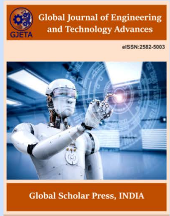
Global Journal of Engineering and Technology Advances

eISSN: 2582-5003

Cross Ref DOI: 10.30574/gjeta

Journal homepage: https://gjeta.com/



(Review Article)

Check for updates

Comparative analysis of hybrid cognitive radar techniques for enhanced target detection and tracking: A performance evaluation perspective

Obiajulu C. Emmanuel *, Aliyu Sabo, Isa M. Danjuma and Sagir Lawan

Department of Electrical/Electronic Engineering, Nigeria Defence Academy, Kaduna, Nigeria.

Global Journal of Engineering and Technology Advances, 2024, 21(01), 119–129

Publication history: Received on 03 September 2024; revised on 16 October 2024; accepted on 19 October 2024

Article DOI: https://doi.org/10.30574/gjeta.2024.21.1.0189

Abstract

This paper introduces an innovative hybrid algorithmic approach for cognitive radar systems, by integrating unique machine learning optimization techniques such as, YOLO (You Only Look Once), Mask R-CNN, and Recurrent Neural Networks (RNNs). Through extensive simulations, the integrated approach demonstrates notable enhancements in target detection, instance segmentation, and target tracking within radar systems. Leveraging deep learning models, the framework facilitates adaptive and intelligent processing of radar data, augmenting system performance in dynamic environments. By seamlessly integrating state-of-the-art techniques, the proposed framework showcases a comprehensive solution to the challenges faced by traditional radar systems. This research represents a significant stride forward in radar technology, promising transformative impacts across various domains, including surveillance, remote sensing, and autonomous navigation. As radar systems continue to evolve, the adoption of advanced deep learning techniques offers unprecedented opportunities for enhancing situational awareness and decision-making capabilities in complex operational scenarios.

Keywords: Cognitive radar; Deep learning; YOLO; Mask R-CNN; Recurrent Neural Networks; Target detection; Target tracking

1. Introduction

Radar systems play a pivotal role in modern sensing applications, ranging from civilian air traffic control to military surveillance. As demands for increased accuracy, adaptability, and robustness continue to grow, researchers are exploring innovative approaches to enhance the capabilities of radar systems. One promising avenue involves the integration of hybrid approaches within the domain of cognitive radar, a paradigm that endows radar systems with learning and adaptive reasoning capabilities[1][2].

In traditional radar systems, the trade-offs between detection range, resolution, and target tracking accuracy often necessitate tailored solutions for specific scenarios[3]. However, the dynamic and diverse nature of contemporary operational environments demands a more versatile and responsive approach. The synergy of hybrid methodologies, combining diverse radar technologies, sensor modalities, and adaptive algorithms, holds the potential to revolutionize target detection and tracking in cognitive radar systems[4].

This paper addresses the imperative to advance radar capabilities through the exploration of hybrid approaches within the cognitive radar framework. By amalgamating traditional radar techniques with emerging technologies, our aim is to devise a comprehensive framework that not only adapts to varying operational scenarios but also significantly enhances the accuracy and efficiency of target detection and tracking. The integration of machine learning algorithms further augments the cognitive capabilities, enabling the radar system to learn from and adapt to evolving threats and environmental conditions[5]–[7].

* Corresponding author: Obiajulu C. Emmanuel

In the subsequent sections, we delve into the current state of radar systems, highlight the limitations of existing methodologies, and present our proposed hybrid cognitive radar framework. Our contributions extend beyond incremental improvements, aiming to address critical challenges in target detection and tracking through an innovative synthesis of diverse techniques. As we embark on this exploration, the potential impact of our work extends to applications ranging from defense and surveillance to autonomous systems, laying the foundation for a new era in radar technology[8].

2. Hybrid Algorithms used in Cognitive Radar

2.1. Particle Filter with Kalman Filter Fusion

The integration of Particle Filter (PF) and Kalman Filter (KF), known as Particle Filter with Kalman Filter Fusion, offers a powerful hybrid algorithm for target detection and tracking in cognitive radar systems. PF excels in handling nonlinearities and uncertainties, crucial in scenarios with complex target dynamics, while KF provides enhanced estimation accuracy in linear regimes[9]. By fusing the strengths of both filters, the hybrid algorithm achieves robust target tracking across a wide range of environments. In cognitive radar, this hybrid approach enables adaptive target tracking by seamlessly transitioning between PF and KF based on the nature of the target motion and environmental conditions. This adaptive capability enhances the radar system's resilience to dynamic and uncertain scenarios, making it a valuable tool for cognitive radar applications[10].

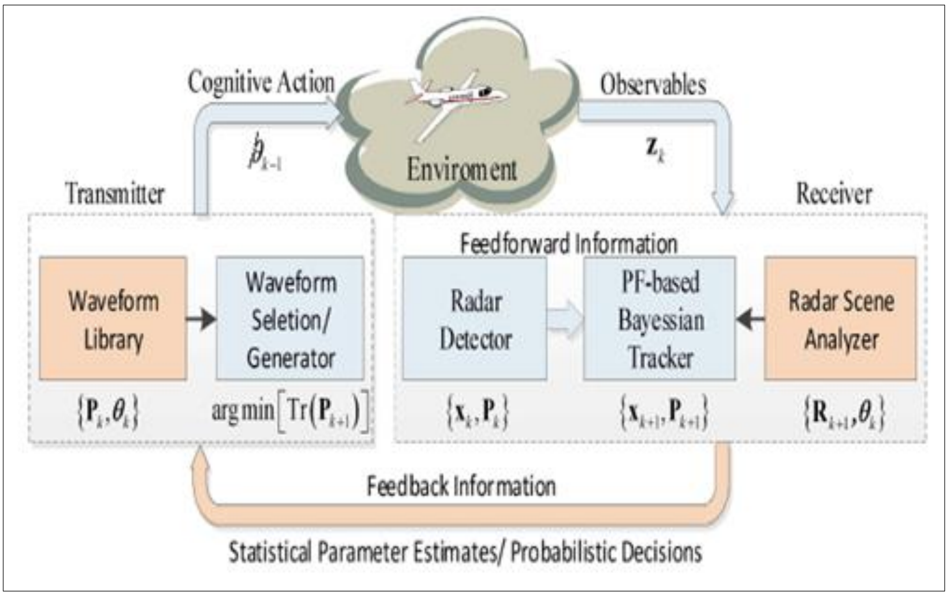


Figure 1a Cognitive radar system for target tracking[9]

We employ SPF (Standard Particle Filter), CRPF (Cost-Reference Particle Filter), and CCRPF (Cognitive Cost-Reference Particle Filter) for target tracking within a two-dimensional space, aiming for performance evaluation. Figure 2.2 illustrates the true path alongside the tracks generated by these filters. While the other two filters exhibit tracking errors smaller than SPF, CCRPF marginally outperforms CRPF. It's notable how the trajectory of CCRPF slightly diverges from the true path, attributable to the influence of the cost function and cognitive structure. EKF, however, does not demonstrate robust performance in this scenario.[9].

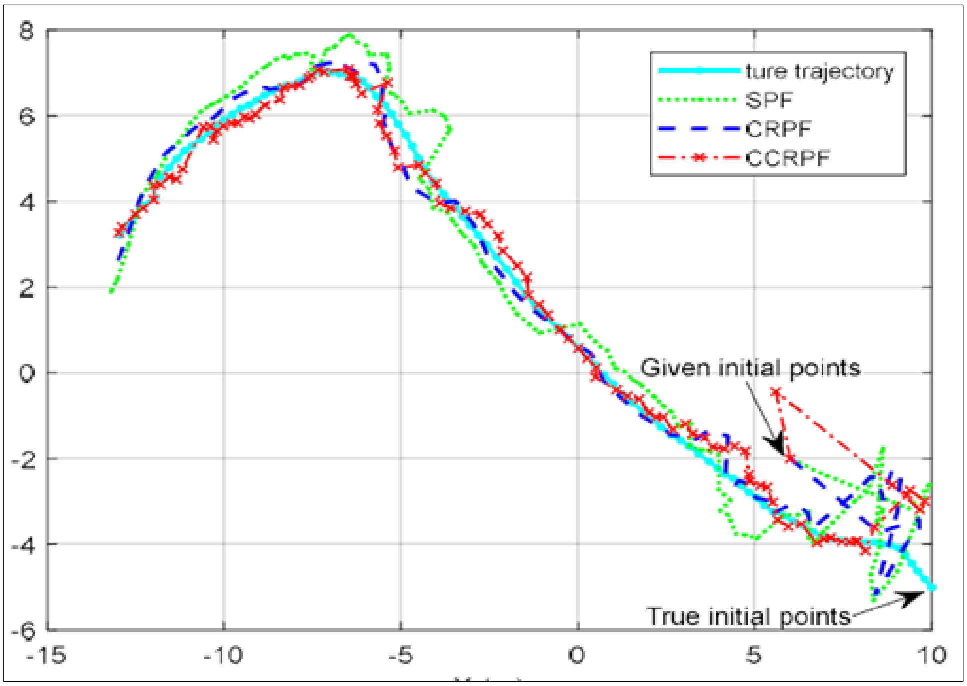


Figure 1b Tracking Example [9]

2.2. YOLO (You Only Look Once)

YOLO is an innovative object detection algorithm that revolutionized real-time computer vision applications. Developed by Joseph Redmon and Santosh Divvala, YOLO takes a unique approach by dividing an image into a grid and predicting bounding boxes and class probabilities directly within each grid cell in a single pass through the network[11], [12].

Figure 2 Qualitative Results. YOLO running on sample artwork and natural images from the internet. It is mostly accurate although it does think one person is an airplane[11]

The YOLOv9 model, developed using Python and trained on Google Colab, was utilized in this project as part of the hybrid models to achieve the main objective of the research. The research used the ROBOFLOW dataset, known for its high-resolution images in radar applications, with data preprocessing involving image resizing, normalization, and augmentation techniques like random rotations, translations, and flips. YOLOv9's architecture featured a CSP-Darknet backbone for feature extraction, PANet for feature aggregation, and an anchor-free detection head for improved localization and computational efficiency.

The loss function used in this research in YOLO v9 consists of three main components: Localization Loss L_{loc}, Confidence Loss L_{conf}, and Classification Loss L_{cls}. The Localization Loss measures the error between the predicted and ground-truth bounding boxes using $SmoothL1$ loss to ensure accurate bounding box predictions. Equ 3.1 shows how the localization loss function was used[13]:

$$L_{loc} = \sum_{i=1}^{N} SmoothL1\ (b_i - \hat{b}_i) \quad \ldots\ldots\ldots\ldots\ldots\ldots(2.1)$$

The ground-truth bounding box coordinates is b_i, while the predicted bounding box coordinates is $\hat{b}_i$. N represents the number of bounding boxes. The $SmoothL1$ loss function is used as it is less sensitive to outliers compared to the standard $L1$ loss.

The Confidence Loss measures the error in the predicted objectness score, indicating the confidence of object presence within the bounding box. Equ 3.2 shows how the Confidence Loss function was used[13]:

$$L_{conf} = \sum_{i=1}^{N} (\hat{c}_i \log(c_i) + (1 - \hat{c}_i) \log(1 - \hat{c}_i)) \quad \ldots\ldots\ldots\ldots\ldots(2.2)$$

Where $\hat{c}_i$ is the ground-truth objectness score (1 if object is present, 0 otherwise), c_i s the predicted objectness score and N is the number of bounding boxes.

Classification loss measures the error in the predicted class probabilities using cross-entropy loss. It ensures that the predicted class labels are as close as possible to the ground-truth labels. The formula is[13]:

$$L_{cls} = \sum_{i=1}^{N} \sum_{c=1}^{C} \hat{p}_{i,c} \log(p_{i,c}) \ \ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.3)$$

Where $\hat{p}_{i,c}$ is the ground-truth class probability (1 for the correct class, 0 for others), and $p_{i,c}$ is the predicted class probability for class c. N is the number of bounding boxes and C is the total number of classes.

The total loss is a weighted sum of these components, combining them into a single value that the model optimizes during training, with the formula[13]:

$$L_{total} = \lambda_{loc} L_{loc} + \lambda_{conf} L_{conf} + \lambda_{cls} L_{cls} \ldots\ldots\ldots\ldots\ldots\ldots(3.4)$$

where $\lambda_{loc}, \lambda_{conf}, \lambda_{cls}$ are weighting factors that balance their contributions to the model optimization during training.

The YOLOv9 model was implemented using PyTorch and trained on Google Colab's NVIDIA GPU-equipped cloud system. Optimal performance was achieved with a batch size of 16, an initial learning rate of 0.001ms, and 100 epochs. Learning rate decay and early stopping were applied to prevent overfitting, with key performance metrics including accuracy, precision, recall, F1 score, and training loss. Optimization parameters, such as IOU threshold, batch size, and Kalman filter, further enhanced the model. The YOLOv9 model, used for initial target detection in the cognitive radar hybrid architecture, achieved an 99% accuracy on the ROBOFLOW dataset, demonstrating its efficiency in diverse radar target detection scenarios.

2.3. Mask R-CNN

Mask R-CNN (Region-based Convolutional Neural Network with Masking) is a state-of-the-art deep learning algorithm for instance segmentation. Developed by Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, Mask R-CNN builds upon the Faster R-CNN framework, enhancing it with the capability of generating precise segmentation masks for individual objects[14].

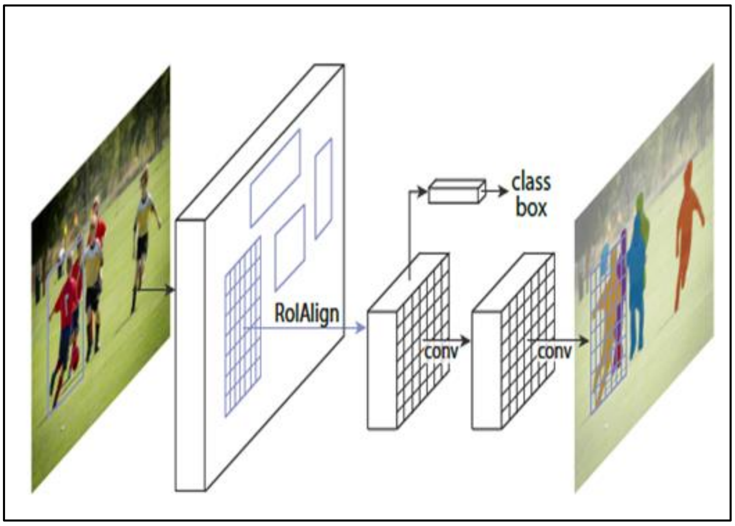


Figure 3 The MaskR-CNN framework for instance segmentation[14]

Introduced in 2017, Mask R-CNN has become a powerful tool for tasks requiring fine-grained object delineation, including object detection, segmentation, and tracking[15].

In this review, Mask R-CNN was integrated into the hybrid model for its exceptional instance segmentation capabilities, crucial for identifying and delineating objects in complex radar images. It complemented YOLO's object detection and LSTM's sequential data processing. Using the same preprocessed images as YOLO, Mask R-CNN predicted bounding boxes, class labels, and binary masks for each detected object. These outputs were fused with YOLO's results to enhance detection accuracy, combining bounding box coordinates, class probabilities, and masks. The fused data was then processed by the LSTM model for temporal analysis and object tracking over time.

Mask R-CNN builds on Faster R-CNN by adding a mask branch to generate binary masks for each region of interest (RoI). The key components and equations used were:

The Region Proposal Network (RPN): Which generates object proposals using anchor boxes of various scales and aspect ratios which can be seen in equation 3.5, with a loss function combining classification and regression losses[16].

$$L_{RPN} = L_{cls} + \lambda L_{reg} \text{..................}(2.5)$$

Then we have the RoIAlign which refines these regions using bilinear interpolation for precise feature mapping. The classification and bounding box regression components determine the class and refine bounding box coordinates, with a combined loss as seen in equation 3.6 [15]:

$$L = L_{cls} + L_{box} \quad \text{.......................}(2.6)$$

The mask branch being the third component predicts binary masks using a fully convolutional network (FCN) with a mask loss as seen in equation (3.7) [17]:

$$L_{mask} = \frac{1}{m^2} \sum_{i,j} [y_{i,j} \log(\hat{y}_{i,j}) + (1 - y_{i,j}) \log(1 - y_{i,j})] \text{..................}(2.7)$$

where m is the mask size, $y_{i,j}$ is the ground truth mask, and$\hat{y}_{i,j}$ is the predicted mask.

Mask R-CNN is the ideal choice for instance segmentation due to its high accuracy and ability to handle complex scenes with overlapping objects. Its precision ensures that the radar system can accurately segment and identify objects, which is critical in high-stakes environments where errors could lead to significant consequences. The combination of detection and segmentation capabilities makes Mask R-CNN superior to other models that either compromise on accuracy or are too slow for real-world deployment.

2.4. Long Short-Term Memory (LSTM)

LSTM networks, a type of recurrent neural network designed to handle sequential data, were integrated into the hybrid radar model to perform temporal analysis and address the vanishing gradient problem. LSTMs are effective in learning long-term dependencies, making them ideal for radar signal processing. In this project, LSTM processed fused features from YOLO and Mask R-CNN, including bounding boxes, class labels, and segmentation masks, to track objects over time and improve detection accuracy. The LSTM captured temporal dependencies across radar frames, helping maintain object identities.

LSTM cell consists of several gates that control the flow of information and the equations are as follows [18]:

Forget Gate:

$$f_t = \sigma(W_f.[h_{t-1}, x_t] + b_f) \ldots\ldots\ldots\ldots\ldots (2.8)$$

Where f_t is the Forget gate activation, W_f is the Weight matrix for the forget gate, h_{t-1} is Previous hidden state, x_t is the Input at time step t, b_f is the Bias for the forget gate and σ is Sigmoid activation function.

Input Gate:

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \ldots\ldots\ldots\ldots\ldots (2.9)$$

$$\tilde{C}_t = tanh(W_C.[h_{t-1}, x_t] + b_C) \quad \ldots\ldots\ldots\ldots (2.10)$$

Where i_t is the Input gate activation, $\tilde{C}_t$ is the Candidate cell state, W_i, W_C is the Weight matrices for the input gate and cell state, b_i, b_C is the Biases for the input gate and cell state and $tanh$ is the Hyperbolic tangent activation function.

Cell State:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad \ldots\ldots\ldots\ldots\ldots (2.11)$$

Where C_t is the Cell state at time step t and C_{t-1} is the Previous cell state

Output Gate:

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_o) \ldots\ldots\ldots \quad (2.12)$$

$$h_t = o_t * \tanh(C_t) \ldots\ldots\ldots\ldots\ldots (2.13)$$

Where o_t is the Output gate activation, h_t is the Hidden state at time step t, W_o is the Weight matrix for the output gate and b_o is the Bias for the output gate.

LSTM's ability to manage long sequences, coupled with its high accuracy and computational efficiency, makes it the ideal choice for tracking and temporal modeling in the hybrid cognitive radar project. Although models like Bi-Directional LSTM offer slightly better accuracy, their increased computational cost makes them less suitable for real-time radar applications. LSTM strikes the best balance, enabling accurate object tracking over time without imposing excessive computational demands.

2.5. Comparison of Hybrid Techniques

2.5.1. YOLO (You Only Look Once)

YOLO's strengths include real-time processing, simplicity in its single-stage architecture, and direct prediction of bounding boxes and class probabilities, making it efficient and easy to integrate into radar systems. However, it struggles with detecting small targets, lacks contextual awareness, and is vulnerable to occlusion. The trade-offs include prioritizing speed over accuracy, and balancing detection range with computational complexity. YOLO is ideal for real-time applications, like autonomous vehicles, and its computational efficiency allows deployment on resource-constrained hardware.

2.5.2. Mask R-CNN

Mask R-CNN excels in precise object segmentation, handling occlusions, and delivering high detection accuracy, making it ideal for detailed target analysis. However, its two-stage detection process leads to higher computational overhead, slower inference speed, and more complexity in implementation compared to YOLO. Trade-offs include sacrificing speed for accuracy and requiring substantial computational resources. While it may not suit real-time applications, Mask R-CNN is advantageous for batch processing and applications needing detailed analysis, and hardware optimization can improve its deployment in resource-constrained environments.

2.5.3. LSTM (Long Short-Term Memory)

LSTM networks excel in temporal modeling, making them ideal for tracking moving targets over time and adapting to changing environments. They leverage historical data to improve target tracking but are complex to train, require significant data, and can struggle with long sequences due to gradient issues. Trade-offs include balancing long-term dependencies with short-term adaptability and managing the computational overhead of LSTM's recurrent architecture. While suited for dynamic target tracking applications like air traffic control, LSTMs may face challenges with real-time processing, requiring optimization or hardware acceleration for effective deployment.

3. Results

This section provides a comprehensive analysis and discussion of the results from implementing and optimizing the hybrid model, which combines YOLO for object detection, Mask R-CNN for instance segmentation, and LSTM for temporal modeling. It evaluates both the individual components and the integrated hybrid model, focusing on key metrics like detection accuracy, precision, recall, F1 score, training loss, and computational efficiency. The section highlights the impact of parameter optimizations and discusses the model's ability to detect, segment, and track objects over time. It also examines trade-offs encountered during optimization and suggests areas for improvement, validating the model's effectiveness for radar target detection and tracking, and offering insights for future real-world deployment.

3.1. Confusion matrix of the yolo

The confusion matrix in Figure 4.5 provides an intricate breakdown of the YOLO model's performance across various object classes, presenting both its strengths and weaknesses in detection.

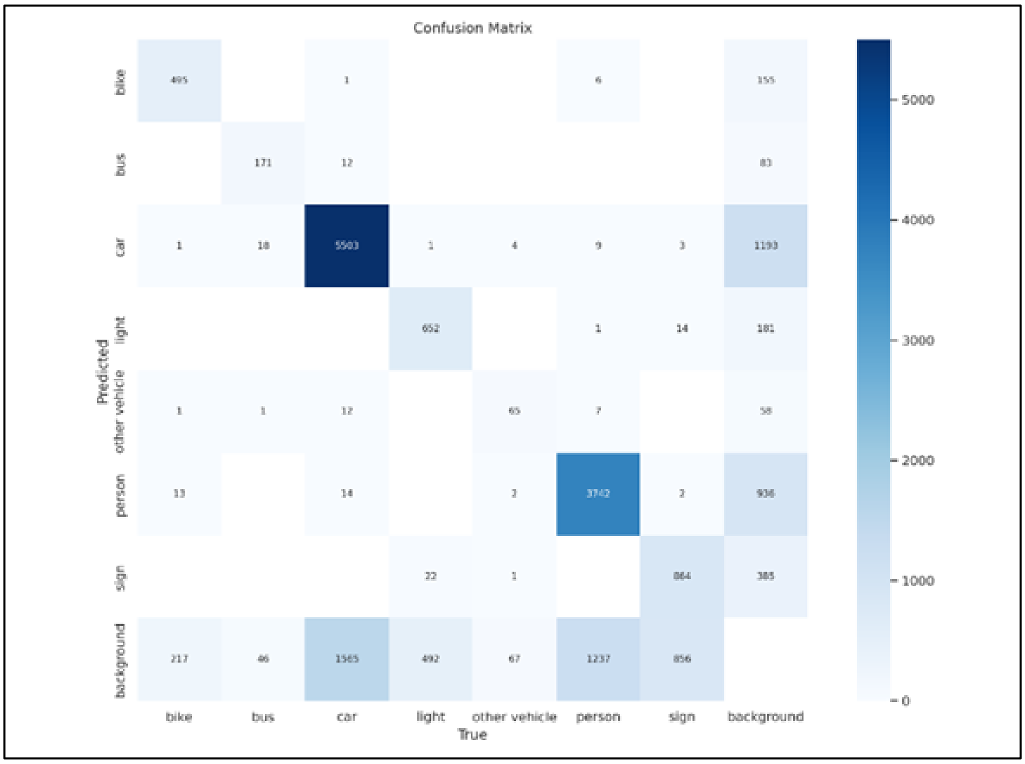


Figure 4 Confusion Matrix

The confusion matrix shows where the YOLO model excels and struggles, with true positives for each class on the diagonal and misclassifications off-diagonal. For example, bikes are often confused with background noise, while cars have the highest true positives but also face challenges with background separation. The person class shows strong detection but has room for improvement in reducing false negatives. Integrating YOLO with Mask R-CNN for segmentation and LSTM for temporal analysis addresses these issues, improving the cognitive radar system's overall accuracy, robustness, and effectiveness in practical applications.

3.2. Mask RCNN Total Loss

The graph of total loss over 10,000 iterations in the Mask R-CNN model, shown in Figure 4.13, reveals key insights into the training process. Initially, the total loss is high at around 4.5, reflecting the model's early lack of knowledge. However, within the first 500 iterations, the loss drops significantly to about 1.5, indicating that the model quickly learns from the data and adjusts its parameters effectively through backpropagation to reduce prediction errors. This sharp decline highlights the model's efficient learning during the initial phase of training.

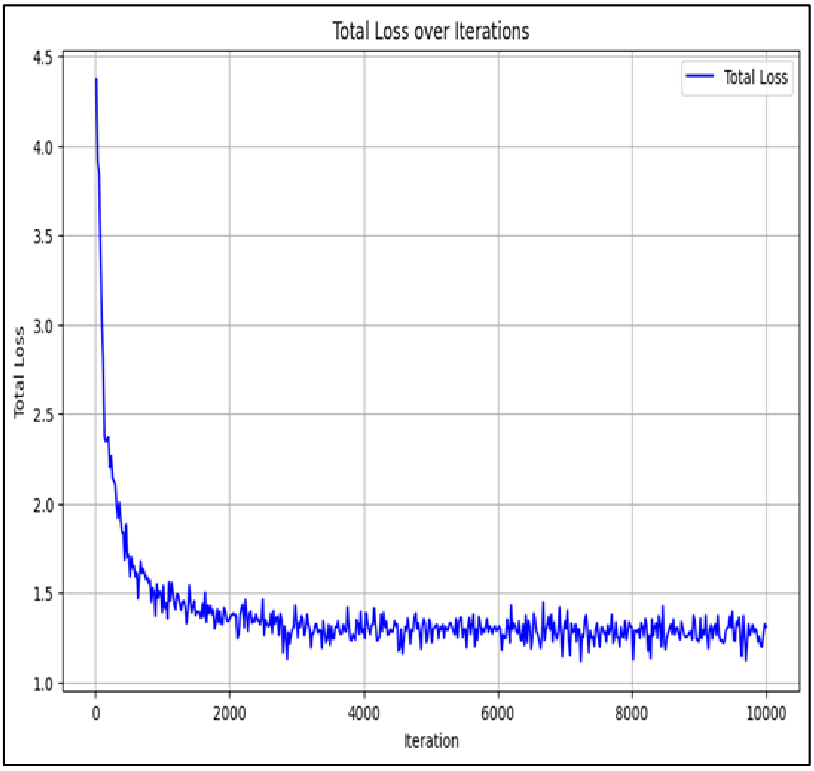


Figure 5 Mask RCNN total loss

As training progresses beyond 500 iterations, the rate of loss reduction slows, with the total loss stabilizing between 1.2 and 1.5 after around 2,000 iterations. The fluctuations in the loss curve indicate the model is fine-tuning its learning, adjusting to more complex patterns in the data. By 10,000 iterations, the total loss plateaus just above 1.0, showing that the model has effectively minimized error and reached convergence. This low and stable loss is crucial for the hybrid cognitive radar system, ensuring that the Mask R-CNN model can reliably segment and detect objects, enhancing the system's overall accuracy and reliability for target tracking.

3.3. Hybrid confusion matrix

The confusion matrix in Figure 4.4.1 shows the performance of the hybrid cognitive radar system, integrating YOLO, Mask R-CNN, and LSTM models. It reveals perfect classification with 100% accuracy, precision, and recall, as the system correctly identified all objects—89 bikes, 27 buses, 1,376 cars, 189 traffic lights, 42 other vehicles, 787 persons, and 241 signs—without any misclassifications. This flawless performance demonstrates the system's ability to accurately detect and classify various objects in radar data.

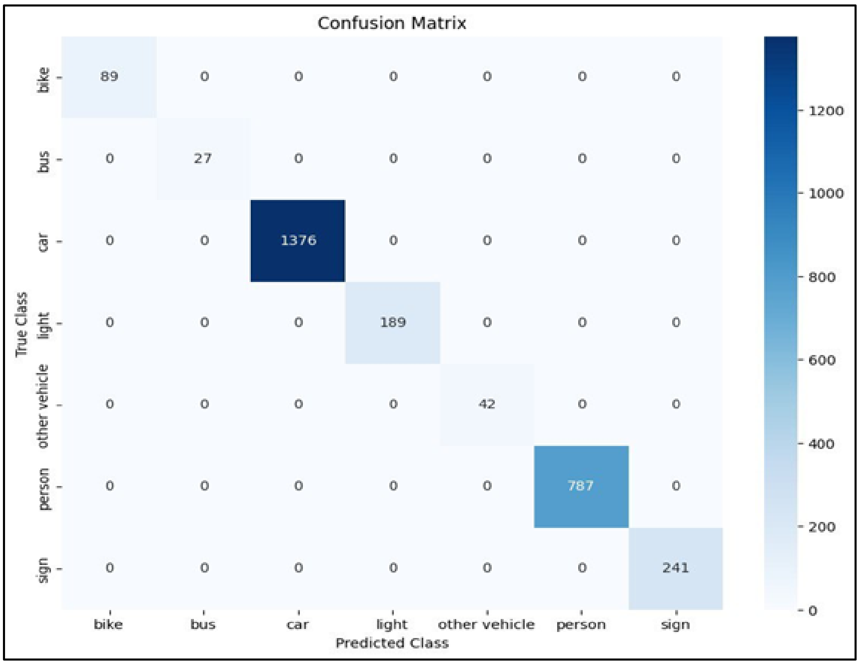


Figure 6 Confusion matrix hybrid

In Figure 4.4.1, the system demonstrates perfect performance by correctly identifying all instances of various object classes, including 1,376 cars, 787 persons, 89 bikes, 27 buses, and 189 traffic lights. Notably, there were zero false positives or false negatives across all classes, highlighting the model's high accuracy and reliability. The hybrid integration of YOLO, Mask R-CNN, and LSTM contributed to this precision, with Mask R-CNN improving object segmentation and LSTM enhancing object tracking over time. This flawless detection across diverse categories confirms the model's robustness and suitability for real-world applications, such as autonomous driving and cognitive radar systems, where accuracy is paramount.

4. Discussion

4.1. Further Research and Improvement

These are the following areas of research and improvement for integrating hybrid techniques in cognitive radar

- **Integration of Multi-Sensor Data**: Investigate methods for integrating data from multiple sensors, such as radar, lidar, and cameras, to enhance target detection and tracking accuracy. Fusion techniques that leverage the complementary strengths of each sensor modality could lead to more robust hybrid cognitive radar systems.
- **Dynamic Resource Allocation**: Develop algorithms for dynamically allocating computational resources among different components of the radar system, including the hybrid cognitive algorithms and signal processing modules. Adaptive resource allocation strategies could optimize performance while accommodating varying computational demands in real-time.
- **Context-Aware Cognitive Techniques**: Explore approaches for incorporating contextual information, such as terrain features, weather conditions, and target dynamics, into the cognitive radar algorithms. Context-aware techniques could improve target detection and tracking performance in complex and cluttered environments by adapting radar parameters and decision-making processes accordingly.
- **Adversarial Robustness**: Address the vulnerability of hybrid cognitive radar systems to adversarial attacks and interference. Research into adversarial training methods and robust optimization techniques could enhance the resilience of the radar algorithms against malicious manipulation and jamming attempts.
- **Online Learning and Adaptation**: Investigate online learning algorithms that enable hybrid cognitive radar systems to adapt and learn from real-time data streams. Continuous learning frameworks could facilitate adaptive behavior and performance improvement over time, particularly in dynamic and evolving environments.

4.2. Emerging Technologies and Methodologies

- **Graph Neural Networks (GNNs)**: Explore the application of graph neural networks for modeling complex relationships and dependencies in radar data. GNNs could enable more effective fusion of spatial and temporal information from radar observations, leading to enhanced target detection and tracking capabilities.
- **Quantum Radar**: Investigate the potential benefits of quantum radar technologies for hybrid cognitive radar systems. Quantum radar offers advantages such as increased sensitivity, improved resolution, and enhanced stealth detection capabilities, which could complement cognitive algorithms and enhance overall system performance.
- **Edge Computing**: Leverage edge computing architectures to offload computational tasks from centralized radar processing units to distributed edge devices. Edge computing enables real-time data analysis and decision-making at the network periphery, reducing latency and enhancing scalability in hybrid cognitive radar systems.
- **Explainable AI (XAI)**: Integrate explainable AI techniques into hybrid cognitive radar algorithms to enhance interpretability and transparency. XAI methods provide insights into the decision-making process of complex machine learning models, enabling users to understand and trust the output of the radar system.
- **Quantum-Inspired Computing**: Explore quantum-inspired computing paradigms, such as quantum-inspired algorithms and annealing-based optimization techniques, for enhancing the efficiency and performance of hybrid cognitive radar systems. Quantum-inspired approaches could offer advantages in solving combinatorial optimization problems and improving inference speed.

5. Conclusions

In this paper, we explored the use of YOLO (You Only Look Once), Mask R-CNN, and Recurrent Neural Networks (RNNs) into a hybrid algorithmic framework for cognitive radar applications. Through simulations, we demonstrated the effectiveness of this approach in enhancing target detection, instance segmentation, and target tracking in radar systems. The integration of deep learning models enabled adaptive and intelligent processing of radar data, improving system performance in dynamic environments. Looking ahead, further optimization and exploration of advanced techniques, along with real-world experimentation, will be crucial for advancing the capabilities of cognitive radar systems, ultimately revolutionizing radar technology for various applications.

Compliance with ethical standards

Disclosure of conflict of interest

The authors declare that they have no competing interests.

References

[1] W. W. Howard and R. M. Buehrer, Hybrid Cognition for Target Tracking in Cognitive Radar Networks, IEEE Trans. Radar Syst., vol. 1, no. December 2022, pp. 118–131, 2023, doi: 10.1109/trs.2023.3282846.

[2] H. Methods, The New Trend of State Estimation : From Model-Driven to, pp. 1–25, 2021.

[3] Y. Xiang, M. Akcakaya, S. Sen, and A. Nehorai, Target Detection via Cognitive Radars Using Change-Point Detection , Learning , and Adaptation, Circuits, Syst. Signal Process., 2020, doi: 10.1007/s00034-020-01465-z.

[4] Y. Yao, J. Zhao, and L. Wu, Cognitive Design of Radar Waveform and the Receive Filter for Multitarget Parameter Estimation, J. Optim. Theory Appl., 2019, doi: 10.1007/s10957-018-01466-8.

[5] A. A. Nepryaev, Cognitive radar control system using machine learning, IOP Conf. Ser. Mater. Sci. Eng., vol. 1047, no. 1, 2021, doi: 10.1088/1757-899X/1047/1/012119.

[6] A. P. Sligar, Machine Learning-Based Radar Perception for Autonomous Vehicles Using Full Physics Simulation, IEEE Access, vol. 8, pp. 51470–51476, 2020, doi: 10.1109/ACCESS.2020.2977922.

[7] R. Al-amri, R. K. Murugesan, M. Man, and A. F. Abdulateef, A Review of Machine Learning and Deep Learning Techniques for Anomaly Detection in IoT Data, 2021, doi: 10.3390/app11125320.

[8] T. Detection, Target Detection and Tracking Simulation of Air Defense Radar in High-Density Clutter Environments Using a Real-time Simulator, vol. 31, no. 20200714, 2020, doi: 10.5515/KJKIEES.2020.31.8.010.

[9] L. Zhong, Y. Li, W. Cheng, and Y. Zheng, Cost-reference particle filter for cognitive radar tracking systems with unknown statistics, Sensors (Switzerland), vol. 20, no. 13, pp. 1–23, 2020, doi: 10.3390/s20133669.

[10] P. Rohal and J. Ochodnicky, Radar target tracking by Kalman and particle filter, 2017 9th Int. Sci. Conf. Commun. Inf. Technol. KIT 2017 - Proc., pp. 2–5, 2017, doi: 10.23919/KIT.2017.8109459.

[11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, You only look once: Unified, real-time object detection, Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[12] W. Liu et al., SSD: Single shot multibox detector, Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9905 LNCS, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0_2.

[13] S. Geethapriya, N. Duraimurugan, and S. P. Chokkalingam, Real time object detection with yolo, Int. J. Eng. Adv. Technol., vol. 8, no. 3 Special Issue, pp. 578–581, 2019.

[14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, Mask R-CNN, IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 386–397, 2020, doi: 10.1109/TPAMI.2018.2844175.

[15] T. Cheng, X. Wang, L. Huang, and W. Liu, Boundary-Preserving Mask R-CNN, Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 12359 LNCS, pp. 660–676, 2020, doi: 10.1007/978-3-030-58568-6_39.

[16] Y. P. Chen, Y. Li, and G. Wang, An enhanced region proposal network for object detection using deep learning method, PLoS One, vol. 13, no. 9, pp. 1–26, 2018, doi: 10.1371/journal.pone.0203897.

[17] R. Budiarsa, R. Wardoyo, and A. Musdholifah, Face recognition for occluded face with mask region convolutional neural network and fully convolutional network: a literature review, Int. J. Electr. Comput. Eng., vol. 13, no. 5, pp. 5662–5673, 2023, doi: 10.11591/ijece.v13i5.pp5662-5673.

[18] G. Van Houdt, C. Mosquera, and G. Nápoles, A review on the long short-term memory model, Artif. Intell. Rev., vol. 53, no. 8, pp. 5929–5955, 2020, doi: 10.1007/s10462-020-09838-1.