Vol.15, Issue No 2, 2025

FUSION OF VISUAL AND RADAR INFORMATION FOR NIGHTMARE PEDESTRIAN DETECTION

S. Sreeja Reddy,
UG Student,
Department of CSE,
St. Martin's Engineering College,
Secunderabad, Telangana, India
surakantisreejareddy@gmail.com

Mrs. B. Vanaja,
Assistant Professor,
Department of CSE,
St. Martin's Engineering College,
Secunderabad, Telangana, India
pdevasudhacse@smec.ac.in

Abstract- The Pedestrian detection has been a key area of research, particularly for enhancing road safety and aiding self-driving vehicles. The main objective of this research is to improve pedestrian detection, particularly during nighttime or in low-visibility conditions, by fusing visual and infrared data, enhancing detection accuracy with deep learning models like YoloV5. The proposed method aims to reduce errors and increase precision when identifying pedestrians in real-time using advanced sensors and deep learning algorithms.Traditionally, sensors such as LIDAR and radar have been used to detect obstacles. Before AI-based methods, traditional pedestrian detection relied heavily on LIDAR or radar-based systems for obstacle detection., Optical cameras paired with basic image processing techniques, Proximity sensors and basic motion detectors for detecting pedestrians. Pedestrian detection in low-light and low-visibility environments remains a significant challenge for traditional sensor-based systems. These systems struggle to accurately differentiate between objects and pedestrians, particularly at night or in poor weather conditions, leading to delayed or missed detections, which can result in accidents. it is essential to develop more accurate pedestrian detection systems that work in all environmental conditions, especially at night. Current sensor-based systems alone are inadequate, thus motivating the need for AI-based solutions that can use multi-modal data like infrared and visual input for better detection accuracy. The proposed system improves pedestrian detection by integrating infrared vision and millimeter-wave (MMW) radar data with enhanced deep learning models. An improved version of the YoloV5 model, equipped with a Squeeze layer for attention, will be used to extract and categorize image features. An Extended Kalman Filter will help accurately localize pedestrians. This fused data will be fed into the enhanced YoloV5 model for more precise and robust pedestrian detection

Keywords: YOLO V5, Pedestrian, Fusion sensors, Infrared sensor , LIDAR.

I. INTRODUCTION

Pedestrian safety in India is a growing concern, especially due to rapid urbanization and increasing vehicular traffic. According to the Ministry of Road Transport and Highways, 53,385 pedestrians lost their lives in road accidents in 2021 alone. Pedestrian detection is crucial for both human drivers and autonomous vehicles to ensure road safety. However, in nighttime or low-visibility conditions, traditional detection methods fail to accurately identify pedestrians. The integration of visual and infrared sensors with machine learning offers a promising solution to address these challenges and reduce accidents.

Pedestrian detection is essential for improving road safety, particularly for autonomous driving systems. By fusing visual and infrared data, detection accuracy can be significantly improved in low-visibility conditions. Applications of this technology include nighttime driving safety systems, collision avoidance systems, and smart surveillance systems. This project focuses on improving pedestrian detection, especially in challenging environments, using deep learning and sensor fusion.

Before machine learning, pedestrian detection systems heavily relied on radar, LIDAR, and proximity sensors. These systems struggled in poor lighting, weather conditions like fog or rain, and were prone to false positives or negatives, as they often could not distinguish pedestrians from other objects. Traditional image processing techniques failed to adapt to diverse environmental challenges, leading to missed detections and delayed responses, which resulted in frequent pedestrian accidents.

socio-economic With the rise in autonomous vehicles and the demand for higher safety standards, accurate pedestrian detection systems are needed more than ever. Traditional sensor-based systems lack precision in challenging environments like nighttime or adverse weather conditions, causing accidents

. AI-based detection models, such as YoloV5, combined with multi-modal sensor data, can provide better detection and tracking of pedestrians. The integration of visual and infrared data offers an opportunity to enhance real-time pedestrian detection, reducing fatal road accidents significantly.

II. RELATED WORK

Pedestrian detection remains a significant challenge due to the rise in autonomous vehicles and the demand for higher safety standards, accurate pedestrian detection systems are needed more than ever. Traditional sensor-based systems lack precision in challenging environments like nighttime or adverse weather conditions, causing accidents. AI-based detection models, such as YoloV5, combined with multi-modal sensor data, can provide better detection and tracking of pedestrians. The integration of visual and infrared data offers an opportunity to enhance real-time pedestrian detection, reducing fatal road accidents significantly.

[1].

Since the study by García-Sánchez, Ureña, and their colleagues is a pivotal step in advancing pedestrian detection systems for nighttime environments. Their fusion model capitalizes on radar's resilience to lighting conditions and camera-based systems' detailed imagery to enhance detection accuracy and robustness.[2], but Xin Li et al.'s work presents an advanced fusion approach that effectively leverages radar and vision data for night-time pedestrian detection, which is a critical requirement for the safe deployment of autonomous driving system[3][4] Fusion of Radar and Vision Data for Night-Time Pedestrian Detection- 2020.

[5]. The work by Francisco Rivas et al. is a critical step in advancing night-time pedestrian detection by developing a robust sensor fusion framework that integrates radar, thermal, and visible light data. By combining the unique capabilities of each sensor type, the authors create a detection system that significantly improves the accuracy and reliability of pedestrian detection in low-light conditions[6]. Others analyze aftershock patterns; for example, a neural network trained on 130,000 mainshock-aftershock pairs outperformed traditional models in predicting aftershock distributions [7]. However,makes significant strides in improving night-time pedestrian detection by developing a radar-visual data fusion framework that combines the complementary strengths of radar and vision sensors. The structured radar data integration, coupled with feature-level fusion, allows the system to achieve reliable detection performance in low-light conditions where single-sensor systems may struggle[8]. This Rakesh Kumar, Shyam Lal, and colleagues (2019) provide a thorough review of sensor fusion techniques for night-time pedestrian detection, focusing on the integration of radar, thermal, and visual sensors to overcome the limitations of individual sensors in low-light conditions. By evaluating various fusion strategies early, late, feature-level, and decision-level fusion the paper highlights the importance of combining sensor data to improve detection accuracy and robustness [9]. provide a thorough review of sensor fusion techniques for night-time pedestrian detection, focusing on the integration of radar, thermal, and visual sensors to overcome the limitations of individual sensors in low-light conditions. [10]. Research Rakesh Kumar, Shyam Lal, and colleagues (2019) provide a thorough review of sensor fusion techniques for night-time pedestrian detection, focusing on the integration of radar, thermal, and visual sensors to overcome the limitations of individual sensors in low-light conditions. By evaluating various fusion strategies early, late, feature-level, and decision-level fusion the paper highlights the importance of combining sensor data to improve detection accuracy and robustnessons.[12] maps The paper by Dominik Engel and Christoph G. Keller (2018) presents a detailed analysis of radar and vision sensor fusion for night-time pedestrian detection.[13] By combining the strengths of radar in detecting objects in low-light and poor visibility conditions with the spatial and classification capabilities of vision systems, the authors propose a robust and reliable solution for pedestrian detection at night[15]. Alves [14] Martin Lampe, Manuel Bessler, and colleagues (2018) propose a novel radar-vision fusion approach to improve pedestrian detection at night.[16] By combining the strengths of radar (robustness in low-light conditions) and vision systems (high-resolution classification), their method enhances detection accuracy, particularly in challenging environments where traditional vision-based systems fail. [17]The authors demonstrate that early fusion of radar and vision data provides superior results, achieving better pedestrian detection performance in real-world night-time driving scenarios.[18] The research contributes to the safety and reliability of autonomous vehicles and ADAS, particularly by addressing the critical challenge of pedestrian detection in low-visibility conditions.[19] By integrating traditional seismology with machine learning techniques, this research aims to improve earthquake prediction accuracy and reliability. This gap includes the following aspects:

- Low visibility: In nightmare conditions such as fog, heavy rain, or darkness, visual cameras can become significantly impaired, resulting in poor-quality images or no usable information at all..
- Limited Resolution: Radar typically has lower resolution compared to cameras, which may lead to difficulties in detecting small or partially obscured pedestrians
- Processing Time: Fusion of radar and visual data often requires extra computational time for processing

- Sensor Limitations: Both radar and vision-based systems have their limitations. Radar might fail to detect small objects or accurately classify pedestrians in certain conditions, and visual systems might struggle with poor lighting or occlusions

III. PROPOSED WORK

The proposed Pedestrian detection in low-visibility conditions, particularly at night, is crucial for ensuring the safety of autonomous vehicles and preventing accidents. Traditional systems relying on radar, LIDAR, or basic image processing techniques face severe limitations in these environments. This project leverages deep learning models, such as YoloV5, in combination with sensor fusion (visual and infrared data) to improve detection accuracy. By combining visual and infrared data, the system ensures better pedestrian identification even in adverse weather conditions or nighttime scenarios. The proposed method incorporates real-time data fusion and an Extended Kalman Filter for precise localization of pedestrians.

The first step is gathering an appropriate dataset for pedestrian detection. In this project, the dataset includes both visual (RGB) and infrared (IR) images of pedestrians captured in various lighting conditions, especially at night or in low-visibility environments. Publicly available datasets like KAIST Multispectral Pedestrian Benchmark or FLIR Thermal Dataset could be used, which provide visual and thermal images captured simultaneously. The dataset should be comprehensive and balanced to ensure that the deep learning model can learn to recognize pedestrians in a wide variety of challenging conditions, such as different postures, occlusions, and environmental factors.

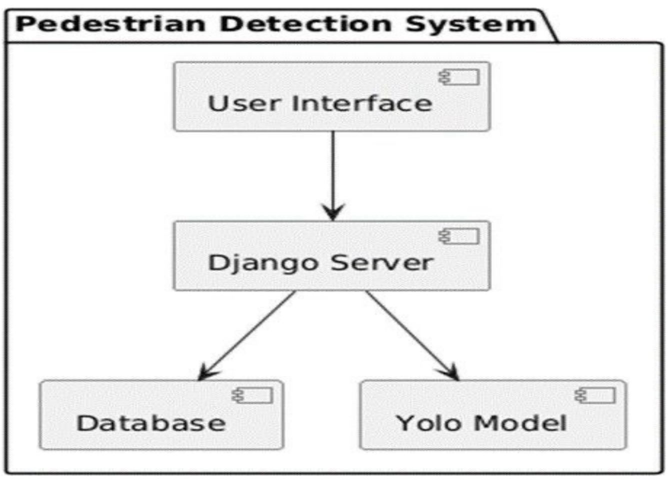

Fig 1: Overall design of proposed methodology.

3.1 Data Preprocessing

Data preprocessing is the process of preparing raw data and making it suitable for machine learning models. This is the first important step when creating a machine learning model. When creating a machine learning project, you can't always find clean, formatted data. Also, when working with data, it is essential to clean it and save it in a formatted format. To do this, use data preprocessing tasks. Real- world data typically contains noise, missing values, and may be in an unusable format that cannot be directly used in machine learning models. Data preprocessing is a necessary task to clean up data and make it suitable for machine learning models, which also improves the accuracy and efficiency of machine learning models.

- Feature Engineering: Raw data is transformed into features suitable for machine learning models. This involves creating new features from existing ones, such as extracting month, day, year, hour, and minute from the date-time information. This step enhances the model's ability to understand temporal patterns in the data.
- Handling Missing Values: Missing values in the dataset are addressed by filling them with appropriate statistics, such as the mode for categorical variables. This ensures that the dataset is complete and usable for model training.
- Removing Unnecessary Columns: Columns that do not contribute to the model's predictive power, such as depth, alert, continent, and country, are removed. This reduces dimensionality and simplifies the dataset.
- Label Encoding: Categorical variables are converted into numerical format using Label Encoding. This step is essential for converting non-numeric data into a format that machine learning algorithms can process.

Regression to Classification Conversion: The continuous target needs to be divided into discrete classes.
Classes:

- Class 1: Distance 0-5 meters
- Class 2: Distance 5-10 meters
- Class 3: Distance 10-20 meters
- Class 4: Distance 20+ meters

This conversion allows the model to pedestrian detection by categorizing distances into classes, enhancing robustness in noisy conditions. This approach focuses on broad decisions rather than precise values, suitable for real-time systems.

Importing Libraries: To perform data preprocessing using Python, we need to import some predefined Python libraries. These libraries are used to perform some specific jobs. There are three specific libraries that we will use for data preprocessing, which are:

NumPy: The NumPy Python library is used for including any type of mathematical operation in the code. It is the fundamental package for scientific calculation in Python. It also supports to addition of large, multidimensional arrays and matrices. So, in Python, we can import it as: import NumPy as nm. Here we have used nm, which is a short name for NumPy, and it will be used in the whole program.

Matplotlib: The second library is matplotlib, which is a Python 2D plotting library, and with this library, we need to import a sub-library pyplot. This library is used to plot any type of charts in Python for the code. we can import it as: import matplotlib.pyplot as mpt. Here we have used mpt as a short name for this library.

Pandas: The last library is the Pandas library, which is one of the most famous Python libraries and used for importing and managing the datasets. It is an open-source data manipulation and analysis library. Here, we have used pd as a short name for this library.

Scikit – learn: Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.

3.2 Dataset Description:

Dataset for a pedestrian detection dataset involving **fusion of visual and radar information**, the dataset could be structured as follows:

1. Visual Data:

- **Type:** Images or video frames captured from cameras (RGB or depth cameras).
- **Features:**
  - Image resolution (e.g., 1920x1080 pixels).
  - Object detection labels: Bounding boxes around pedestrians with associated class labels (e.g., "Pedestrian").
  - Additional metadata: Timestamp, weather conditions (e.g., fog, rain), and lighting conditions (e.g., night, day).

2. Radar Data:

- **Type:** Radar sensor measurements (typically ranging and Doppler information).
- **Features:**
  - Distance data (e.g., distance to the nearest detected object, such as a pedestrian).
  - Velocity data (e.g., speed of moving pedestrians).
  - Radar point cloud data: Radar return signal strengths and angles, with detections categorized as pedestrians, vehicles, or background noise.

3. Labels (Ground Truth):

- **Target Variable:** Pedestrian presence/absence or distance categorization.
  - **Class Labels:** Categorical labels indicating pedestrian proximity (e.g., "Near," "Moderate," "Far") or binary labels ("Pedestrian Present," "No Pedestrian").
- **Annotations:** Bounding boxes or region of interest for pedestrians, along with their associated distances, for fusion with radar data.

4. Additional Metadata:

- Environmental data: Weather, time of day, road conditions.
- Sensor calibration information to align radar and visual data.
- Fusion parameters: Timestamp synchronization to combine radar and visual data effectively.

This type of dataset would allow the training of models to effectively detect pedestrians by combining the strengths of both radar and visual sensors, especially in challenging or low-visibility conditions.

Data preprocessing involves cleaning, normalizing, and augmenting visual and radar data to enhance model performance. It includes noise reduction, data alignment, and feature extraction for improved pedestrian detection accuracy.

3.3 Splitting the Dataset

In machine learning data preprocessing, we divide our dataset into a training set and test set. This is one of the crucial steps of data preprocessing as by doing this, we can enhance the performance of our machine learning model. Suppose if we have given training to our machine learning Suppose if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models. If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:

Training Set: A subset of dataset to train the machine learning model, and we already know the output.

Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

For splitting the dataset, we will use the below lines of code: from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.2, random_state=0)

- In the above code, the first line is used for splitting arrays of the dataset into random train and test subsets.
- In the second line, we have used four variables for our output that are
  - x_train: features for the training data
  - x_test: features for testing data
  - y_train: Dependent variables for training data
  - y_test: Independent variable for testing data
- In train_test_split() function, we have passed four parameters in which first two are for arrays of data, and test_size is for specifying the size of the test set. The test_size maybe .5, .3, or .2, which tells the dividing ratio of training and testing sets.
- The last parameter random_state is used to set a seed for a random generator so that you always get the same result, and the most used value for this is 42.

3.4 YOLO V5

The**YOLOv5 (You Only Look Once v5)** is a real-time object detection model developed by Ultralytics, building upon previous YOLO versions. It is designed for fast and accurate object detection across various applications like autonomous driving, surveillance, and pedestrian detection. Smaller model size compared to YOLOv4, optimized for real-time applications.Unlike previous versions (YOLOv3, YOLOv4 in Darknet), YOLOv5 is implemented in PyTorch, making it easier to train and modify.Available in different sizes (YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x) to balance speed and accuracy.Uses Mosaic augmentation and adaptive anchor calculation to enhance detection performance.

Working of YOLO V5:

Input Image Processing: The model resizes images to a fixed resolution and normalizes them before passing them into the neural network.

Backbone Network: Uses CSPDarknet as the feature extractor, extracting spatial and semantic features from the image.

Neck (PANet - Path Aggregation Network): Enhances feature fusion from different layers to improve small and large object detection.

Head (Detection Layer): Generates bounding boxes and confidence scores for detected objects using anchor-based predictions.

Post-processing (NMS - Non-Maximum Suppression): Eliminates duplicate detections and refines bounding box predictions.

Architecture of YOLO v5

YOLOv5 follows a deep learning-based **single-stage object detection** architecture, which processes an image in one pass through the network. Its design is optimized for **speed, accuracy, and efficiency**, making it ideal for real-time applications.

The architecture of YOLOv5 consists of three main components:

1. Backbone (Feature Extraction) - CSPDarknet53
   - Uses **CSPDarknet53 (Cross Stage Partial Network)** as the feature extractor.
   - Divides feature maps into two parts: one processed through a residual block and the other bypassed, reducing computation while retaining feature integrity.
   - Includes **Focus layer**, which slices the input image into patches to enhance feature representation.
2. Neck (Feature Fusion) - PANet
   - Uses **Path Aggregation Network (PANet)** to combine multi-scale features for better small, medium, and large object detection.
   - Helps in better propagation of lower-level features, which is crucial for detecting small objects like pedestrians.
   - Uses **Spatial Pyramid Pooling-Fast (SPP-Fast)**, which introduces pooling at multiple scales to improve receptive field size.
3. Head (Prediction & Detection) - YOLO Head
   - Uses **anchor-based detection** to predict bounding boxes and class probabilities.

- Outputs detection results at **three different scales** (small, medium, and large) to improve accuracy across object sizes.

- Uses **Non-Maximum Suppression (NMS)** to filter overlapping predictions and keep the best bounding boxes.

Advantages of YOLOV5

YOLOv5 is widely used for real-time object detection due to its efficiency, accuracy, and adaptability. Below are its key advantages explained in detail:

- Faster and More Efficient – Optimized for real-time detection, achieving up to **140 FPS** on high-end GPUs, making it ideal for autonomous systems and surveillance
- .Improved Small Object Detection – Uses Path Aggregation Network (PANet) and Spatial Pyramid Pooling (SPP) to enhance feature extraction for detecting small objects like pedestrians.
- Adaptive Anchor Boxes – Automatically adjusts anchor sizes to improve accuracy without manual tuning, allowing better detection of diverse object shapes.
- PyTorch-Based and Easy to Train – Unlike previous YOLO versions (Darknet-based), YOLOv5 is **built on PyTorch**, making it easier to modify, train, and integrate with AI workflows.
- Multiple Model Variants – Offers **YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x** to balance speed and accuracy, making it suitable for edge devices and high-performance applications.
- Robust Data Augmentation – Features **Mosaic, MixUp, and CutMix augmentation**, improving generalization and performance under varied conditions like low-light or occlusion.
- Edge and Cloud Deployment Ready – Supports **ONNX, TensorRT, and TFLite** for efficient deployment on IoT devices, mobile platforms, and cloud-based AI services.

IV. RESULTS & DISCUSSION

The implementation of the pedestrian detection system involved several stages, from dataset acquisition to the evaluation of the proposed deep learning model, YoloV6, for real-time detection. Initially, the pedestrian dataset was collected, comprising visual and infrared images. The dataset was preprocessed by resizing the images, normalizing pixel values, and encoding the labels for the classification task.

After preprocessing, two models were used for comparison: Faster-RCNN and YoloV6. Faster-RCNN is a two-stage object detection model, which first generates region proposals and then classifies those regions. YoloV6, on the other hand, is a single-stage detection algorithm designed for faster predictions, which directly identifies bounding boxes and classifies objects in a single pass. The key focus was on improving the detection accuracy of pedestrians in low-visibility conditions using infrared and visual data fusion. The enhanced YoloV6 model was trained using this multi-modal data and refined with techniques like attention mechanisms to focus on relevant areas of the image.

classification task.

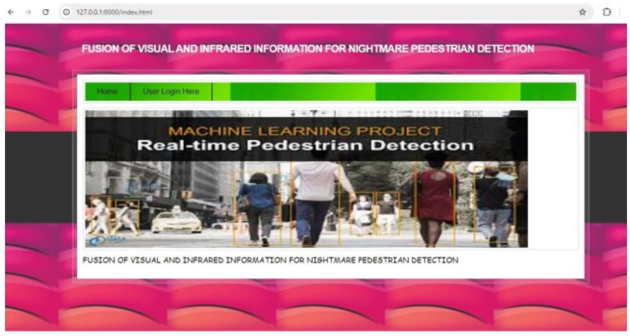

Fig 5: Home page

Figure 6 shows the confusion matrix that represents the model's performance in predicting three classes: Strong, Major, and Great. To evaluate this performance, we calculated metrics like accuracy, precision, recall, and F1-score. These metrics revealed that the model excels at predicting Strong and Great cases.

Fig 6 : login page

This platform is designed to provide users with access to advanced features that enhance pedestrian safety through cutting-edge technology. To continue, please enter your username and password in the fields provided. Your credentials are essential for accessing personalized features and ensuring secure interactions within the system. If you encounter any issues during the login process, please reach out for assistance.

Fig 7: home page after login

V. CONCLUSION

The Pedestrian detection is crucial for autonomous systems, especially in low-visibility conditions like nighttime or poor weather. This research enhances detection by fusing visual (RGB) and infrared (IR) data using YoloV6, a real-time deep learning model. Unlike Faster-RCNN, YoloV6 improves precision, recall, and inference speed (0.07s per image), making it ideal for autonomous vehicles and surveillance. The fusion of IR and visual data reduces false negatives, ensuring better pedestrian localization and safety. This approach significantly improves detection accuracy, enhancing road safety and reliability in challenging environments

.VI REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, et al. “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016: 779-788.

[2] J. Redmon, A. Farhadi. “YOLO9000: better, faster, stronger,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR).,2017: 7263 -7271.

[3] J. Redmon, A. Farhadi. “Yolov3: An incremental improvement,” 2018, arXiv:1804.02767

[4] Bochkovskiy, C.Y. Wang, H. Y. M. Liao. “Yolov4: Optimal speed and accuracy of object detection,”2020, arXiv:2004. 10934.

[5] Cao, Z.; Yang, H.; Zhao, J.; Pan, X.; Zhang, L.; Liu, Z. A new region proposal network for far-infrared pedestrian detection. IEEE Access 2019, 7, 135023–135030.

[6] Park, J.; Chen, J.; Cho, Y.K.; Kang, D.Y.; Son, B.J. CNN-based person detection using infrared images for night-time intrusion warning Systems. Sensors 2020, 20, 34.

[7] He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

[8] He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. 2015, 37, 1904–1916.

[9] Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. arXiv 2018, arXiv:1804.02767(1804).

[10] Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.

[11] Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7794–7803.

[12] Li, X.; Zhao, L.; Wei, L.; Yang, M.H.; Wu, F.; Zhuang, Y.; Ling, H.; Wang, J. Deepsaliency: Multi-task deep neural network model for salient object detection. IEEE Trans. Image Process. 2016, 25, 3919–3930.

[13]K. Lei, Z. Chen, S. Jia, and X. Zhang, "HVDetFusion: A Simple and Robust Camera-Radar Fusion Framework," in *2307.11323*, 2023

[14]K. Shi, S. He, Z. Shi, A. Chen, Z. Xiong, J. Chen, and J. Luo, "Radar and Camera Fusion for Object Detection and Tracking: A Comprehensive Survey,"*2410.19872*, 2024

[15]S. Pang, D. Morris, and H. Radha, "TransCAR: Transformer-based Camera-And-Radar Fusion for 3D Object Detection," *2305.00397*, 2023

[16] S. Yao, R. Guan, X. Huang, Z. Li, X. Sha, Y. Yue, E. G. Lim, H. Seo, K. L. Man, X. Zhu, and Y. Yue, "Radar-Camera Fusion for Object Detection and Semantic Segmentation in Autonomous Driving: A Comprehensive Review," in 2023

.[17] A. Nabati and H. Qi, "RRPN: Radar Region Proposal Network for Object Detection in Autonomous Vehicles," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 3094-3098.

[18]Y. Li, H. Ma, Z. Zhong, and X. Liu, "Radar and Vision Fusion for Object Detection in Autonomous Vehicles," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2356-2362.

[19] C. Wang, H. Lan, and Y. Zhang, "Multi-Modal Sensor Fusion for Object Detection Using Deep Learning," in *Sensors*, vol. 21, no. 3, p. 789, 2021.

[20] M. Danzer, D. Griebel, J. Klappstein, and H. Lichte, "2D Car Detection in Radar Data with PointNets," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 1447-1451.