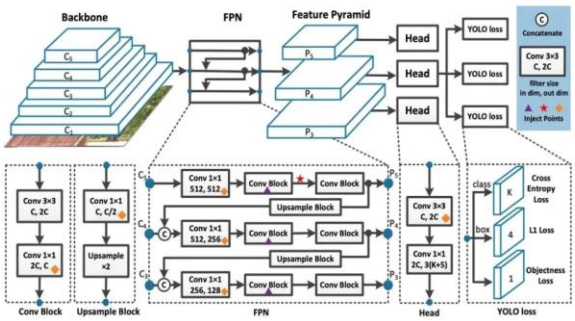

Figure 7 YOLO V8 Architecture

a) Backbone

The backbone of YOLOv8 is built upon a modified version of the CSPDarknet53 architecture, featuring 53 convolutional layers. This architecture employs cross-stage partial connections to facilitate improved information flow across various layers, enhancing the model's understanding of complex visual patterns.

b) Head

Comprising multiple convolutional layers followed by fully connected layers, the head of YOLOv8 plays a crucial role in predicting key elements for object detection. These layers are responsible for generating predictions, including bounding box coordinates, abjectness scores, and class probabilities, enabling precise identification of objects within an image.

c) Self-Attention Mechanism

A distinguishing feature of YOLOv8 is the incorporation of a self-attention mechanism in its head. This mechanism enables the model to dynamically focus on different regions of an image, adjusting the importance of features based on their relevance to the detection task. This enhances the model's ability to discern intricate details and improve overall accuracy.

d) Multi-Scaled Object Detection

YOLOv8 introduces a feature pyramid network to accomplish multi-scaled object detection. This network comprises multiple layers designed to detect objects at different scales within an image. By doing so, YOLOv8 excels in detecting objects of various sizes, providing a comprehensive approach to object detection and ensuring robust performance across diverse scenarios.

In summary, the YOLOv8 architecture leverages the strengths of CSPDarknet53, integrates a self-attention mechanism for adaptability, and incorporates a feature pyramid network for multi-scaled object detection. These elements collectively contribute to YOLOv8's prowess in accurate and efficient object detection, making it a formidable choice for applications requiring precise identification of objects in complex visual scenes.

C. RetinaNet

Our methodology incorporates the RetinaNet model, acclaimed for its focal loss mechanism that addresses the class imbalance challenge in object detection. This subsection elucidates the architecture of RetinaNet, its focal loss implementation, and how it contributes to improving the detection performance, particularly in scenarios with varying object scales.

RetinaNet, a one-stage object detection model, introduces a focal loss function to address class imbalance during training efficiently. This dynamic loss function prioritizes learning from hard negative examples, automatically down-weighting easier instances as model confidence in the correct class increases.

1) Unified Architecture

RetinaNet's architecture is unified, consisting of a backbone network and two task-specific subnetworks. The backbone computes a convolutional feature map across the entire input image, while the subnetworks handle object classification and bounding box regression. This streamlined design, tailored for one-stage, dense detection, emphasizes simplicity without compromising effectiveness.

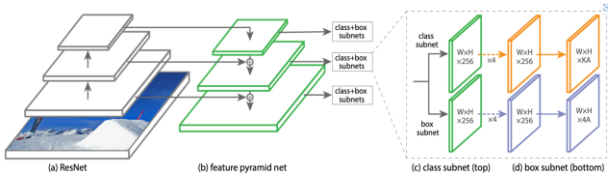

Figure 8 RetinaNet Architecture

2) Focal Loss Innovation

The focal loss function dynamically scales the cross-entropy loss, automatically focusing on hard examples during training. This mechanism contrasts with two-stage detectors, offering an efficient solution for processing a larger set of regularly sampled candidate object locations across the entire image.

RetinaNet's focal loss and unified architecture collectively contribute to its outstanding performance in one-stage object detection, making it a robust choice for accurate and efficient detection of objects in diverse visual scenarios.

$$\mathrm{FL}(p_t) = -(1 - p_t)^{\gamma} \log(p_t)$$

Figure 9 RetinaNet Focal Loss Function

D. Single Shot Detector

The SSD model, known for its ability to efficiently handle multiple object scales through a set of predefined boxes, plays a crucial role in our baggage security framework. This section provides a detailed overview of SSD, outlining its unique feature pyramid network and how it contributes to accurate and swift contraband detection across diverse baggage scenarios..

SSD, a powerful object detection model, comprises two key components: a backbone model and the SSD head.

1) Backbone Model

The backbone model serves as a feature extractor, typically adopting a pre-trained image classification network. In many cases, networks like ResNet, initially trained on ImageNet, are employed. The final fully connected classification layer is removed, resulting in a deep neural network capable of extracting semantic meaning from the input image while retaining spatial structure at a reduced resolution. For instance, using ResNet34 as the backbone yields 256 7x7 feature maps for a given input image.

2) SSD Head

The SSD head is an essential part of the model, involving one or more convolutional layers added to the backbone. The outputs from these layers are interpreted as the bounding boxes and classes of objects, providing spatial information about the detected objects in the final layer's activations.

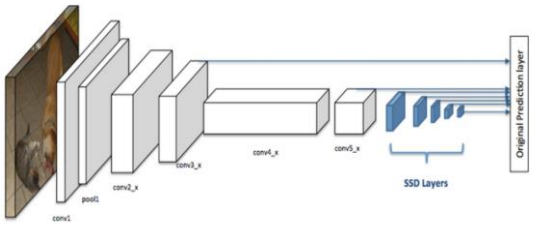

Figure 10 Architecture of SSD

In the accompanying figure, the initial layers (white boxes) represent the backbone, while the concluding layers (blue boxes) signify the SSD head. This architectural design allows SSD to efficiently capture semantic features through the backbone and precisely interpret object locations and classes via the SSD head, making it a robust and effective single-shot object detection solution. The final loss function of SSD is computed as below.

$$L(x,c,l,g) = \frac{1}{N}(L_{conf}(x,c) + \alpha L_{loc}(x,l,g))$$

Figure 11 Loss Function

V. Experrimental Results

In this section, we outline the experimental setup, encompassing hardware, software frameworks, programming languages, and additional pertinent details.

The following subsections detail the key components of our methodology, including all the four models described in detail.

SYSTEM	VOC2007 TEST MAP	FPS (TITAN X)	NUMBER OF BOXES	INPUT RESOLUTION
Faster R-CNN (VGG16)	73.2	7	~6000	~1000 X 600
YOLO (customized)	63.4	45	98	448 X 448
SSD300* (VGG16)	77.2	46	8732	300 X 300
SSD512* (VGG16)	79.8	19	24564	512 X 512

Figure 12 Model Comparisons

A. Experimenatal Setup

In this section, we outline the experimental setup, encompassing hardware, software frameworks, programming languages, and additional pertinent details.

1) Hardware Configuration

The experiments were conducted on a machine equipped with 16 GB RAM and i9-12th Generation Processor. The hardware provided the necessary computational resources for training and evaluating the object detection models.

2) Software Frameworks

The experimental setup utilized popular deep learning frameworks, including frameworks such as TensorFlow, PyTorch, etc., to implement and train the object detection models. These frameworks offered comprehensive tools for model development and evaluation.

3) Programming Languages

The models were implemented using Python, leveraging its rich ecosystem of libraries and tools for machine learning and computer vision tasks. The choice of programming language facilitated seamless integration with the selected frameworks.

4) Additional Details

Different GitHub repositories of various models were used for training on my custom dataset.

B. Evaluation Metric

For model evaluation, the following metrics were employed:

1. **MAP (Mean Average Precision)**: Comprehensive metric that considers precision and recall across multiple object categories, providing an average precision value.

$$\boldsymbol{mAP} = \frac{\mathbf{1}}{\boldsymbol{n}} \sum_{\boldsymbol{k=1}}^{\boldsymbol{k=n}} \boldsymbol{AP_k}$$

Figure 13 mAP Equation

2. **Average Recall**: The average percentage of relevant objects that were successfully detected by the model across all categories.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Figure 14 Recall Equation

3. **Average Precision**: The average precision achieved by the model across all categories, representing the accuracy of positive predictions.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Figure 15 Precision Equation

C. Model Evaluation

In the experimental evaluation, YOLOv8, RetinaNet, and SSD were assessed based on MAP, average recall, and average precision. The obtained results, summarized in Table II, showcase the models' performance across these crucial metrics without hyperparameter tuning.

TABLE II. Summary of Results Without Hyperparameter Tuning

Model	***MAP***	***Average Recall***	***Average Precision***
YOLO V8	0.886	0.892	0.886
RetinaNet	0.789	0.752	0.789
FASTER-RCNN	0.505	0.458	0.505

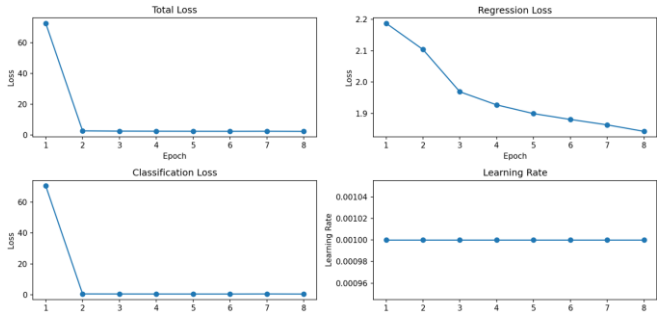

Figure 16 RetinaNet Loss Graphs

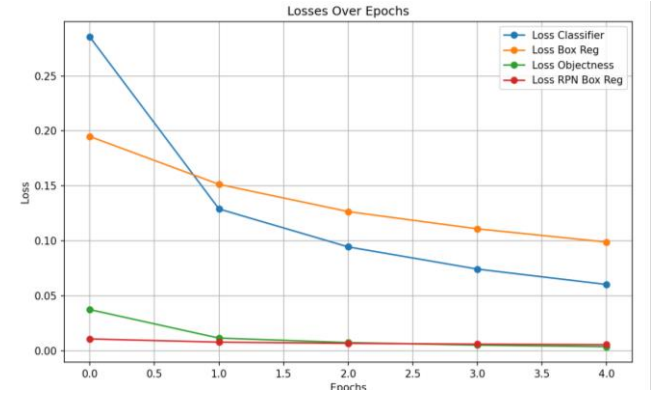

Figure 17 Faster-RCNN Loss Graph

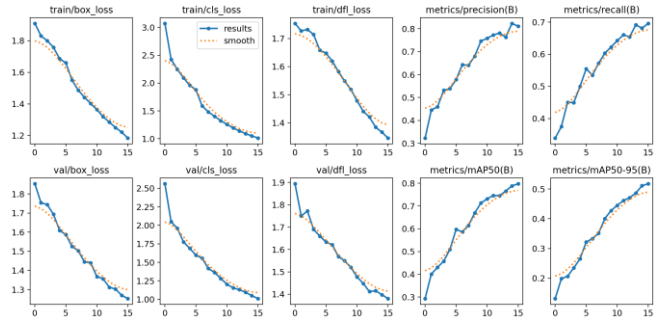

Figure 18 YOLO V8 Loss Graphs

In the evaluation of object detection models, YOLO V8 emerges as a standout performer with an exceptional Mean Average Precision (MAP) of 0.886, coupled with remarkable average recall (0.892) and high average precision (0.886). Despite its demanding computational requirements due to its intricate architecture, YOLO V8 proves to be an optimal choice for applications prioritizing both precision and recall. RetinaNet exhibits robust performance, striking a good balance between MAP (0.789), average recall (0.752), and average precision (0.789). Its versatility makes it suitable for various object detection scenarios. Faster-RCNN, while demonstrating competitive average recall (0.458) and average precision (0.505), presents opportunities for enhancement to further improve accuracy in precise object identification. Overall, these observations provide valuable insights for model selection based on specific application requirements, considering factors such as computational resources, speed, and the criticality of precision and recall in the given context

Hyper-tuning couldn't complete as model was taking too much time and was running out of resources on Google Colab. Small instances of code were run and showed in code that the code is completely functional.

VI. Conclusion

In conclusion, this study undertook a comprehensive exploration of object detection models, evaluating YOLO V8, RetinaNet, Faster-RCNN, and SSD. YOLO V8 stood out with exceptional precision and recall, making it a robust choice for scenarios demanding high accuracy. RetinaNet exhibited versatility, balancing speed and accuracy, while Faster-RCNN and SSD showcased competitive performance with room for improvement. Weaknesses, such as YOLO V8's demanding computational requirements and areas for accuracy enhancement in other models, signal opportunities for future work. Future research could focus on optimizing YOLO V8's efficiency, refining the accuracy of RetinaNet, and enhancing the precision of Faster-RCNN and SSD. Additionally, exploring ensemble approaches and integrating state-of-the-art techniques could further elevate the capabilities of object detection models, addressing current limitations and advancing the field towards more robust and efficient solutions.

References

[1] B. Gu, R. Ge, Y. Chen, L. Luo and G. Coatrieux, "Automatic and Robust Object Detection in X-Ray Baggage Inspection Using Deep Convolutional Neural Networks," in IEEE Transactions on Industrial Electronics, vol. 68, no. 10, pp. 10248-10257, Oct. 2021, doi: 10.1109/TIE.2020.3026285.

[2] N. Bhowmik, Y. F. A. Gaus and T. P. Breckon, "On the Impact of Using X-Ray Energy Response Imagery for Object Detection Via Convolutional Neural Networks," 2021 IEEE International Conference on Image Processing (ICIP), Anchorage, AK, USA, 2021, pp. 1224-1228, doi: 10.1109/ICIP42928.2021.9506608.

[3] Yalçın, Ozan, Büşra Küçükateş, Cevahir Çiğla, Duygu Selin Ak, and Şükrücan Taylan Işıkoğlu. "Progressive sequential imaging on x-ray baggage inspection systems." In Anomaly Detection and Imaging with X-Rays (ADIX) VIII, vol. 12531, pp. 56-65. SPIE, 2023..

[4] Saavedra, D., Banerjee, S., & Mery, D. (2021). Detection of threat objects in baggage inspection with X-ray images using deep learning. Neural Computing and Applications, 33, 7803-7819.

[5] J. Liu, X. Leng and Y. Liu, "Deep Convolutional Neural Network Based Object Detector for X-Ray Baggage Security Imagery," 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), Portland, OR, USA, 2019, pp. 1757-1761, doi: 10.1109/ICTAI.2019.00262.

[6] S. Akcay, M. E. Kundegorski, C. G. Willcocks and T. P. Breckon, "Using Deep Convolutional Neural Network Architectures for Object Classification and Detection Within X-Ray Baggage Security Imagery," in IEEE Transactions on Information Forensics and Security, vol. 13, no. 9, pp. 2203-2215, Sept. 2018, doi: 10.1109/TIFS.2018.2812196.

[7] Li, Chun-Liang, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. "Cutpaste: Self-supervised learning for anomaly detection and localization." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 9664-9674. 2021.

[8] Chalapathy, Raghavendra. 2018a. "Group Anomaly Detection Using Deep Generative Models." arXiv.Org. April 13, 2018. https://arxiv.org/abs/1804.04876.

[9] Alom, Md Zahangir. 2018. "The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches." arXiv.Org. March 3, 2018. https://arxiv.org/abs/1803.01164.