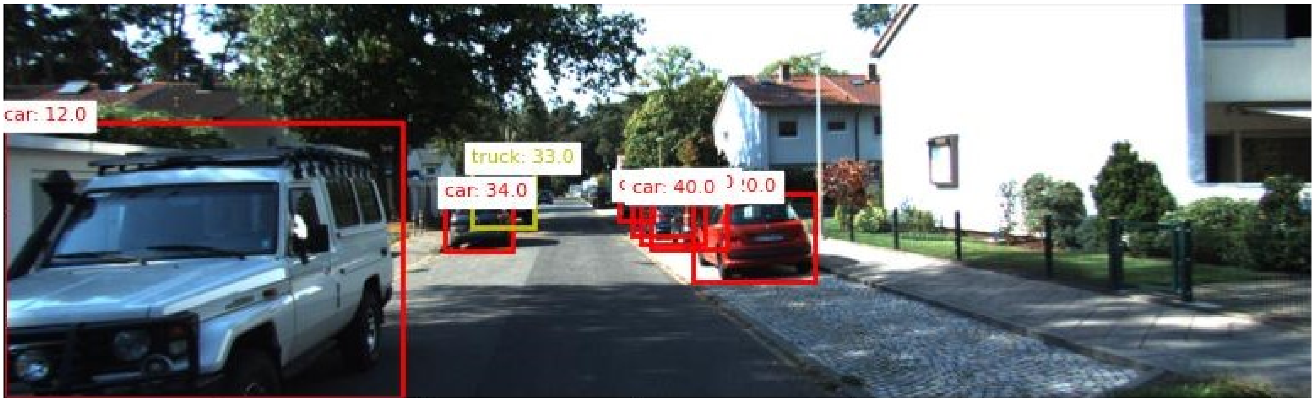
Enhancing YOLOv5 Object Detection and Tracking Integrated with DeepSORT Algorithm for Autonomous Vehicles

Asma Bin Thabit*
Michigan State University
binthabi@msu.edu

Hamed Khatounabadi*
Michigan State University
khatouna@msu.edu



Abstract

In this paper, we introduce an advanced approach for object detection and tracking, applied to the sequentially structured KITTI dataset, utilizing the capabilities of YOLOv5. Our approach is geared towards improving the accuracy of detecting and tracking multiple objects in video sequences, specifically addressing challenges such as detection failures and reduction of false positives. Conventional methods typically face difficulties in maintaining consistent object recognition throughout video frames, resulting in less than ideal detection results. To address these challenges, we incorporate DeepSORT, which significantly enhances tracking and precise identification of each detected object. This amalgamation not only improves the accuracy of detection but also effectively addresses misidentification and reduces false positives in following frames. Our experimental outcomes demonstrate marked enhancements over existing benchmarks in both object detection and tracking. This progress is particularly relevant to autonomous driving and traffic surveillance, contributing significantly to enhancing the safety and reliability of such systems.

1. Introduction

In the advancing field of autonomous driving and traffic surveillance, the precision of object detection and tracking systems is crucial. High accuracy in these areas is essential for ensuring the safety and reliability of autonomous vehicles and traffic monitoring systems. While there have been significant advancements in object detection technologies, such as YOLOv5 [4], which offers impressive speed and accuracy, the challenge of consistently tracking objects across video frames in dynamic environments remains.

This paper introduces a novel approach that integrates YOLOv5 [4] with DeepSORT [7], an advanced object-tracking algorithm. YOLOv5 is renowned for its object detection capabilities, while DeepSORT provides robust tracking features. By combining these two technologies, we aim to address the limitations faced by object detection models when used independently for tracking purposes in complex scenarios, such as those presented in the KITTI dataset [3].

Our approach focuses on leveraging the strengths of YOLOv5 in detecting objects with high accuracy [2] and DeepSORT's ability to maintain consistent tracking of these objects across successive frames. The integration is designed to enhance the overall performance of object detection and tracking systems, particularly in terms of consistency and reliability.

Through this paper, we explore the potential of this integrated approach in the context of autonomous driving and traffic monitoring, highlighting the importance of accurate and reliable object tracking in these domains [9]. We believe that our method can contribute significantly to the advancement of technologies in these fields, paving the way for safer and more efficient autonomous systems.

2. Related Work

In the evolving landscape of object detection and tracking, the integration of advanced algorithms and robust datasets has led to significant strides. This section reviews pivotal contributions that have shaped the field, particularly focusing on the innovations in technologies like YOLO and DeepSORT, and their broad impact from autonomous navigation to mobile robotics.

Pereira et al. [7] explored new data association metrics in mobile robotics, underscoring the effectiveness of SORT and Deep-SORT in dynamic environments. Their findings emphasized the importance of improved tracking accuracy and data association, especially in real-time robotic applications. Complementing this, Pujara and Bhamare [9] demonstrated the integration of DeepSORT with YOLO and TensorFlow, marking a significant advancement in achieving real-time tracking with enhanced accuracy, vital for applications like surveillance systems.

In the realm of autonomous vehicles, Perera et al. [8] utilized an improved DeepSORT algorithm in tandem with YOLOv4 for vehicle tracking. This study is particularly valuable for its implications in autonomous vehicle systems where precise and timely vehicle detection is paramount. Adding to the versatility of these technologies, Durve et al. [2] provided a comparative analysis of YOLOv5 and YOLOv7 models integrated with DeepSORT, focusing on droplet tracking applications. Their work offers critical insights into the performance nuances and application-specific tuning of these models.

In another significant advancement, Rakotoniaina et al. [10] introduced LIV-DeepSORT, optimizing DeepSORT for autonomous vehicles using camera and LiDAR data fusion. This research provides a robust framework for enhancing tracking accuracy in complex environmental conditions. Azevedo and Santos [1] further expanded the application scope by employing YOLO-based object detection and tracking on edge devices for autonomous vehicles, offering practical insights into deploying these complex algorithms in resource-constrained settings.

Additionally, Wojke et al. [11] proposed an approach that combines online and real-time tracking with a deep association metric. This methodology addresses the challenges of maintaining object identities over time and under varying environmental conditions, making a substantial contribution to the field of real-time tracking.

Collectively, these studies not only contribute to the understanding and advancement of object detection and tracking technologies but also offer valuable insights into algorithmic improvements, practical applications, and the integration of various data sources and technologies, all of which are essential for the ongoing development in this field. These studies lay a robust foundation for our project. They provide critical insights into the strengths and limitations of current object detection and tracking technologies, particularly in the context of dynamic and complex environments like those encountered in autonomous driving. The advancements in YOLOv5 and DeepSORT, as well as the valuable benchmarks provided by the KITTI dataset, serve as key reference points for our approach. By integrating YOLOv5 with DeepSORT, our project aims to address the specific challenges of object tracking in autonomous vehicles, leveraging the high detection accuracy of YOLOv5 and the enhanced tracking capability of DeepSORT. This integration, tested against the comprehensive scenarios presented in the KITTI datasets, is expected to yield significant improvements in the reliability and efficiency of autonomous vehicle systems, pushing the boundaries of current technologies in this rapidly evolving field.

3. Methodology

Our methodology involves integrating YOLOv5 [4] with DeepSORT for enhanced object detection and tracking on the KITTI dataset [3]. The process is structured to leverage the strengths of both YOLOv5's detection capabilities and DeepSORT's tracking efficiency. Below is a detailed description of the methodology:

3.1. Dataset and Preprocessing

1. **Datasets:**

   The KITTI dataset, renowned for its diverse and realistic driving scenarios, serves as the primary dataset for our study. For training purposes, we utilize 7,481 samples. For testing, we have carefully selected a specific subset, namely "Subset 0007," from the multi-object tracking section of the KITTI dataset available on the KITTI website [5]. This subset comprises 800 samples and was chosen due to its diverse range of annotations and sequence order, which are critical for effective object detection and tracking tasks.

2. **Preprocessing**:

   The selected datasets undergo thorough preprocessing, which is vital for the effective application of YOLOv5. This preprocessing includes normalizing the images to ensure consistency in lighting and scale. Additionally, we generate labels that are compatible with the YOLOv5 format, facilitating accurate object detection. The preprocessing stage also involves aligning and calibrating the sensor data, crucial for ensuring precision in both detection and tracking tasks.

3.2. Object Detection with YOLOv5

YOLOv5, known for its speed and accuracy in object detection, serves as the backbone of our detection system [2]. The model is trained on the KITTI dataset focusing on five main classes: Car (class 0), Van (class 1), Misc (class 2), Truck (class 3), and Tram (class 4). This classification covers a comprehensive range of vehicles from standard cars to larger and uniquely shaped vehicles like trucks and trams. The model is fine-tuned to recognize the distinct features of each category, ensuring accurate detection crucial for the safety and efficiency of autonomous driving systems.

1. **Training**:

   The YOLOv5 model is trained on the annotated images from the KITTI dataset. This involves feeding the model with labeled images, allowing it to learn the features and characteristics of different objects in the five categories mentioned above.

2. **Optimization**:

   The model begins training with YOLOv5's default hyperparameters, establishing a solid baseline. Subsequently, these hyperparameters are fine-tuned to better address the specific challenges of the KITTI dataset, like detecting small or partially occluded objects.

3.3. Object Tracking with DeepSORT

DeepSORT [7] is integrated with YOLOv5 for object tracking. This algorithm enhances the tracking process by using deep learning features alongside traditional tracking methods like Kalman filters.

1. **Integration and Configuration of DeepSORT**:

   After YOLOv5 detects objects, DeepSORT takes over to track these objects across video frames. It begins by assigning unique IDs to each detected object. This is crucial for maintaining individual tracking consistency, especially in sequences where multiple objects are present. DeepSORT employs a combination of motion information and appearance features to follow each object, ensuring that the same ID is retained across successive frames, even in challenging scenarios like rapid movements or crowded scenes.

   It is configured with parameters like `max_age`, `n_init`, and `max_cosine_distance`, which manage track longevity, initialization, and detection association, respectively.

   The system utilizes `clip_ViT-B/32` for feature extraction, optimizing for performance and adapting for RGB images. The `update_tracks` function, essential for the tracking process, processes detections and current frame images, ensuring consistent tracking across frames.

2. **Feature Extraction and Advanced Tracking**:

   DeepSORT utilizes a CNN in our case `clip_ViT-B/32` to extract rich appearance features from each detected object. This feature extraction is pivotal for differentiating between objects that may look similar but are distinct entities. The algorithm also utilizes Kalman filtering for predicting the objects' locations in subsequent frames, which is particularly effective in handling temporary occlusions or when objects move out of the frame temporarily. By combining motion and appearance data, DeepSORT can accurately track objects even when they are closely interacting or overlapping with others.

   DeepSORT employs `clip_ViT-B/32` to extract features from detected objects. Then combined with Kalman filtering, it predicts object locations in subsequent frames.

   The algorithm applies sophisticated data association techniques, like the Hungarian algorithm and a deep association metric [6], to maintain consistent tracking. This approach is necessary for correctly identifying objects across frames, especially in dynamic environments.

3. **IOU and Track Association**:

   The code calculates the IOU (Intersection Over Union) between YOLOv5 detections and DeepSORT tracks to associate detections with existing tracks. An IOU threshold (**0.6** in our case) is used to determine if a detection corresponds to an existing track. If the IOU is above this threshold, the detection is considered part of the track. This step ensures that the object detections by YOLOv5 are accurately associated with the correct tracks maintained by DeepSORT, even across several frames.

   The system scales bounding boxes to fit the native space and formats them for DeepSORT. Each track's reliability is assessed, and its ID and bounding box are extracted.

   An Intersection Over Union (IOU) calculation associates YOLOv5 detections with DeepSORT tracks. Detections are matched to tracks based on an IOU threshold, ensuring accurate track continuation across multiple frames.

4. **Handling Track Consistency Across Frames**:

   The implementation includes a mechanism to ensure track consistency across multiple frames. It compares the current frame's track IDs with those of previous frames to validate and correct any inconsistencies. If a tracking ID is found to be inconsistent over consecutive frames, it is adjusted for continuity, ensuring that the same object is correctly tracked throughout its presence in the video.

In the following section, we present a detailed summary of the DeepSORT algorithm's steps. Table 1 encapsulates the sequence of operations and parameters that are central to the algorithm, providing a clear overview of the process from initialization to the final output.

Algorithm 1 DeepSORT Algorithm Summary

```
Input: $X = \{X_1, \ldots, X_n\}$, matrix of $n$ detected objects
Parameters: $max\_age, n\_init, max\_cosine\_distance$
Initialize: $Tracks \leftarrow \emptyset$, $ite \leftarrow 0$
Perform detection using YOLOv5 to obtain initial detections
Initialize DeepSORT with the parameters
repeat
    Extract features for detected objects using clip_ViT-B/32
    Predict new object states with Kalman Filter
    Associate detections to existing tracks using Hungarian algorithm
    Update track states with associated detections
    Remove tracks that exceed $max\_age$
    $ite \leftarrow ite + 1$
until End of video sequence or no detections left to process
Output: List of tracks with their trajectories
```

3.4. Performance Evaluation

The performance of our integrated YOLOv5 and DeepSORT system is assessed using key metrics to ensure its effectiveness in detection and tracking such as accuracy, precision, recall, and mean Average Precision (mAP).

Figure 1 shows all the stages of our process, starting with dataset processing, followed by detection and tracking, and culminating in the final results with correct label identification.

4. Experiments

In this section, we detail the experimental setup and procedures used to evaluate the performance of the integrated YOLOv5 and DeepSORT algorithms. The objective of these experiments is to validate the effectiveness of the proposed system in accurately detecting and tracking objects within the KITTI dataset.

4.1. Implementation Details

The implementation of our system was performed using Python 3.9. The deep learning components were developed with the PyTorch framework.

1. **Object Detection**

   For object detection, we utilized the YOLOv5 model, trained on 7,481 samples at an image resolution of 640x640 pixels. The training was initially conducted over 10 epochs, followed by an additional 4 epochs, using a batch size of 16 and the model's default hyperparameters, establishing a baseline for our experiment. The training process, which lasted approximately 1 hour on our specified hardware, was adequate to encompass the diversity present in the KITTI dataset. Subsequently, the trained model was tested on 800 samples from the multi-object tracking subset of the KITTI dataset, which is characterized by its sequence order.

   Table 1 presents the set of hyperparameters used during the training phase. These parameters were chosen to balance the computational efficiency with the accuracy of the model.

Hyperparameter	**Value**
Batch Size	16
Epochs	10 + 4
Image Resolution	640x640 pixels
Learning Rate	$1e^{-2}$
Optimizer	SGD

Table 1. Training Hyperparameters for YOLOv5

   Similarly, Table 2 details the hyperparameters used during the testing phase. Adjustments in batch size, non-maximum suppression, and confidence thresholds were made to ensure that the model accurately detects and tracks objects in various testing scenarios. These parameters were crucial in evaluating the model's performance on the selected subset of the KITTI dataset.

Hyperparameter	**Value**
Batch Size	128
Image Resolution	640x640 pixels
Confidence Threshold	0.5
IOU Threshold	0.6

Table 2. Testing Hyperparameters for YOLOv5

2. **Object Tracking**

   In our object tracking implementation, we utilized the DeepSORT algorithm configured with specific parameters to optimize performance and accuracy.

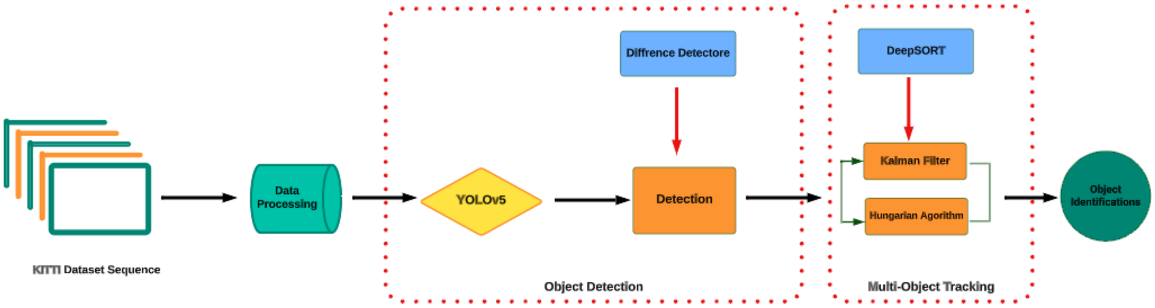


Figure 1. Pipeline of Multi-Object Detection and Tracking Using YOLOv5 and DeepSORT

Table 3 details this configuration. The maximum age of tracks was set to 10, allowing the algorithm to track objects over a reasonable number of frames, while the initialization threshold was set to 3 to ensure robust track initiation. A num-maximum suppression overlap of 1.0 was chosen to effectively manage overlapping detections, and a maximum cosine distance of 0.2 was used for accurate association of detections. The neural network budget was adaptive, providing flexibility based on the number of objects in the scene. The feature embedding was performed using the clip_ViT-B/32 architecture, operating without half-precision mode to balance computational efficiency and accuracy, with GPU acceleration enabled for enhanced processing speed.

Parameter	**Value**
Maximum Age of Tracks	10
Initialization Threshold	3
NMS Overlap	1.0
Maximum Cosine Distance	0.2
Neural Network Budget	Unspecified (Adaptive)
Feature Embedder Architecture	clip_ViT-B/32
Half-Precision Mode	False
GPU Acceleration	True

Table 3. DeepSORT Algorithm Configuration for Object Tracking

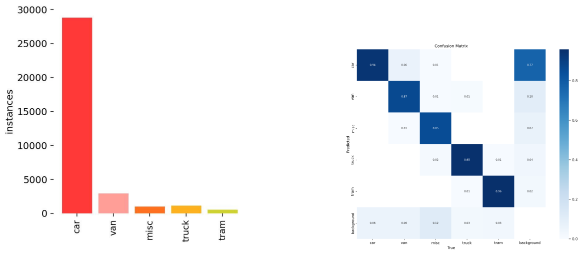
5. Results

This section presents the outcomes of our study, structured into three distinct stages: training, baseline testing, and testing post-implementation of DeepSORT. Each stage is critical in evaluating the efficacy of our integrated YOLOv5 and DeepSORT approach for object detection and tracking.

5.1. Training Stage

Initially, the **training stage** focuses on the model's learning process using the YOLOv5 architecture. Here, we discuss the training performance, highlighting key metrics that indicate the model's ability to learn and generalize from the training dataset.

In Figure 2, we present a comprehensive analysis of our training dataset and the subsequent model performance. Figure 2a details the distribution of instances per class, where a notable class imbalance is observed, with the 'car' class significantly outnumbering the others. Such a distribution can have a profound impact on the model's training, potentially leading to a bias towards the 'car' class. Correspondingly, Figure 2b illustrates the model's classification performance across these classes. Despite the imbalance, the model exhibits a commendable true positive rate for 'car' instances, while also maintaining high accuracy for the 'tram' class. These results collectively offer insights into the model's detection capabilities and highlight the need for further tuning to address the challenges posed by less represented classes.



(a) Distribution of instances per class in the training dataset.

(b) Confusion matrix showcasing the classification performance during the training phase.

Figure 2. Analysis of the training dataset and model performance.

The collection of figures from 3 to 4d provides a comprehensive analysis of the model's performance during the training phase. The training loss and metric progressions shown in Figure 3 reflect the model's learning trajectory, with box, object, and classification losses all improving over epochs. The F1-Confidence Curve in Figure 4a illustrates the balance between precision and recall across different confidence thresholds, indicating the model's accuracy in classifying various objects. The Precision-Confidence Curve (Figure 4b) and Recall-Confidence Curve (Figure 4d) further detail the model's ability to confidently predict true positives, while the Precision-Recall Curve (Figure 4c) emphasizes the model's overall effectiveness across all classes. These visualizations collectively underscore the robustness of our approach and aid in fine-tuning the model for optimal performance.

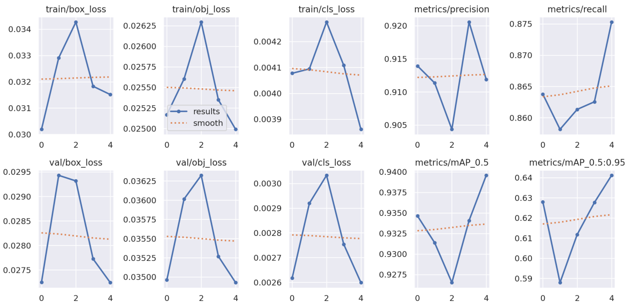
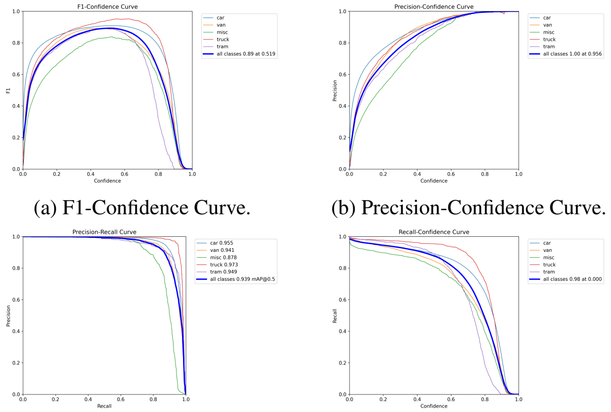


Figure 3. Training Loss and Metrics



(a) F1-Confidence Curve.

(b) Precision-Confidence Curve.

(c) Precision-Recall Curve.

(d) Recall-Confidence Curve.

Figure 4. Training YOLOv5 on KITTI dataset performance metrics

5.2. Baseline Testing Stage

Following this, we transition to the **baseline testing stage**, where the trained model is evaluated in a controlled environment to establish a benchmark for its detection capabilities. This stage is crucial for understanding the model's performance before integrating the DeepSORT algorithm.

The confusion matrix provided in Figure 5 delineates the object detection model's predictive performance in the testing phase prior to the integration of the DeepSORT algorithm. The values along the diagonal represent the model's precision in correctly identifying each class, with a particularly high true positive rate for 'car' at 0.96 and 'truck' at 0.95. Notably, the matrix reveals a marked misclassification rate with 'van' and 'misc' classes, evidenced by lower diagonal values of 0.68 and 0.81, respectively. Such insights are pivotal as they spotlight the model's strengths and areas that may benefit from further calibration, which is anticipated to be improved upon the subsequent application of the DeepSORT tracking algorithm

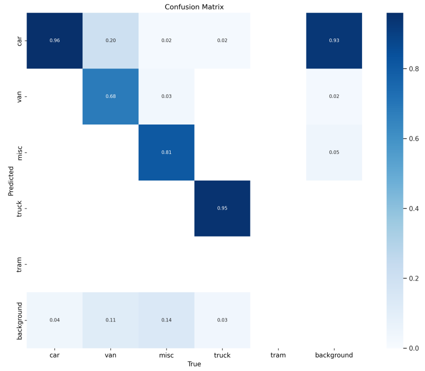


Figure 5. Confusion matrix of the object detection model during the testing phase before applying DeepSORT.

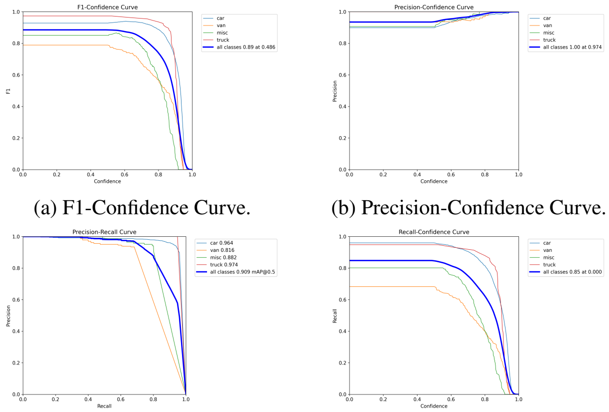
The testing phase performance, as illustrated in 6, provides a multifaceted evaluation of the object detection model. The Recall-Confidence Curve (Figure 6d) demonstrates a high recall at lower confidence thresholds, which declines as the threshold increases. This trade-off is expected as higher confidence levels typically yield fewer false positives but may also miss some true positives. The plateau at the higher recall rates for 'car' and 'truck' classes underscores the model's sensitivity to these objects.

Conversely, the F1-Confidence Curve (Figure 6a) reaches its apex at a moderate confidence level, suggesting an optimal balance between precision and recall is achieved at this point. Beyond this threshold, the F1 score diminishes, highlighting the diminishing returns of an overly stringent confidence criterion.

The Precision-Confidence Curve (Figure 6b) reveals a high precision score across all classes at high confidence levels. This indicates that when the model predicts an object with high certainty, it is usually correct, which is crucial for applications requiring high precision to avoid false alarms.

The Precision-Recall Curve (Figure 6c) further elucidates the model's performance, with the 'truck' class showing exceptionally high precision across all levels of recall. The 'van' class, however, exhibits lower precision, especially at higher recall levels, which could be attributed to fewer training instances or higher intra-class variance.

Overall, the results underscore the strengths of the model in detecting certain classes with high reliability while also pointing to areas where performance could be improved. The insights gained from these curves are instrumental for fine-tuning the model's thresholds and enhancing overall accuracy. Moreover, they establish a baseline against which the impact of incorporating DeepSORT for object tracking can be measured.



(a) F1-Confidence Curve.

(b) Precision-Confidence Curve.

(c) Precision-Recall Curve.

(d) Recall-Confidence Curve.

Figure 6. Testing phase performance metrics before applying DeepSORT. (a) F1-Confidence Curve showing the balance of precision and recall. (b) Precision-Confidence Curve indicating precision at different confidence levels. (c) Precision-Recall Curve reflecting the trade-off between precision and recall for varying thresholds. (d) Recall-Confidence Curve demonstrating the model's ability to correctly identify classes at different confidence thresholds.

The baseline performance metrics of our object detection model, summarized in Table 4, lay the groundwork for assessing the impact of the subsequent integration of DeepSORT. Analyzing across all classes, the model exhibits a strong precision (P) of 0.935 and a robust recall (R) of 0.848, with an overall mAP@0.50 of 0.909, indicating high reliability in detecting objects when the confidence threshold is set to 50%. However, the mAP@0.50:0.95 at 0.703 suggests there is room for improvement in detection consistency across different IoU thresholds.

5.3. DeepSORT Testing Stage

Finally, we explore the **testing stage post-DeepSORT implementation**, which showcases the impact of incorporating the DeepSORT algorithm on our model's tracking accuracy and efficiency. Comparisons are drawn with the baseline results to highlight the improvements or changes brought about by this integration.

By looking at both confusion matrices 5 and 7, the pre-DeepSORT matrix indicates a strong identification of 'truck' with a high degree of accuracy, while some confusion is evident between 'car' and 'van' classes. After integrating DeepSORT, it is observed that the accuracy for 'van' improves, suggesting that temporal information aids in distinguishing between similar object classes. The 'misc' category also shows a slight improvement, reinforcing the benefit of sequence data in object differentiation. Overall, the implementation of DeepSORT enhances the precision of tracking across frames, which is reflected in the increased clarity of the post-DeepSORT confusion matrix.

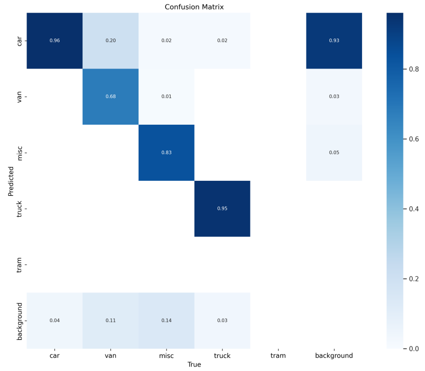


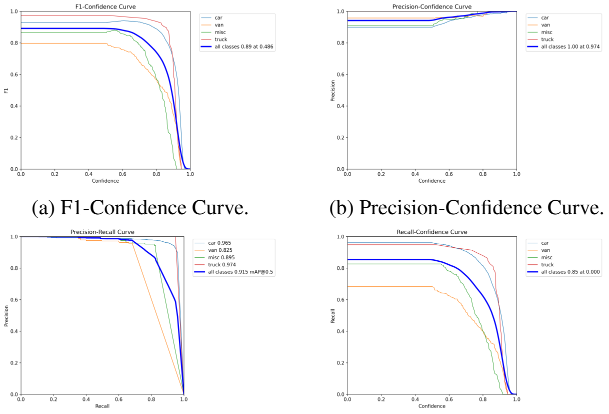
Figure 7. Confusion matrix of the object detection model during the testing phase after applying DeepSORT.

As a comparison between the results before and after applying the DeepSORT algorithm,it is nticiable that the performance curves (8a, 8b, 8c, 8d) post-DeepSORT integration show that the YOLOv5 model sustains its detection efficacy. The F1-Confidence Curve depicts a consistent precision-recall balance, while the Precision-Confidence Curve confirms high accuracy in high-certainty predictions. Notable is the improved precision for 'van' in the Precision-Recall Curve, suggesting that DeepSORT's temporal context enhances class distinction. The Recall-Confidence Curve remains high for low thresholds, indicating preserved sensitivity. Collectively, these figures (8) demonstrate DeepSORT's positive impact on model performance, particularly in tracking precision.

The enhanced performance metrics of our object detection model after the integration of DeepSORT, as detailed in Table 5, demonstrate notable advancements over the baseline metrics. A particularly significant increase is observed in the 'Van' class, where precision improved remarkably from 0.935 to 0.957, highlighting DeepSORT's effectiveness in accurately identifying this category. Across all classes, the application of DeepSORT has augmented

Table 4. Baseline Model Performance before Applying DeepSORT

Class	Images	Instances	P	R	mAP@.50	mAP@.50:.95
all	800	2667	0.935	0.848	0.909	0.703
Car	800	2258	0.899	0.96	0.964	0.783
Van	800	230	0.935	0.683	0.816	0.66
Misc	800	121	0.907	0.802	0.882	0.586
Truck	800	58	1	0.948	0.974	0.781



(a) F1-Confidence Curve.

(b) Precision-Confidence Curve.

(c) Precision-Recall Curve.

(d) Recall-Confidence Curve.

Figure 8. Testing phase performance metrics before applying DeepSORT.

the model's overall precision (P) to 0.941 and recall (R) to 0.855. This improvement signifies a more accurate and consistent object detection capability, especially when considering the increased mAP@0.50 to 0.915. This metric underscores the model's heightened reliability in detecting objects at a 50% confidence threshold. Furthermore, the rise in mAP@0.50:0.95 to 0.707 indicates a better detection performance across a range of Intersection over Union (IoU) thresholds, showcasing the model's enhanced versatility and robustness in varying detection scenarios. The consistency of this performance improvement across different object classes, such as Cars, Vans, and Miscellaneous objects, reinforces the effectiveness of DeepSORT in refining the model's object detection proficiency.

In frames 60 and 61, the Yolov5 model incorrectly classified the objects, mistakenly identifying a car as a van, as depicted in figures 9 and 11. Similarly, in frame 180, an object belonging to the 'Misc' category was misclassified as a van, as illustrated in 10 and 12. However, upon applying DeepSORT, these misclassifications were rectified, with the corrected classifications evident in 13 and 14. This demonstrates the efficacy of DeepSORT in enhancing the accuracy of object classification in the Yolov5 model.

The results from these stages collectively provide insights into the performance enhancements achieved through our methodology and pave the way for a comprehensive discussion on the effectiveness of combining YOLOv5 with DeepSORT for advanced object detection and tracking in autonomous vehicles.

6. Conclusion and Future Work

In conclusion, our project has made significant strides in creating an effective object detection and tracking system by successfully integrating YOLOv5 with DeepSORT. This combination has shown promising results in improving object detection consistency in video sequences.

For future work, several avenues can be explored to enhance the robustness and applicability of our system. Firstly, expanding and diversifying the dataset to include a broader range of scenarios will help in testing and improving the model's accuracy and reliability under various conditions. Secondly, implementing real-time processing capabilities will make the system more applicable in dynamic and time-sensitive applications. Additionally, exploring advanced machine learning techniques, such as reinforcement learning, could provide more sophisticated tracking and prediction capabilities. Finally, optimizing the algorithm for different environmental conditions and scalability to larger datasets will broaden the system's applicability in real-world scenarios. These improvements not only aim to refine the model's performance but also to contribute significantly to the field of computer vision.

Our Work can be accessed through this link: https://drive.google.com/drive/folders/1J1hlcGRPuSxILSY-xL4kcD2FfARag2Wl?usp=drive_link

6.1. Challenges,Lessons Learned, and Contribution

1. **Challenges:**

    - Finding a dataset with sequential order that provides enough examples to demonstrate the effectiveness of our algorithm was a significant challenge.
    - Integrating YOLOv5 with DeepSORT, without prior knowledge of either, required substantial

Table 5. Model Performance after Applying DeepSORT

Class	Images	Instances	P	R	mAP@.50	mAP@.50:.95
all	800	2667	0.941	0.855	0.915	0.707
Car	800	2258	0.899	0.961	0.965	0.784
Van	800	230	0.957	0.683	0.825	0.667
Misc	800	121	0.909	0.826	0.895	0.595
Truck	800	58	1	0.948	0.974	0.781

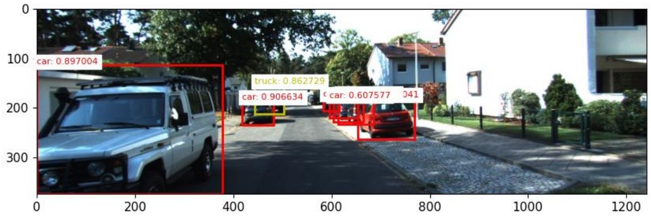


Figure 9. Initial Object Detection Frame 000054

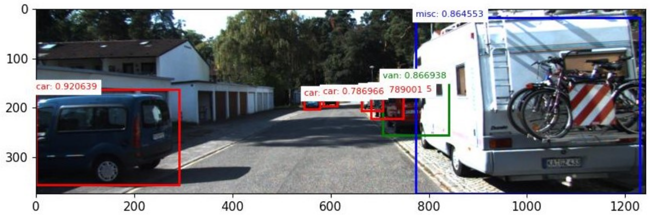


Figure 10. Initial Object Detection Frame 000178

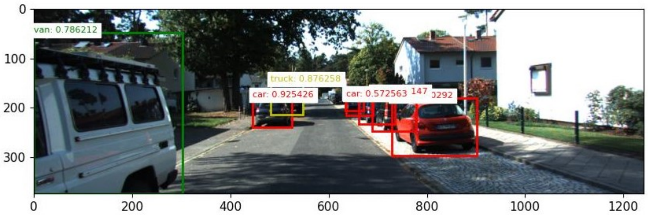


Figure 11. A Mislabeled Object Frame 000060 before DeepSORT

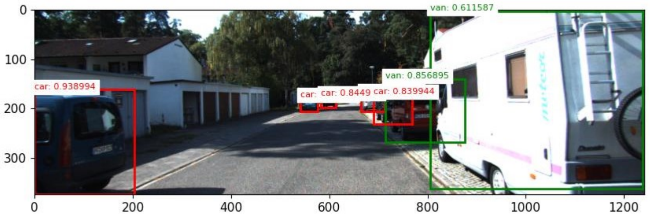


Figure 12. A Mislabeled Object Frame 000180 before DeepSORT

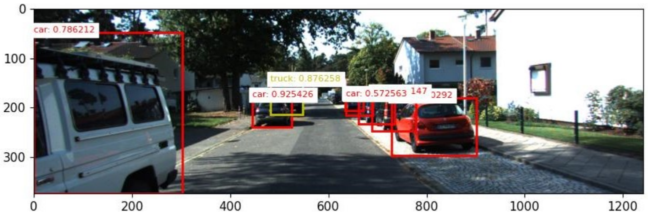


Figure 13. Consistent Object Detection Frame 000060 after Applying DeepSORT

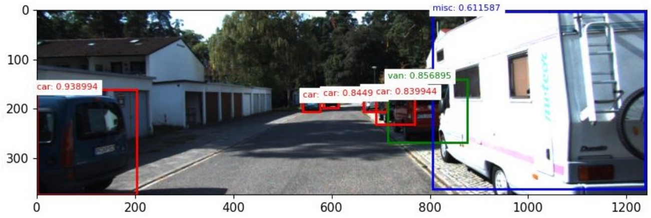


Figure 14. Consistent Object Detection Frame 000180 after Applying DeepSORT

time to understand and effectively implement both systems.

2. **Lessons Learned:**

   - Gained insights into object detection using YOLOv5 and its application across different datasets, not limited to KITTI.
   - Developed skills in managing and utilizing the KITTI dataset effectively.
   - Learned about optimizing the model for enhanced performance and gained familiarity with libraries relevant to our work.

3. **Contribution:**

   - Both team members put in considerable effort in implementing, testing, and optimizing the project, working collaboratively.
   - Collaboration extended to writing the paper and preparing the presentation, ensuring a cohesive and comprehensive delivery of our work.

References

[1] P. Azevedo and V. Santos. Yolo-based object detection and tracking for autonomous vehicles using edge devices. In D. Tardioli, V. Matellán, G. Heredia, M.F. Silva, and L. Marques, editors, *ROBOT2022: Fifth Iberian Robotics Conference*, volume 589 of *Lecture Notes in Networks and Systems*. Springer, Cham, 2023.

[2] M. Durve, S. Orsini, A. Tiribocchi, et al. Benchmarking yolov5 and yolov7 models with deepsort for droplet tracking applications. *European Physical Journal E*, 46(32), 2023.

[3] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research*, 2013. https://www.cvlibs.net/datasets/kitti/eval_tracking.php.

[4] Glenn Jocher et al. Yolov5. https://github.com/ultralytics/yolov5, 2020.

[5] KITTI Dataset Contributors. KITTI Vision Benchmark Suite. https://www.cvlibs.net/datasets/kitti/eval_tracking.php.

[6] F. Luetteke, X. Zhang, and J. Franke. Implementation of the hungarian method for object tracking on a camera monitored transportation system. In *ROBOTIK 2012; 7th German Conference on Robotics*, pages 1–6, Munich, Germany, 2012.

[7] R. Pereira, G. Carvalho, L. Garrote, and U.J. Nunes. Sort and deep-sort based multi-object tracking for mobile robotics: Evaluation with new data association metrics. *Applied Sciences*, 12:1319, 2022.

[8] I. Perera et al. Vehicle tracking based on an improved deepsort algorithm and the yolov4 framework. In *2021 10th International Conference on Information and Automation for Sustainability (ICIAfS)*, pages 305–309, Negombo, Sri Lanka, 2021. IEEE.

[9] A. Pujara and M. Bhamare. Deepsort: Real time & multi-object detection and tracking with yolo and tensorflow. In *2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, pages 456–460, Trichy, India, 2022. IEEE.

[10] Z. A. T. Rakotoniaina, N. E. Chelbi, D. Gingras, and F. Faulconnier. Liv-deepsort: Optimized deepsort for multiple object tracking in autonomous vehicles using camera and lidar data fusion. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–7, Anchorage, AK, USA, 2023. IEEE.

[11] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, 2017.