

Multi-Camera Enhanced Real-Time Content-Aware Vehicle Detection

Mohamed Mafaz^{1*}, I.Karthiga, S.Sharon Priya

¹Department of Artificial Intelligence and Data Science, GST Rd, Perungalathur, 600063, Chennai, India.

²Department Computer Science Engineering, Grand Southern Trunk Rd, Vandalur, 603210, Chennai, India

³Department Computer Science Engineering, Grand Southern Trunk Rd, Vandalur, 603210, Chennai, India

Corresponding Author. Email: mohdmafaz200303@gmail.com;

Contributing Authors. Email: karthiga.cse@crescent.education; sharonpriya@crescent.education;

No funds, grants, or other support was received.

Abstract—This paper introduces an open-source solution for real-time object awareness in the context of intelligent vehicles, designed to seamlessly adapt to simulation environments. The system transcends traditional vehicle detection, offering a comprehensive framework that includes object classification, precise location, and rotation estimation.

Leveraging a multi-camera setup, our approach provides a comprehensive understanding of the surrounding environment, extending beyond vehicle recognition to include various other objects. What sets our solution apart is its capability to efficiently reformat and transition to simulation environments like Blender, enabling integration and testing within virtual contexts.

The core components of our system encompass data collection, preprocessing, deep learning model selection, Use of Look up table, model training, efficient inference, and post-processing techniques. Designed for open-source collaboration, this solution is positioned for continuous improvement, adaptation to evolving needs, and addressing emerging challenges in the field of intelligent transportation and related domains.

This paper represents a foundational step towards establishing an accessible and adaptable real-time object awareness system, encouraging innovation and research in the realm of intelligent vehicles and their applications, both in the real world and in simulated environments.

I.

INTRODUCTION

Earlier approaches that led to real-time content awareness and its ability to extract location and orientation was a task that was accomplished by LiDAR [1], that is precise yet deals with its own problems such as cost, complexity, data volume, weather sensitivity, and environmental concerns. Camera-based solutions, especially when combined with multiple cameras, offer a more cost-effective, accessible, and practical approach for achieving real-time object awareness in intelligent vehicles and related applications [2].

Division of the entire algorithm into code blocks provides simplicity and easier modification for future improvements.

II.

MODEL BUILDING

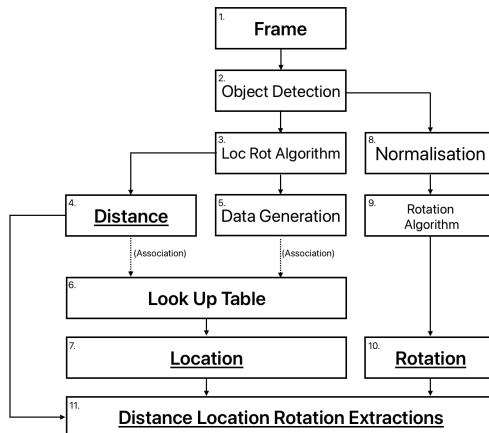


Fig. 1. Model Architecture. Our Detection network works by analysing one frame at a time. Frames are analysed for the specified (vehicle) object using YOLOv5s. Calculations are performed for each class of object to find Distance. Dedicated algorithms are utilised in association with the Look up Table to extract Location. Rotation Algorithm after normalisation is able to extract the objects orientation.

A. Object Detection

Using the YOLOv5s model in our object detection system has proven to be immensely beneficial [3]. Not only does it excel in accurately identifying and localising objects in real-time, but it also provides the key attributes required for advanced calculations, such as determining the distance to detected objects. YOLOv5s trained with COCO dataset [4] delivers precise bounding box coordinates, which enable us to calculate object distances effectively, particularly when we rely on height-based formulas [5].

The precision of the bounding boxes provided by the YOLOv5s model goes beyond accurate object localisation; it plays a pivotal role in enabling precise object cropping from the frame. This cropping operation is essential for subsequent

stages of our system, particularly in the context of rotation prediction.

By leveraging the precise bounding box coordinates, we can extract the detected object from the frame with high fidelity, ensuring that only the relevant object is included in the subsequent processing steps. This precision is crucial for eliminating extraneous information and optimising computational efficiency.

YOLO (You Only Look Once) model is renowned for its capability to provide real-time object detection. YOLO is designed to swiftly and accurately identify objects within images or video frames, making it a valuable tool for applications that require timely and responsive object recognition.

B. Loc Rot Algorithm, Distance

Distance estimation is a critical aspect of computer vision applications [6], especially in the context of object localisation and tracking. To achieve accurate distance measurements, we employ the triangle similarity method. This technique is based on the relationship between the actual width (W) of a known object or marker and its apparent width in pixels (P) when captured by a camera at a distance (D). By applying the formula :

$$F = (P \times D) / W \quad (1)$$

We can calculate the focal length (F) of the camera. This calculated focal length is a fundamental parameter for distance determination. The utilisation of such geometric principles is instrumental in ensuring the precision and reliability of distance calculations, which have broad implications for applications like robotics, augmented reality, and autonomous systems.

Continuing our exploration, as we vary the camera's distance from the object or marker, we can consistently apply the triangle similarity principle to ascertain the object's distance from the camera. The distance (D') between the camera and the object can be calculated using the formula:

$$D' = (W \times F) / P, \quad (2)$$

where W represents the known width of the object, F is the calculated focal length of the camera, and P denotes the apparent width in pixels. This approach allows us to dynamically measure the distance between the camera and the object as the camera's position changes. The capability to adapt to changing distances is pivotal in various applications [7], such as robotics and real-time tracking, where object distance awareness is fundamental to decision-making and control.

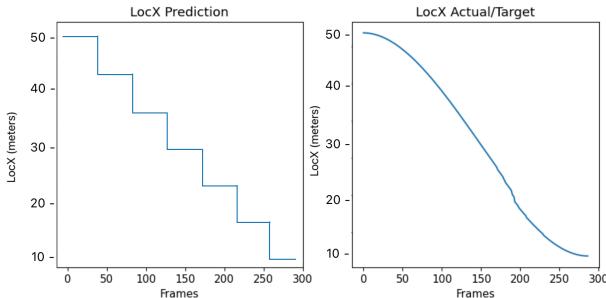


Fig. 2. Simulation Path on X Axis. Model Architecture was tested upon this simulation path, the vehicle followed this path for X axis.

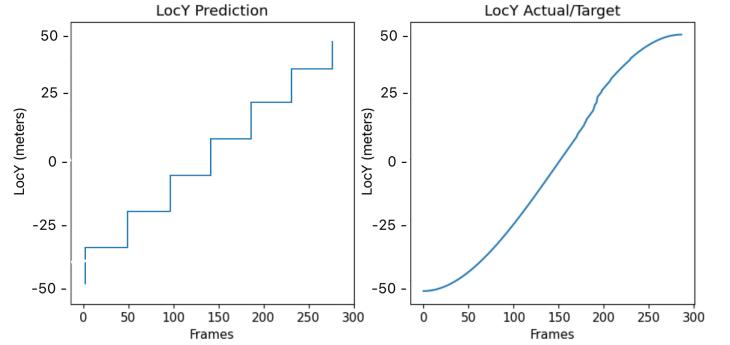


Fig. 2.1. Simulation Path on Y Axis. Model Architecture was tested upon this simulation path, the vehicle followed this path for Y axis.

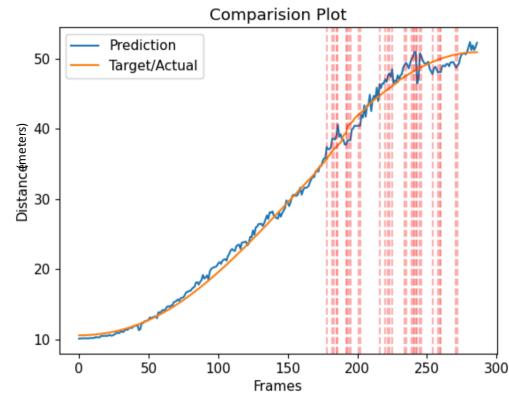


Fig. 3. Comparison Plot for Major Errors. This figure illustrates the comparison between actual distances and distances predicted by our distance equation. Red lines are indicative of the 'Major Error' threshold, set at 1.5 meters.

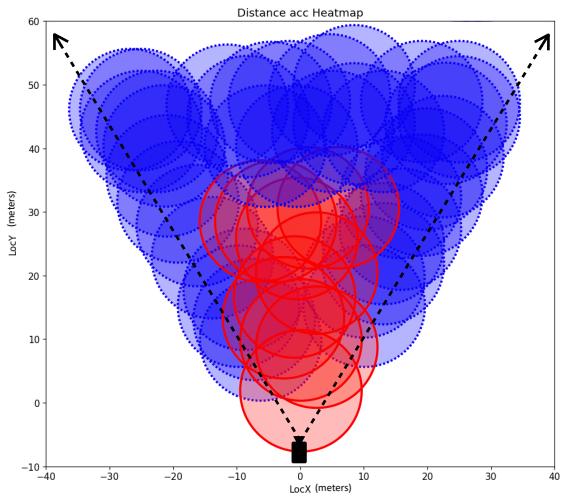


Fig. 4.1. Custom Heatmap. The heatmap is generated for a focal length (F) of 1964.28, with a confidence threshold set at 90%. The heatmap's circles are blue-dotted outline when the accuracy exceeds 90%.

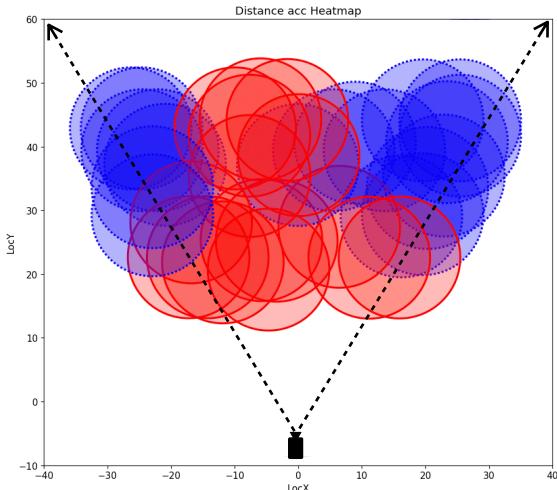


Fig. 4.2. Custom Heatmap. The heatmap is generated for a focal length (F) of 1375.0, with a confidence threshold set at 90%. The heatmap's circles are blue-dotted outline when the accuracy exceeds 90%.

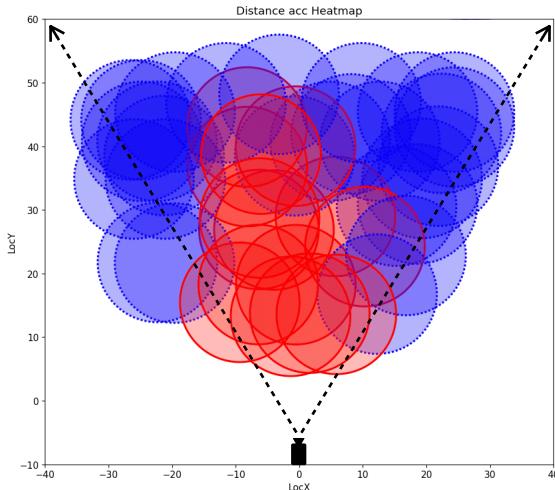


Fig. 4.3. Custom Heatmap. The heatmap is generated for a focal length (F) of 1774.19, with a confidence threshold set at 90%.

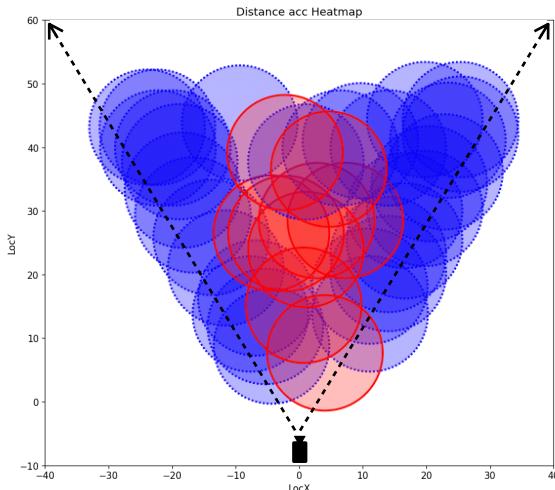


Fig. 4.4. Custom Heatmap. The heatmap is generated for a focal length (F) of 1617.64, with a confidence threshold set at 90%. The heatmap's circles are blue-dotted outline when the accuracy exceeds 90%.

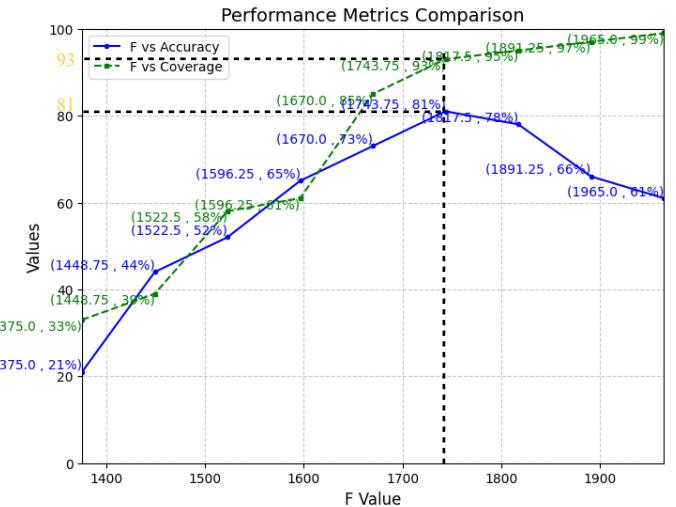


Fig. 5. Performance Metrics Comparison. Comparison of F value with Accuracy Metric and Coverage

Experimental Result Interpretation:

The experimental results, particularly the analysis of the F value derived from the equation (1), provide valuable insights into the impact of variations in vehicle height on accuracy and coverage. These findings shed light on the significance of maintaining precise estimations of vehicle height during both experiments and real-world applications. Average height of the car is estimated to be around 1.6 meters [7][8].

i. Impact of Underestimating Vehicle Height:

When experimenting with an error in car height that is less than the original, it is evident that the F value, and by extension, the accuracy and coverage, can be significantly affected. The results indicate that an error as small as 0.2 meters in car height can lead to a substantial decline in accuracy and coverage, with a potential reduction of up to 21%. This underlines the sensitivity of the system to underestimations of vehicle height.

ii. Impact of Overestimating Vehicle Height:

Conversely, experimenting with an error in car height that is greater than the original has a less drastic impact on accuracy. However, this scenario still results in a significant reduction in both accuracy and coverage, with the potential for a decline of up to 61% when there is a 0.4-meter error in car height.

These findings highlight the critical importance of accurately estimating vehicle height during system operation. To mitigate the adverse effects on accuracy and coverage, it is advisable to err on the side of caution and overestimate the vehicle's height rather than underestimating it. This emphasis on height accuracy is pivotal for applications where precise object positioning and tracking are essential, such as enhanced lane assist and autopilot systems.

In consideration, the results underscore the necessity of training the model to estimate vehicle height accurately during real-time operations and experiments. This precautionary approach is crucial for maintaining the reliability and effectiveness of the system, ensuring that it can deliver accurate and consistent results in dynamic and ever-changing real-world environments.

Formula of F value calculation for fig (4.1):

$$F = \frac{55 * 50}{1.4} = 1964.28$$

Formula of F value calculation for fig (4.2):

$$F = \frac{55 * 50}{2} = 1375$$

Formula of F value calculation for fig (4.3):

$$F = \frac{55 * 50}{1.54} = 1774.19$$

Formula of F value calculation for fig (4.4):

$$F = \frac{55 * 50}{1.7} = 1617.54$$

Vehicle Type	Dimension		
	Length (meter)	Width(meter)	Height (meter)
Compact Cars	4.2	1.8	1.5
Midsize Cars	4.8	1.8	1.7
Full Size Cars	5.4	1.8	1.4
Sports Cars	5	1.9	1.2
S U V s a n d Crossovers	5.1	2.1	1.6
Pickup Trucks	6	2	1.8
Minivans	5.4	2.1	1.8
Luxury Cars	5.4	1.8	1.4
Station Wagons	4.8	1.8	1.6

Table 1: Car Category and their Maximum Dimension

Rationale for Using Maximum Dimension of Each Car Class:

The selection of the maximum dimension of each car class [9], specifically in the context of length, width, or height, serves as a deliberate strategy in our methodology. This choice is grounded in the observation that the algorithm's accuracy experiences a more gradual decline when overestimating the dimension as opposed to underestimating it.

By focusing on the maximum dimension [10], we adopt a conservative approach that mitigates the adverse impact on accuracy. Underestimating the dimension can lead to more significant errors in positioning and tracking, especially in dynamic real-world scenarios. In contrast, overestimating the dimension offers a safety margin, allowing the algorithm to maintain a reasonable level of accuracy even in the presence of uncertainties.

This approach aligns with the overarching goal of our methodology, which prioritises robust and reliable object detection and tracking in diverse environments. By emphasising the maximum dimension, we enhance the algorithm's adaptability and its ability to provide accurate results while minimising the risk of performance degradation.

The choice to utilise the maximum dimension of each car class is a deliberate strategy that contributes to the resilience and accuracy of our methodology, as it minimises the potential pitfalls associated with dimension underestimation, particularly in real-world applications such as enhanced lane assist and autopilot systems.

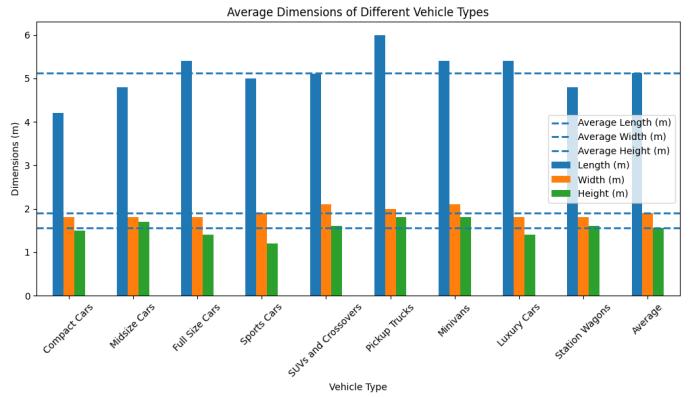


Fig 6: Car Category and Average dimension.

Dimension	Average Value (meter)
Height	1.5
Length	5.1
Width	1.9

Table 2: Average Values of Car dimension.

D. Look-up Table

In consideration of the research findings, it is evident that the utilisation of the triangle similarity method in conjunction with the custom heatmap has led to precise distance predictions, resulting in a substantial reduction of major errors. This work offers a valuable contribution to the fields of computer vision and object distance awareness, providing a practical framework for real-time applications. The incorporation of a confidence threshold, coupled with the visual representation of accuracy using the custom heatmap's blue dotted circles, is of significant importance for maintaining accuracy levels exceeding 90%. The implications of our research extend to a wide range of applications, including object tracking, robotics, and autonomous systems, where precise distance awareness holds a pivotal role in the context of decision-making and control. Our future work will focus on optimising and broadening the applicability of this methodology in diverse real-world scenarios.

C. Loc Rot Algorithm, Data Generation.

i. Lookup Table Utilisation

The prediction of object (vehicle) location is a critical component of many computer vision applications. In this study, we employ a lookup table to facilitate the prediction of object locations based on key attributes. The lookup table consists of several columns, including 'Area' (of bounding box), 'Center_x' (of bounding box), 'Center_y' (of bounding box), 'Distance' (from the training dataset), 'LocX' (from the training dataset), 'LocY' (from the training dataset), and 'Loc' (from the training dataset). Models built upon Dense Network have been used in this field of research [11].

The utilisation of the lookup table presents several advantages. First and foremost, it offers expedited access times, enhancing the efficiency of our location predictions. This efficiency is of paramount importance for real-time applications and training processes, as it reduces computational overhead. The lookup table accelerates the training phase, resulting in faster convergence and more time-efficient model development.

ii. Benefits of Lookup Table Access

Accessing the lookup table offers various advantages, including reduced access time, faster computation, and efficient model training [12]. The table serves as a valuable resource for data-driven location predictions, which are fundamental to many computer vision tasks. These advantages highlight the practicality of utilising a lookup table in our approach and its potential for enhancing the performance of object location prediction models.

In our location prediction method, we employ the function `get_Dis_Loc`, which takes several parameters and returns the predicted object location in terms of `LocX` and `LocY`. This function is designed to relate an input image to a pre-existing lookup table containing valuable data for distance and location calculations.

The process begins with the acquisition of the input image. Using computer vision techniques and a pre-trained model, we analyse the image and obtain object detection results. Specifically, we extract the centre coordinates (`Center_x`, `Center_y`) (fig 5) and dimensions (height '`h_`') of the detected objects.

Subsequently, we utilise a formula that involves the object's height and focal length (`f`) to calculate the expected distance (`d_`) from the camera. We then establish a link between this expected distance and the data stored in the lookup table. The lookup table is consulted to determine the objects in the vicinity that match the expected distance. The choice of data points is based on the object's position, specifically, whether it lies to the left or right of the image centre.

For each relevant data point, we compute the difference between the expected and recorded distance, as well as the disparity in centre coordinates (`Center_x`, `Center_y`). These differences are combined using a weighted sum with the '`dis_weight`' parameter to assess the best match.

The data point with the minimum combined difference is selected as the most likely prediction for the object's location. This prediction is represented by the '`LocX'` and '`LocY`' coordinates. In situations where an object prediction is unsuccessful, the previous data is retained to maintain continuity in plotting, ensuring a smoother visualisation process.

Our location prediction method, as implemented in the `get_Dis_Loc` function, offers a robust approach to estimating the precise location of detected objects, which is invaluable in various computer vision and object tracking applications.

$$\text{data_sub} = \begin{cases} \text{look_up_table}["LocX"] \geq 0, & \text{if } \text{Center_x} (1920/2) \\ \text{look_up_table}["LocX"] < 0, & \text{otherwise} \end{cases} \quad (3)$$

The optimisation of our data retrieval process is achieved through a strategic conditional statement. We evaluate the position of an object's centre, denoted as '`Center_x`', relative to the horizontal centre point of the image, which is `1920/2`. If '`Center_x`' is situated to the right of this centre point, we restrict our query to records in the lookup table where '`LocX`' is greater than or equal to `0`. Conversely, when '`Center_x`' falls to the left of the centre, our search focuses on records where '`LocX`' is less than `0`. This tailored approach effectively reduces search time by half, streamlining the query process to examine only the relevant 50% of records within the lookup table. This optimisation significantly enhances the efficiency of our location prediction method, catering to real-time and data-driven applications."

```
LocX, LocY = FindLocation(data_sub, d_, Center_x,
Center_y, dis_weight)
```

Where:

- 'data_sub' is a subset of the lookup table.
- 'd_' represents the expected distance from the object.
- 'Center_x' denotes the object's centre x-coordinate.
- 'Center_y' signifies the object's centre y-coordinate.
- 'dis_weight' is a weighting factor.

The function 'FindLocation' calculates 'LocX' and 'LocY' as follows:

1. Absolute difference between the 'Distance' in 'data_sub' and the expected distance 'd_' is computed.
2. Absolute difference between the 'Center_x' in 'data_sub' and the object's centre x-coordinate 'Center_x' is determined.
3. Absolute difference between the 'Center_y' in 'data_sub' and the object's centre y-coordinate 'Center_y' is calculated.
4. The 'Center_difference_' is computed as the sum of the absolute differences in x, y, and a weighted difference in distance.
5. The index of the minimum 'Center_difference_' value in 'data_sub' is identified.
6. 'LocX' and 'LocY' values corresponding to the identified index in the original lookup table 'data' are retrieved.

This formula and its description elucidate the method for ascertaining 'LocX' and 'LocY' based on the provided parameters and calculations.

$$\text{Distance_Difference_array} = \sum_{n=0}^i |\text{data_sub}["\text{Distance}"] - d_| \quad (4)$$

$$\text{Center_x_Difference_array} = \sum_{n=0}^i |\text{data_sub}["\text{Center}_x"] - \text{Center}_x| \quad (5)$$

$$\text{Center_y_Difference_array} = \sum_{n=0}^i |\text{data_sub}["\text{Center}_y"] - \text{Center}_y| \quad (6)$$

$$\text{Center_difference} = (\text{Center}_x_{\text{Difference}}_{\text{array}} + \text{Center}_y_{\text{Difference}}_{\text{array}} + (W * \text{Distance}_{\text{Difference}}_{\text{array}})) / 3 \quad (7)$$

$$\text{data_min} = \min(\text{Center}_difference)$$

Index Henceforth gets the corresponding record from the Look-up Table, record corresponding to Locational information.

Here 'Center_x_Difference_array' represents the absolute difference in the x-coordinate of the object's centre, 'Center_y_Difference_array' signifies the absolute difference in the y-coordinate of the object's centre, and 'Distance_Difference_array' corresponds to the absolute difference in the distance values. The parameter 'W' is the known width of the object's bounding box.

To pinpoint the optimal location prediction, we execute the following steps:

1. 'data_min' is identified as the index with the minimum 'Center_difference_' value within 'data_sub.'

2. 'LocX' and 'LocY' are acquired by referencing the corresponding values in the original lookup table 'data' using the 'data_min' index.

These steps collectively yield the predicted object location, represented by 'LocX' and 'LocY,' signifying its coordinates within the given context.

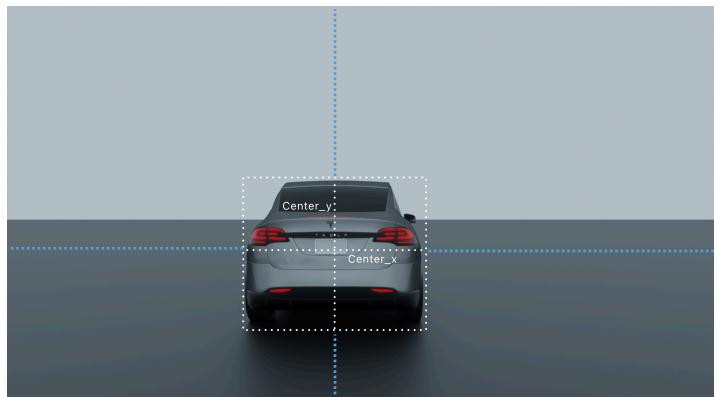


Fig. 7. Center_x, Center_y. A Visualisation of where the x-axis centre and y-axis centre lie. Local Centres, Global Coordinates.

E. Location Simulation

In real-time applications, the ability to obtain and utilise distance and location data swiftly is of paramount importance. Once we have successfully acquired this critical information, we can seamlessly integrate it into our simulation framework to generate real-time results. The obtained distance measurements, alongside the corresponding object locations, enable us to precisely position and track objects within our simulated environment. This dynamic tracking, informed by

the real-time data, supports a wide range of applications, from augmented reality and object tracking to autonomous systems and robotics.

The efficiency of our approach lies in its capability to deliver timely and accurate results. This is achieved by optimising the data retrieval process through strategies like the use of lookup tables and selective searching based on the object's position within the image frame. Such optimisations reduce computation time and support real-time decision-making.

Furthermore, the real-time results provide valuable insights for immediate action and control. For instance, in autonomous vehicles, these results can contribute to enhanced lane assist systems, adaptive cruise control, and overall safety. In augmented reality, real-time object tracking and positioning enable immersive experiences and interactive applications [12]. Our approach's real-time capability opens doors to innovative and responsive solutions across various domains, ensuring that our research findings are not only insightful but also practically applicable in fast-paced, dynamic environments.

Re-identification of vehicles within our system has been achieved through a lightweight and effective method based on sorting the vehicles from left side to right side. While it may not be the most accurate method, it offers an efficient solution that balances accuracy and computational efficiency. This approach has proven to be particularly suitable for real-time applications, where quick and reliable vehicle re-identification is essential for tasks such as tracking and monitoring.

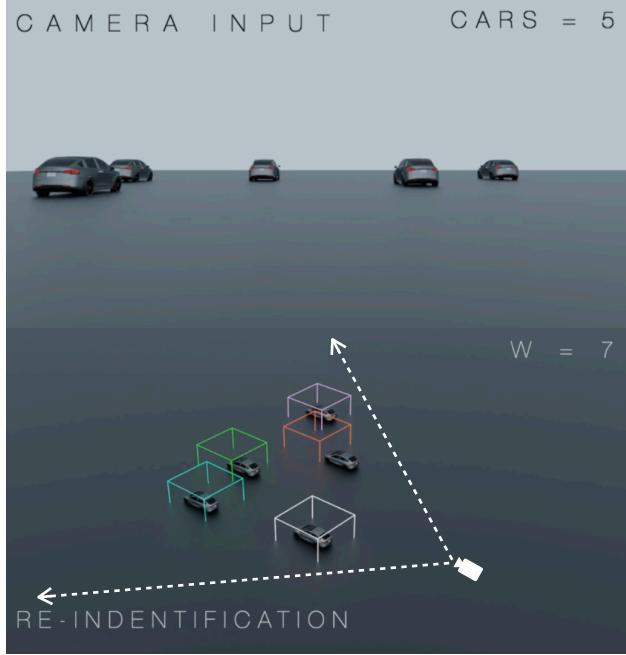


Fig. 8.1 Camera Input and 3D predicted Bounding box. W indicated the Distance and Location Multiplier (equation (7)). Simulation inside 3D environment - Blender. Cars - 5. Re-identification methodology applied.

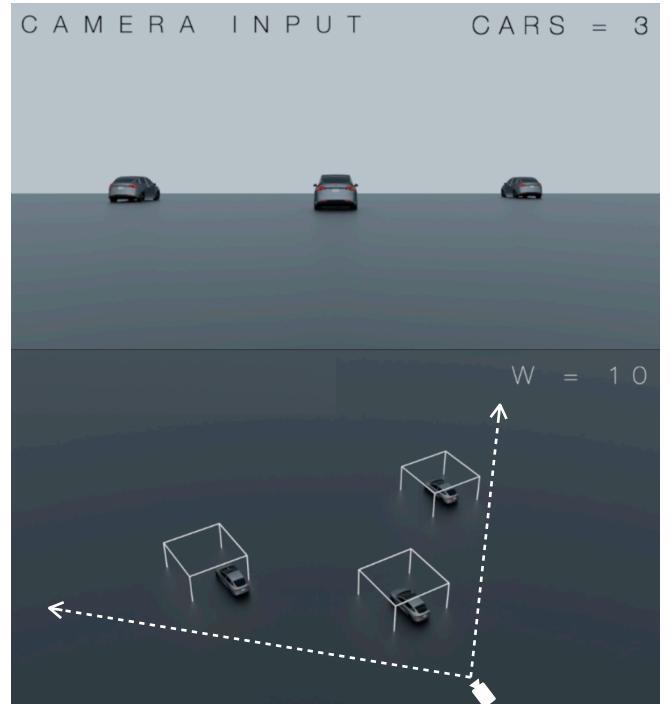


Fig. 8.2 Camera Input and 3D predicted Bounding box. W indicated the Distance and Location Multiplier (equation (7)). Simulation inside 3D environment - Blender. Cars - 3. Re-identification methodology not applied

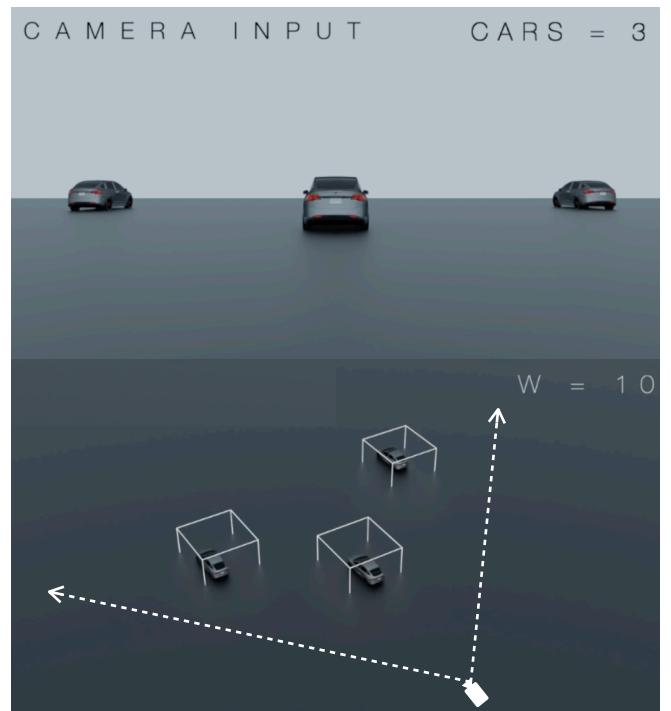


Fig. 8.3 Camera Input and 3D predicted Bounding box. W indicated the Distance and Location Multiplier (equation (7)). Simulation inside 3D environment - Blender. Cars - 3. Re-identification methodology not applied.

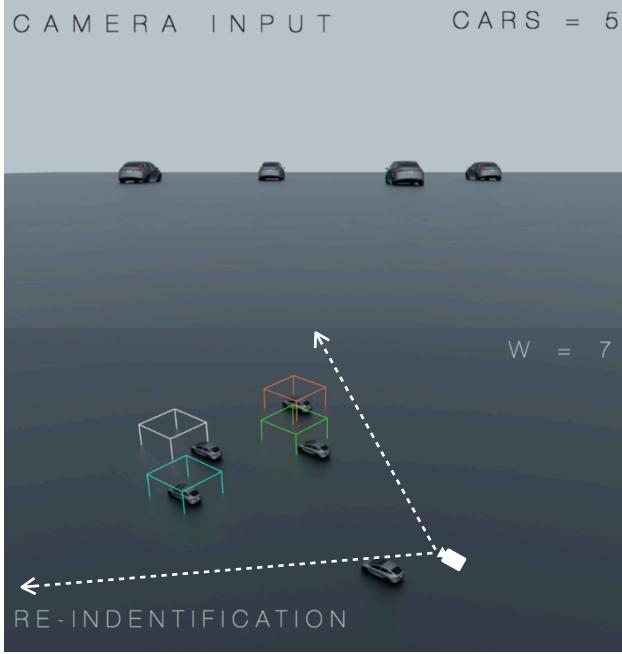


Fig. 8.4 Camera Input and 3D predicted Bounding box. W indicated the Distance and Location Multiplier (equation (7)). Simulation inside 3D environment - Blender. Cars - 5. Re-identification methodology applied.

F. Rotation - Normalisation

The image normalisation process is a critical step in our methodology, aimed at transforming the resolution of a cropped image to a predefined fixed resolution while meticulously preserving its aspect ratio [13]. This meticulousness is vital to ensure that the image's dimensions are adjusted without any stretching or distortion, thereby maintaining the integrity of the visual representation.

To achieve this precise transformation, we employ a two-fold approach:

i. **Aspect Ratio Preservation:** The initial step involves the calculation of a scaling factor that is instrumental in maintaining the aspect ratio. This scaling factor, denoted as SF, is derived from the desired final width, W_{Final} , and the original width of the image, $W_{Original}$. By scaling both dimensions consistently, we uphold the aspect ratio, which is an indispensable attribute for retaining the object's original proportions.

$$SF = \frac{W_{Final}}{W_{Original}}$$

ii. Resolution Adjustment with Padding: The second phase involves the adjustment of the image's dimensions. After

calculating the new height, H_{New} , by multiplying the original height $H_{Original}$, with the previously determined scaling factor, we can resize the image to the desired final width W_{Final} , and H_{New} . While this resizing action respects the aspect ratio, it may result in a disparity between the achieved height and the desired fixed height, H_{Final} . To rectify this discrepancy, padding is judiciously applied. Padding is executed by introducing a supplementary background, typically black, around the resized image. This padding, denoted as PI (Padded Image), is strategically distributed at the top and bottom of the image to reach the desired final height, H_{Final} . The amount of padding needed is precisely calculated to ensure that the final image adheres to the fixed resolution.

The padding calculation includes the determination of the padding at the top (PT) and the bottom (PB):

$$PT = \frac{H_{final} - H_{New}}{2}$$

$$PB = H_{Final} - H_{New} - PT$$

This meticulous and principled approach to image normalisation plays a pivotal role in maintaining visual consistency and fidelity, offering a dependable representation of the object in line with the defined fixed resolution. It underscores our commitment to precision and accuracy in object analysis and prediction, particularly in dynamic real-time scenarios.

G. Rotation Prediction Algorithm



Fig.9.1 Camera Input. Training Dataset.



Fig. 9.2 Normalised Object. Normalised Resolution is set to (800x800) pixels

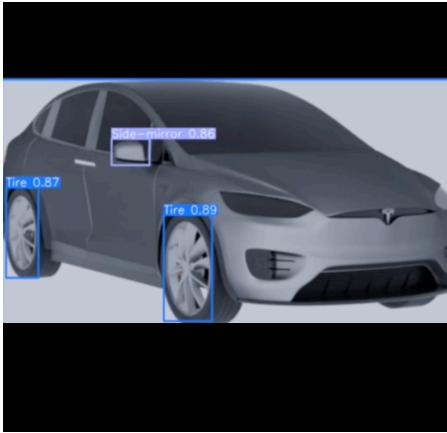


Fig. 9.3 Key Component Detection. Key Component were analysed for this were, Tires, Side view mirror, Number Plate.

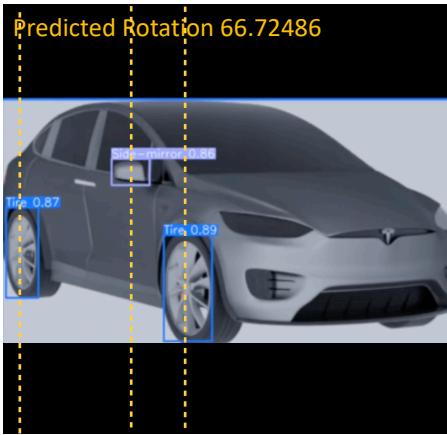


Fig. 9.4 Vertical Lines generation at Center of Local Horizontal Axis to Key Components and Prediction.

Steps to Convert Image to Training and Prediction:

Our methodology encompasses a structured process for converting an image into a format suitable for training and prediction. The following steps delineate this process:

i. Localisation of Custom Key Points in Local Metric (Normalised Image):

In the initial phase, we systematically identify and localise custom key points within the image. These key points are selected based on their significance to the object, such as tires, sideview mirrors, and number plates for a vehicle. This localisation is conducted within the local metric space of the normalised image, ensuring uniformity regardless of the image's original size or orientation.

ii. Data Storage - Horizontal Axis Centers in a Data frame:

Subsequently, we store the horizontal axis centres of these custom key points in a structured data frame. This data storage method facilitates organised access to key point information, an essential requirement for subsequent training and prediction tasks.

iii. Dataset Compilation and Training Data frame:

The data compilation phase involves gathering the dataset and preparing it for training. Notably, our approach focuses on training the data frame itself rather than the raw images. This unique strategy significantly accelerates the training process, resulting in a lightweight and highly scalable solution. Furthermore, this approach allows for fine-tuning key point predictions, the addition of more key points, and the enhancement of key point prediction quality.

iv. Training Using RandomForestRegressor: Training is executed using the RandomForestRegressor model. This choice of model aligns with our goal of lightweight and efficient training, enabling accurate rotation prediction with minimal computational resources.

v. Model Evaluation:

After training, the model's performance is evaluated to assess its effectiveness and reliability in predicting rotations accurately. (Fig 8)

vi. Rotation Prediction:

The final step involves utilising the trained model for rotation prediction, harnessing the insights from key points to make precise predictions regarding the object's orientation. (Fig7.4)

Rationale for Analysing Vertical Aspect of Key Components:

The decision to focus on the vertical aspect of key components within our methodology stems from a fundamental consideration of object behaviour in three-dimensional (3D) space, particularly when the object encounters slopes or variations in its orientation.

Key components, such as tires, sideview mirrors, and number plates, are strategically chosen because they offer stability and relative stationarity along their vertical axes, even when the object undergoes tilting or changes in its 3D orientation [14]. This inherent stability in the vertical aspect is a crucial factor when dealing with objects operating in dynamic and diverse real-world environments.

By analysing the vertical aspect of key components, we are better equipped to maintain the reliability and consistency of key point detection, even in scenarios where the object is tilted or situated on inclined surfaces. This approach enhances the robustness and adaptability of our methodology, making it suitable for a wide range of practical applications, including those involving 3D object tracking and orientation prediction in complex and dynamic settings.

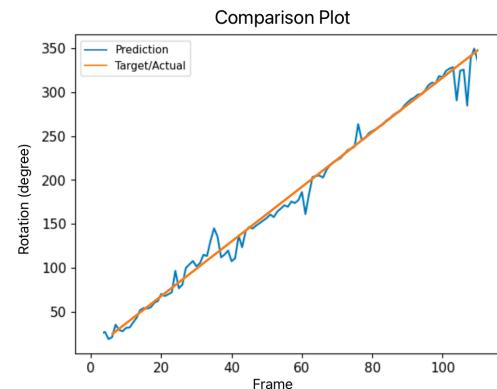


Fig. 10 Rotation Comparison plot. Vehicle rotates 0 - 360 degrees, and the Prediction/Actual data were compared.

III. SIMULATION AND DEPLOYMENT

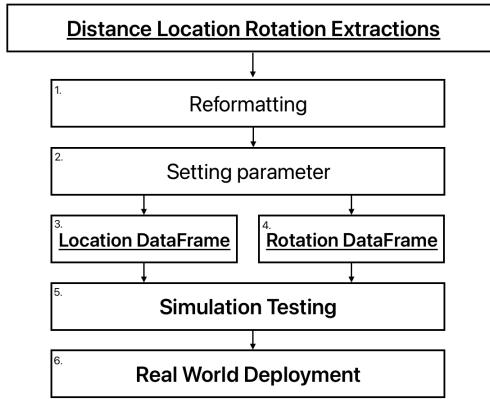


Fig. 11. Simulation and Deployment flowchart.

A. Reformatting and Setting parameter

The culmination of our paper's approach leads to the crucial phase of simulation and deployment, where the information gathered in terms of distance, location, and rotation is harnessed to create a practical and efficient system. This phase encompasses several key components that optimise the use of hardware and tailor the system to meet specific requirements.

i. Data Reformatting and Parameter Setting: The first step involves reformatting the data obtained from the paper's approach. This data, which includes distance, location, and rotation information, is structured and organized to facilitate seamless integration into the simulation and deployment framework. Additionally, critical parameters are set to configure the system and fine-tune its performance.

ii. Threshold Values and Hardware Usage Limitations: To ensure optimal utilisation of hardware resources, threshold values are established. These values serve as control parameters that influence various aspects of the system's behaviour. For instance, a maximum limit on the number of cars allowed to render can be set to prevent resource overuse and maintain system efficiency. These threshold values are instrumental in achieving a balance between realism and hardware resource constraints.

The simulation and deployment phase is a pivotal stage where the theoretical concepts and data gathered from the paper's approach are translated into practical application. This phase ensures that the system is not only capable of accurate predictions but also operates efficiently within the constraints of the hardware environment. By fine-tuning parameters and implementing threshold values, our approach optimises the utilisation of computational resources, making it a valuable solution for a variety of real-world applications.

B. Locational Data frame Rotation Data frame

i. Locational Information Analysis: The first component to be isolated is locational information. This segment of the data is dedicated to the precise positioning and tracking of objects within the simulated environment [15]. By segregating locational information, we can subject it to specialised analysis that is geared towards optimising object placement, movement, and interaction within the simulation. This separation enables us to fine-tune and enhance object localisation, a critical aspect for a variety of applications such as augmented reality, autonomous systems, and object tracking.

ii. Rotational Information Analysis: The second component earmarked for individualised analysis is rotational information. This segment focuses on predicting the orientation and rotation of objects, a task that is central to many real-world scenarios. By segregating rotational data, we can apply dedicated algorithms and techniques aimed at precise rotation prediction. This isolation ensures that the system can accurately anticipate and respond to changes in object orientation, contributing to enhanced performance in areas such as augmented reality and robotics.

C. Simulation Testing and Real-World Deployment

Real-World Deployment:

Once the methodology has demonstrated its efficacy in simulation testing, it proceeds to the real-world deployment phase. This practical application is characterised by the following key components:

i. Field Deployment: The methodology is deployed in real-world settings, such as urban environments, transportation systems, or industrial applications. Real-world deployment is essential to assess its performance under actual conditions and challenges.

ii. Data Collection: Data is collected in real-time from the deployed system, including distance, location, and rotation information. This data is invaluable for assessing the methodology's adaptability and performance in dynamic scenarios.

iii. Validation in Real Scenarios: Real-world deployment allows for the validation of the methodology's predictions and the comparison of its results with ground truth measurements. This validation process is instrumental in confirming the methodology's accuracy and reliability [16].

iv. Continuous Improvement: Insights gained during real-world deployment are used for continuous improvement. Any shortcomings or challenges encountered in practical applications drive further enhancements and refinements to ensure the methodology's effectiveness.

The combined phases of simulation testing and real-world deployment enable our approach to transition from theory to practical implementation. Through rigorous testing and real-world validation, our methodology proves its utility and reliability, offering a valuable solution for diverse applications in the fields of object tracking, augmented reality, and beyond.

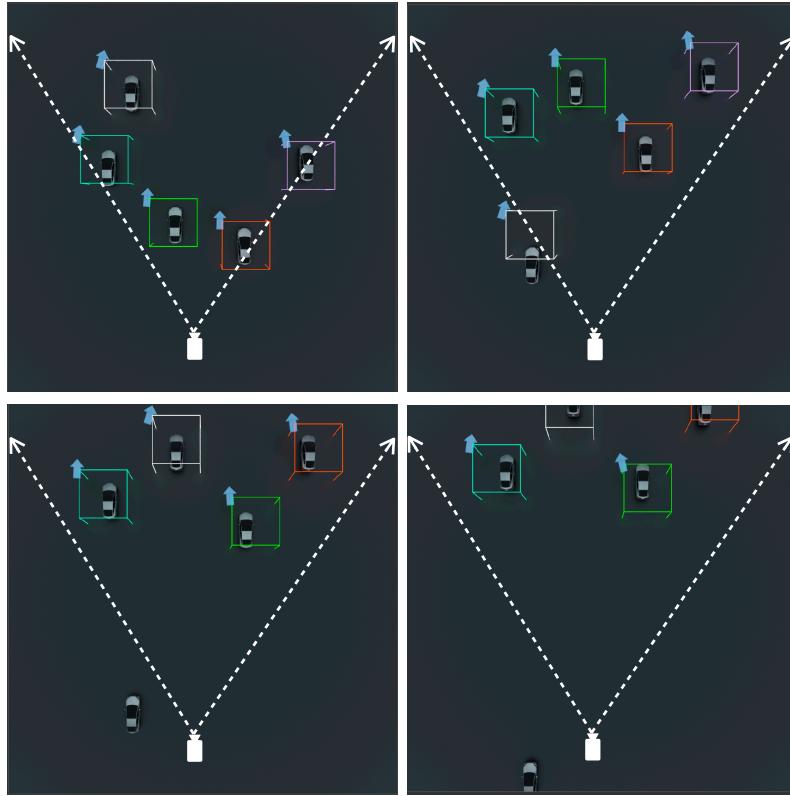


Fig. 12 Simulation Testing. Simulation Environment was testing open this paper. Prediction - 3D Bounding Box and Rotation depicted by the arrows corresponding to individual vehicle bounding box.

IV. ROTATION PREDICTION CALIBRATION

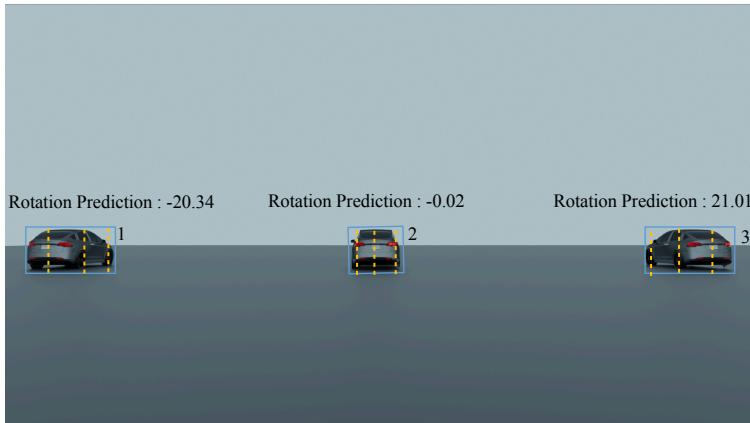


Fig. 13.1 Rotation Differences between 3 cars. All three cars are pointing in the same direction (i.e. 0 degrees). Due to the lack of calibration in the rotation algorithm there appears to be drastic difference in orientation.

As visible from fig 13.1 the rotation doesn't seem to be correct, there's a clear need for a calibration, the algorithm itself isn't predicting wrong, the error between actual and

predicted rotation is negligible. The reason behind this being is the actual working of the algorithm [17].

The algorithm even it being lightweight comes with its own unique problem, this would require its own calibration technique.

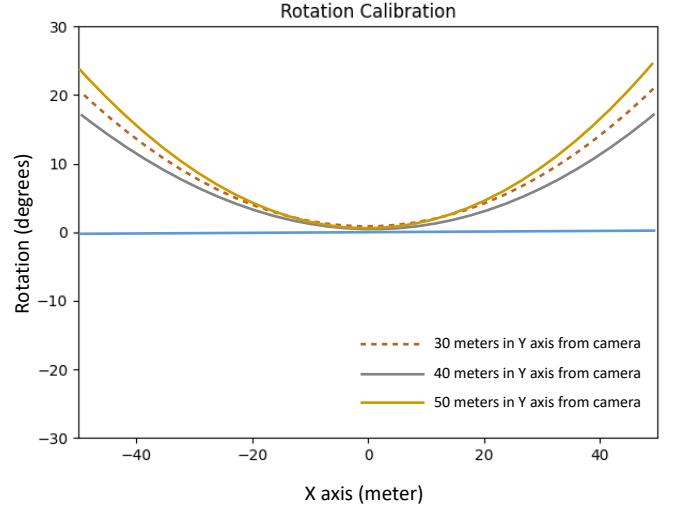


Fig. 13.2. Rotation Calibration Graph.

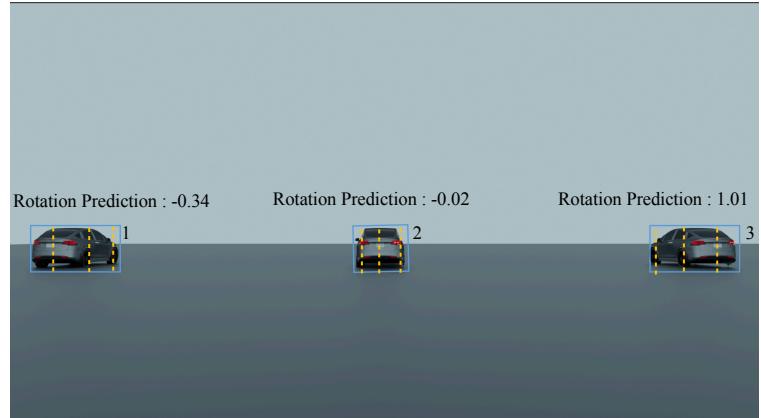


Fig. 13.3. Rotation Differences between 3 cars. After the Calibration

From experimentation, for this particular camera the calibration that will contribute by the 30 meter in the Y axis from the camera will act as a mean calibration for the rest of the algorithm.

Car 1, 2, 3 from fig 13.1 even though being in the same line on y axis and almost similar distance from the camera, see a huge difference in rotation before calibration.

After Calibration (fig 13.2) the rotation difference seems to be much more in control (fig 13.3).

V.

CONCLUSION

In conclusion, this paper introduces an innovative and practical solution for real-time vehicle awareness that is both cost-effective and highly accessible. By analysing key components and focusing on individual data segments, we have harnessed the power of computer vision to revolutionise object detection and prediction. This methodology holds promise for applications ranging from enhanced lane assist to autopilot systems, offering a versatile and robust solution for real-time content awareness [18]. As technology continues to advance, our methodology provides a solid foundation for future research and development in the field of computer vision and object tracking.

VI.

DECLARATIONS

A. Funding: This research received no external funding.

B. Conflict of interest/Competing interests: The authors declare that they have no conflicts of interest or competing interests that are relevant to the content of this article. The authors have no affiliations with or involvement in any organization or entity with any financial or non-financial interest in the subject matter or materials discussed in this manuscript.

C. Ethics approval: The study was conducted in accordance with ethical guidelines, and all applicable ethical standards were followed.

D. Consent to participate: Informed consent was obtained from all participants included in the study.

E. Consent for publication: All authors consent to the publication of this manuscript.

F. Availability of data and materials: Data and materials supporting the findings of this study are available upon request.

G. Code availability: The code for this study is currently not available as future plans for its use are in progress.

H. Authors' contributions:

i. Mohamed Mafaz contributed the most to the conception and design of the study, data collection, analysis, and interpretation of results.

ii. Dr. I.Karthiga and Dr. S.Sharon Priya provided significant guidance and assistance to the team throughout the research process.

VII.

REFERENCES

- [1] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharat Hariharan, Mark Campbell, and Kilian Q. Weinberge. Pseudo LiDAR from Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)
- [2] Clarke, R. (2001). Person location and person tracking - Technologies, risks and policy implications. *Information Technology & People*, 14(2), 206–231. <https://doi.org/10.1108/09593840110695767>
- [3] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *Vehicle Detection*. <https://doi.org/10.1109/cvpr.2016.91>

- [4] Neda Cvijetic in blogs.nvidia.com - 2019 Muhammad Abdul Haseeb, Jianyu Guan, Danijela Ristic
- [5] COCO dataset, <https://arxiv.org/pdf/1405.0312.pdf>
- [6] Blaschke, T. (2010). Object based image analysis for remote sensing. *Isprs Journal of Photogrammetry and Remote Sensing*, 65(1), 2–16. <https://doi.org/10.1016/j.isprsjprs.2009.06.004>
- [7] Durrant Axel Gräser. DisNet: A novel method for distance estimation from monocular camera. In book Computer Vision System (pp.457 - 469)
- [8] Agarwal, N., Chiang, C. C., & Sharma, A. (2019). A Study on Computer Vision Techniques for Self-driving Cars. *Lecture Notes in Electrical Engineering*, 629–634. https://doi.org/10.1007/978-981-13-3648-5_76
- [9] Satish Chandra. Effect of Lane Width on Capacity under Mixed Traffic Conditions in India. In *Journal of transportation engineering* 129(2)
- [10] J. Park, J.-H. Lee, S. H. Son, A Survey of Obstacle Detection using Vision Sensor for Autonomous Vehicles, 2016 IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications.
- [11] Yuan, X., Lu, Y., & Sarraf, S. (1994). Computer Vision System for automatic vehicle classification. *Journal of Transportation Engineering*, 120(6), 861–876. [https://doi.org/10.1061/\(ASCE\)0733-947x\(1994\)120:6\(861](https://doi.org/10.1061/(ASCE)0733-947x(1994)120:6(861)
- [12] Pinggera, U. Franke, R. Mester, High-performance long Range obstacle detection using stereo vision, 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2015).
- [13] Schneiderman, H., & Kanade, T. (2004). Object detection using the statistics of parts. *International Journal of Computer Vision*, 56(3), 151–177. <https://doi.org/10.1023/b:visi.0000011202.85607.00>
- [14] Nistér, D., Naroditsky, O., & Bergen, J. R. (2006). Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1), 3–20. <https://doi.org/10.1002/rob.20103>
- [15] Zhang, S., & Huang, P. S. (2007). Phase error compensation for a 3-D shape measurement system based on the phase-shifting method. *Optical Engineering*, 46(6), 063601. <https://doi.org/10.1117/1.2746814>
- [16] Luente, M. (1993). Interactive computation of holograms using a look-up table. *Journal of Electronic Imaging*, 2(1), 28. <https://doi.org/10.1117/12.133376>
- [17] Xavier Sebastian. Height of a car and why it matters!. In way.com

- [18] Bétaille, D., & Toledo-Moreo, R. (2010). Creating enhanced maps for Lane-Level vehicle navigation. *IEEE Transactions on Intelligent Transportation Systems*, 11(4), 786–798. <https://doi.org/10.1109/tits.2010.2050689>