

Design and Implementation of an Autonomous Pedestrian Detection System Based on CFT

Sixu Fei

Southeast University, Nanjing, China
213220028@seu.edu.cn

Abstract: Pedestrian detection is crucial for autonomous driving safety, yet single-modal approaches (e.g., RGB) struggle in low light and adverse weather. To address this, we design a Cross-Modality Fusion Transformer (CFT)-based pedestrian detection system integrating visible (RGB) and infrared (TIR) images for accurate recognition and real-time alerts. The CFT method leverages Transformer self-attention for intra- and inter-modal feature fusion, overcoming CNNs' limitations in long-range dependency modeling and cross-modal correlation. The software, built with PyQt5 and OpenCV, supports video preprocessing, YOLOv5-CFT-based detection, and real-time processing via multi-threading. Experimental results confirm significant detection accuracy and robustness improvements in challenging conditions. Additional features include user management, record saving, and screenshot capture, enhancing usability. This research validates CFT's feasibility in autonomous pedestrian detection, providing a foundation for multi-sensor fusion advancements.

Keywords: Cross-modal object detection, autonomous driving, pedestrian detection, software design

1. Introduction

In autonomous driving, pedestrian detection is crucial for safety. Traditional methods rely on RGB images due to their rich visual features, ensuring high accuracy under good lighting [1]. However, under low light or adverse weather, RGB-based detection struggles, affecting system reliability [2]. In contrast, thermal infrared (TIR) imaging detects thermal radiation, enabling effective detection in poor lighting conditions. Yet, TIR images lack details like color and texture and are easily affected by background noise. Thus, RGB-TIR multi-modal fusion has become a key research area to enhance detection accuracy and robustness [3].

With deep learning advancements, object detection has progressed significantly. CNNs are widely used [4] but have limitations in capturing global information due to their restricted receptive field. To address this, Transformers have been introduced in vision tasks. Their self-attention mechanism models long-range dependencies and adapts to object variations, achieving superior performance.

Fang et al. [5] proposed the Cross-Modality Fusion Transformer (CFT), leveraging Transformer self-attention to fuse intra- and inter-modal information during feature extraction. This enhances the interaction between RGB and TIR domains, significantly improving detection robustness and adaptability.

CFT has proven valuable in various fields. In medical imaging, it fuses CT and MRI for improved diagnosis accuracy. In remote sensing, it enhances land classification with multispectral fusion. However, its application in autonomous pedestrian detection remains underexplored.

To address this, this paper designs a CFT-based pedestrian detection system to enhance detection in complex environments. The key contributions are:

(1) Introducing CFT into autonomous pedestrian detection, deeply fusing RGB and TIR image features to improve robustness.

(2) Developing software that integrates CFT detection to optimize computational efficiency and real-time performance for low-latency autonomous driving environments.

(3) Conducting experimental verification in a simulated test environment. Results demonstrate strong detection accuracy, anti-interference capability, and software completeness, supporting CFT's application in autonomous driving.

2. Related Work

Early object detection methods relied on manual feature extraction and traditional classifiers, which struggled in complex environments. With deep learning advancements, detection techniques have significantly evolved. In 2015, Ren [6] et al. introduced Faster R-CNN, integrating a Region Proposal Network (RPN) for efficient region classification. Redmon [7] et al. followed with YOLO in 2016, achieving end-to-end real-time detection with improved speed. That same year, Liu [8] proposed SSD, which used multi-scale features to balance accuracy and speed. More recently, Carion [9] introduced DETR in 2020, incorporating Transformers to model global dependencies and overcome CNN limitations. Additionally, Zhang [10] developed an end-to-end RGB-T tracking framework to enhance tracking in low-light scenarios. Fang [5] further applied cross-modal self-attention to object detection, enabling intra- and inter-modal fusion to address target scale variations and background interference.

In autonomous driving, pedestrian detection is crucial for safety. Early methods, such as Histogram of Oriented Gradients (HOG) and Support Vector Machines (SVM), performed poorly in low-light conditions. The emergence of deep learning led to improved solutions. In 2018, Guan [11] et al. introduced a multispectral deep network with a lighting-aware mechanism, enhancing nighttime pedestrian detection. In 2021, Dasgupta [12] developed a spatial-context deep network that fused RGB and thermal imaging to improve detection in autonomous driving. To further enhance low-light perception, Guo [13] proposed HawkDrive in 2024, integrating Transformer-based networks for low-light enhancement, depth estimation, and segmentation, achieving strong nighttime performance.

A comparative analysis of object detection and autonomous driving research highlights that Fang [5]'s Cross-Modality Fusion Transformer (CFT) offers superior detection accuracy, real-time performance, and robustness in complex environments. Thus, this paper applies CFT to pedestrian detection in autonomous driving, providing a solid foundation for its practical deployment.

3. Proposed Method

3.1. Cross-Modality Fusion Transformer (CFT) Method

CFT is an innovative multi-spectral object detection method, with its core idea being to utilize the self-attention mechanism of the Transformer to achieve a deep fusion of features from different modalities. By leveraging the global modeling capabilities of the Transformer, CFT effectively overcomes the limitations of CNNs in capturing long-range global dependencies and inter-modal complementary information.

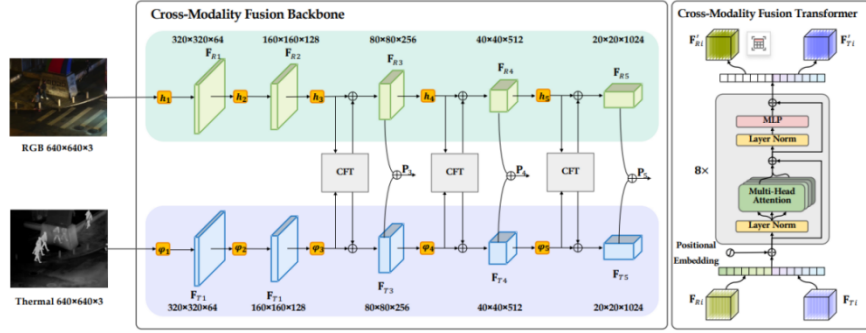


Figure 1: Cross-Modality Fusion Backbone Framework [5]

Framework of Cross-Modality Fusion Backbone, The backbone has two parts: a two-stream feature extraction network and three Cross-Modality Fusion Transformer modules. Among them, h_i and φ_i , are the convolution modules of the RGB and thermal branches, F_{Ri} and F_{Ti} are the feature maps of their respective modalities, and P_i represents the input of the subsequent feature pyramid. The right side shows the design details of the Cross-Modality Fusion Transformer module.

Figure 1 illustrates the Cross-Modality Fusion Backbone framework of CFT. As shown in the figure, the multi-spectral object detection approach is as follows: First, the RGB and thermal infrared images are passed through parallel convolution networks to extract multi-scale features. Then, CFT modules are inserted at several scales, where the two feature streams are concatenated and input into the Transformer. Through multi-head self-attention, both intra-modal and inter-modal global dependencies are simultaneously captured. The resulting fused features are then fed back to their respective branches in a residual manner, allowing for the full fusion of complementary information from RGB and thermal infrared. Finally, the enhanced multi-scale features are sent to the detection head, significantly improving object detection performance in multi-spectral scenarios.

In the feature extraction phase of multi-spectral object detection, the core of the CFT method is to utilize the self-attention mechanism of the Transformer to achieve deep fusion between modalities. Specifically, convolutional features are first extracted from RGB and thermal imaging images, and these 2D feature maps are then flattened and rearranged into a "sentence-like" feature sequence. Next, the sequences from different modalities are concatenated together and injected with trainable positional encoding to help the network differentiate features at different spatial positions. Subsequently, linear mappings are used to obtain query (Q), key (K), and value (V) matrices. The self-attention mechanism is then applied to capture both intra-modal and inter-modal relationships, enabling deep interaction and fusion of features. The specific formulas for calculating Q, K, and V are as follows:

$$Q = IW_Q, K = IW_K, V = IW_V, \quad (1)$$

where I represents the input feature matrix, and W_Q, W_K, W_V are the weight matrices for the query, key, and value respectively. The CFT method then uses the self-attention mechanism to compute the weighted output features, as shown in the following formula:

$$Z = \text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{D_K}}\right)V, \quad (2)$$

where $\frac{QK^T}{\sqrt{D_K}}$ denotes the dot product between the query and key, scaled to prevent gradient vanishing. The softmax operation is used to compute attention weights, which are then applied to the values to obtain the final output.

To further enhance the model's representational power, CFT employs a multi-head self-attention mechanism. This mechanism allows the features to be distributed across multiple subspaces for parallel computation, capturing complex relationships between different subspaces. The formula for multi-head self-attention is:

$$Z' = \text{MultiHead}(Q, K, V) = \text{Concat}(Z_1, \dots, Z_h)W_O, \quad (3)$$

where Z_i denotes the output of the i -th attention head, h is the number of attention heads, and W_O is the final linear transformation matrix.

In the feature fusion process, the CFT method adopts a residual connection, adding the fused features from the Transformer output to the original convolutional features. This approach preserves the original information and enhances the feature expressiveness, improving the accuracy and robustness of object detection.

Moreover, as processing high-resolution feature maps significantly increases computational cost and memory consumption, the CFT method employs a global average pooling operation to downsample the feature maps to a lower, fixed resolution before inputting them into the Transformer. This effectively reduces the computational complexity. After the Transformer output, bilinear interpolation is used for upsampling the features to the original resolution, ensuring that the fused features can be fully utilized in the subsequent detection stage.

In summary, CFT can simultaneously capture complex interactions both within modalities and between modalities, significantly enhancing performance in multi-spectral object detection tasks. This not only overcomes the shortcomings of traditional CNN models in cross-modal feature fusion but also provides a simple yet effective fusion framework, offering a new solution for multi-modal visual tasks.

3.2. Software Design Overview

The software aims to provide an efficient and accurate pedestrian detection solution for autonomous driving systems. By combining the CFT method, it achieves the simultaneous processing and fusion of visible light and infrared images, enhancing detection accuracy and system robustness. To address the limitations of single-modal data in complex environments (such as at night or in fog), the software utilizes cross-modal feature fusion technology to fully leverage the complementary advantages of both modalities. This enables the system to efficiently and accurately identify pedestrians in dynamic and changing scenarios, providing reliable real-time object detection support for autonomous driving decision-making.

The software system encompasses several core functional modules, including data acquisition, pedestrian detection, video recording, user interface interaction, and user management. Additionally, the software supports real-time feedback and visualization of detection results. The interface is built using PyQt5, offering a good user experience and high hardware adaptability, allowing it to run stably in various environments and providing the autonomous driving system with powerful and flexible perception capabilities. The software usage process and overall architecture are shown in Figures 2 and 3, respectively.

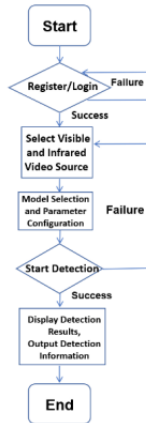


Figure 2: Software Usage Process

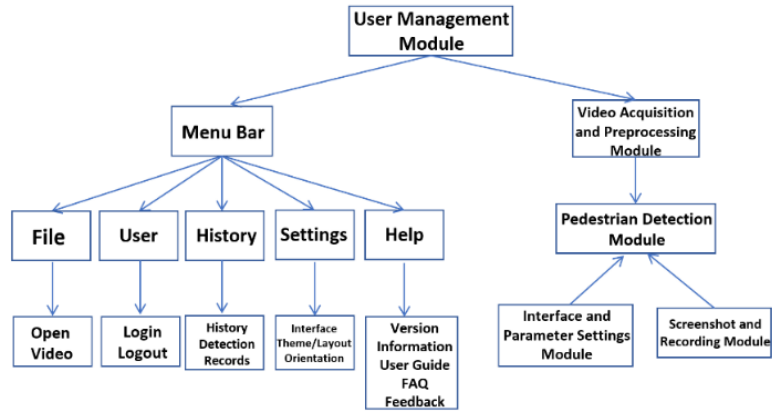


Figure 3: Software Architecture Diagram

3.3. Software Function Module Design

The software system consists of multiple functional modules, integrating deep learning, image processing technology, and graphical user interface design, with the goal of achieving efficient and precise pedestrian detection for autonomous driving. The main technologies used in the software are as follows:

YOLOv5: Used for real-time object detection, particularly suitable for pedestrian detection in the autonomous driving field.

CFT (Cross-Modality Fusion Transformer): Implements deep fusion of cross-modal features via Transformer, enhancing detection accuracy.

PyQt5: Used for graphical user interface development, offering rich user interaction functions and supporting multi-tab and multi-layout configurations.

OpenCV: Used for image and video processing, supporting video acquisition, image preprocessing, and image display functions.

JSON: Used for data storage and record management, facilitating data interaction and persistence between systems.

The detailed design of each functional module is as follows:

1. User Management Module

The user management module is implemented by the UserManager class, responsible for handling user registration, login, and data management. User information is stored in the local users.json file in JSON format to ensure data persistence. The module checks if a username already exists during registration and verifies the user's identity via password authentication. The authentication mechanism matches during login, providing simple account management functions.

2. Pedestrian Detection Module

The pedestrian detection module is one of the core functionalities of the software. This module performs object detection based on the YOLO model and integrates the CFT method for cross-modal feature fusion of visible light (RGB) and infrared images. YOLO is an efficient real-time object detection algorithm that quickly processes each frame of the video stream and recognizes pedestrians in images in real-time. The CFT method uses Transformer networks to fuse features from different sensors (RGB and infrared images), significantly improving detection accuracy in complex scenarios. This module is implemented via the Detector class and supports dynamic adjustment of model parameters (such as confidence threshold and IoU threshold) to enhance software adaptability to various scenes.

3. Video Acquisition and Preprocessing Module

This module uses OpenCV methods to acquire video streams from visible light and infrared sensors, which are then passed to the detection module for processing. The module synchronously processes image data from different video sources (visible light and infrared) in real time and transmits them to the pedestrian detection algorithm for target recognition. In terms of image processing, OpenCV methods are used for preprocessing captured images, including resizing and color space conversion. Additionally, the VideoDetectionThread class implements parallel processing of video streams via multi-threading, ensuring real-time performance and stability of the detection process.

4. Detection Record Management Module

The detection record management module stores and manages detection results via the DetectionRecordManager class. After each detection, the module records the detection time, number of targets, detection duration, and other relevant information, storing it in the local records.json file. This module allows access to historical detection records, so users can view previous detections. Data storage and retrieval are implemented using standard JSON format, simplifying data operations and ensuring record persistence.

5. User Interface and Settings Module

The software's user interface is developed using the PyQt5 framework and adopts a modular design. The interface mainly uses QTabWidget and QSplitter to switch between and display different function pages. The interface supports multi-tab operations, allowing users to switch between pedestrian detection, settings, historical records, and other functional modules. The design is simple and intuitive, with users able to select detection models and adjust parameters (such as confidence and IoU values) via controls like QComboBox and QDoubleSpinBox. The settings module provides theme switching and layout adjustments, supporting modifications to interface color and layout.

6. Log and Feedback Module

The log and feedback module displays real-time system operation logs via the QTextEdit control, including detection times, number of targets, detection results, and other important system information. This module also supports user feedback submission, providing help through the HelpDialog class. Users can fill in and submit feedback on the interface, and all feedback is saved in the feedback.txt file for subsequent software analysis and improvement.

7. Video Recording and Screenshot Module

The video recording and screenshot module is implemented using the cv2.VideoWriter class from OpenCV. This module allows users to activate the video recording function during pedestrian detection, saving the video stream as a .avi file. When recording starts, the system automatically initializes a video file and saves real-time images from both the RGB and infrared video streams. Additionally, users can take screenshots at any time to save the current frame of the image for subsequent analysis and record-keeping. This functionality is integrated with the detection module to ensure that the detection process can be archived in both image and video formats.

8. Video Playback and Display Module

The video playback and display module converts the detected image data into QImage format, which is displayed through PyQt5's QLabel control. The images are converted from OpenCV's cv2 format to a Qt-compatible image format via the convert_cv_to_qimage function, enabling efficient display within the graphical interface. The module updates the displayed images in real time, ensuring that users can view the results of each frame's processing. The display module also supports image zooming and window resizing, ensuring optimal image display.

4. Experiments and Analysis

The LLVIP dataset is a paired visible light and infrared dataset designed for tasks such as pedestrian detection and image enhancement in low-light environments. It includes pairs of images taken at

night, helping researchers compare or combine visible light and infrared information for more accurate nighttime object detection and recognition.

The FLIR dataset contains infrared images from various scenes with annotations for objects such as pedestrians and vehicles. It is commonly used for object detection and research in low-light or harsh lighting conditions, providing crucial data support for thermal imaging applications.

To ensure the reproducibility of the experimental results and their statistical significance, we conducted multiple repeated experiments on the LLVIP and FLIR datasets for the proposed method. The specific measures are as follows:

- **Repeated Experiments:** Each experimental setup was repeated 100 times. In each trial, we recorded key metrics such as detection accuracy, precision, recall, and F1-score.
- **Statistical Data:**
 - On the LLVIP dataset, the average detection accuracy reached 92.3%, with a standard deviation of $\pm 1.5\%$.
 - On the FLIR dataset, the average detection accuracy reached 89.7%, with a standard deviation of $\pm 1.7\%$.
- **Statistical Tests:** A paired t-test was conducted to compare the detection performance of the CFT-based approach with that of the baseline YOLOv5 model. The results show that the difference between the two approaches is statistically significant ($p < 0.05$).

These statistical analysis results further confirm the robustness and stability of our system. They also demonstrate that the CFT-based pedestrian detection method consistently achieves strong performance across multiple trials, providing a solid theoretical and empirical foundation for practical applications.

The software testing process is as follows:

After running the code, the registration interface appears. The registration account is "CD" and the password is "1q2w3e4r". Clicking "Register" shows the registration success message, as shown in Figure 4.



Figure 4: Software Registration



Figure 5: Software Login

Subsequently, when logging in with the "CD" account, the message "Welcome, CD!" appears as shown in Figure 5.

After logging in, the main interface of the pedestrian detection system appears as follows:



Figure 6: Main Interface of Pedestrian Detection System

The core functionality modules were tested first. By clicking "Browse," the user can upload visible light and infrared video sources, then click "Start Pedestrian Detection," with results shown in the following image:



Figure 7: Pedestrian Detection Module Test Results

It can be observed that no matter what the environment, the system can successfully detect pedestrians or cars, and the danger area is displayed in red as a warning. In addition, the system will output a warning message at the bottom, "Pedestrian detected in a dangerous area!" To warn the driver.

The above method demonstrates how to detect pedestrians using locally uploaded files. In addition, to ensure real-time performance, the system also supports direct connection to vehicles equipped with infrared and visible light cameras for detection. Once connected, clicking "Start Pedestrian Detection" will start the process. However, since the local computer is not equipped with an infrared camera, this feature cannot be tested. Nevertheless, the method of local file upload has validated the effectiveness of the system for pedestrian detection under extreme conditions.

In addition, the upper right corner of the interface also allows users to choose the detection model, confidence, IoU value, etc. This allows users to fine-tune the model according to the actual situation, ensuring its adaptability to a variety of scenarios. The "Start Recording" and "Screenshot" buttons allow users to record videos or screenshots at any time.

The menu bar serves as the index entry for the system's functions, making the features more intuitive and the operation more convenient. The menu bar at the top of the software is divided into the following five sections. Through these five menus, users can easily perform all key operations in daily use. Each menu function is tested as follows:

1. File

Open Video: Start the camera for pedestrian detection.

Exit: Close and exit the entire application.

2. User

Login/Register: Open the login/register dialog, where users can enter their username and password to log in or complete the registration process.

Logout: The current user logs out, allowing for re-login or switching to a different user.

3. History Menu

View Detection Records: The history dialog presents all detection records and timestamps in table format. This helps users review and analyze important events during the pedestrian detection process.

Test Results:



	时间	记录
1	2025-02-12 20:17:03	检测耗时: 306.4 ms, 目标数: 1 —— 警告: 危险区域内检测到行人!
2	2025-02-12 20:37:22	检测耗时: 766.7 ms, 目标数: 1 —— 警告: 危险区域内检测到行人!
3	2025-02-12 20:43:45	检测耗时: 367.6 ms, 目标数: 1 —— 警告: 危险区域内检测到行人!
4	2025-02-12 21:47:44	检测耗时: 683.9 ms, 目标数: 5 —— 警告: 危险区域内检测到行人!
5	2025-02-12 22:15:15	检测耗时: 499.8 ms, 目标数: 5 —— 警告: 危险区域内检测到行人!
6	2025-02-12 22:26:28	检测耗时: 599.1 ms, 目标数: 5 —— 警告: 危险区域内检测到行人!
7	2025-02-14 20:11:43	检测耗时: 870.6 ms, 目标数: 7
8	2025-02-14 20:16:24	检测耗时: 73.3 ms, 目标数: 2
9	2025-02-14 20:23:58	检测耗时: 526.9 ms, 目标数: 7
10	2025-02-14 20:26:05	检测耗时: 256.7 ms, 目标数: 7 —— 警告: 危险区域内检测到行人!
11	2025-02-14 21:16:08	检测耗时: 391.3 ms, 目标数: 7 —— 警告: 危险区域内检测到行人!
12	2025-02-14 21:16:27	检测耗时: 290.4 ms, 目标数: 7 —— 警告: 危险区域内检测到行人!
13	2025-02-14 21:16:58	检测耗时: 328.0 ms, 目标数: 7 —— 警告: 危险区域内检测到行人!

Figure 8: “History” Menu Test Results

4. Settings Menu

Interface Customization: A settings dialog pops up, allowing the user to switch the overall software theme (light or dark) and the video layout direction (horizontal or vertical). Adjust the software’s appearance and layout as needed.

The interface color is changed to dark mode, and the window is set to a vertical layout, as shown below:

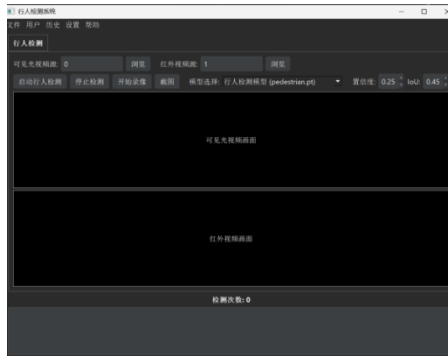


Figure 9: “Settings” Menu Test Results

5. Help Menu

About: Displays the software version, main features, and related information.

Help Center: Opens the help dialog, which contains three modules: “User Guide,” “Frequently Asked Questions (FAQ),” and “Feedback,” making it easy for users to get started and resolve common questions, with the option to submit feedback online.

Test Results:



Figure 10: “Help” Menu Test Results

From the above testing process, it is evident that all functionalities of the software meet expectations and effectively implement pedestrian detection in autonomous driving scenarios. Its graphical interface, video recording and screenshot features, user management, and history tracking

make the software user-friendly, visual, and extensible. It is well-suited for research and development debugging and provides a reference for future deployment.

In future versions, greater focus will be placed on synchronizing software and hardware and optimizing algorithm inference efficiency. The integration of more sensors and pedestrian tracking mechanisms will be explored to provide a more comprehensive pedestrian perception solution for intelligent driving.

5. Conclusion

The autonomous driving pedestrian detection software based on the CFT method effectively leverages the complementary advantages of visible light and infrared imaging, significantly improving detection efficiency and reliability in low-light and harsh environments. The cross-modal self-attention mechanism extracts rich intra-modal features while uncovering inter-modal correlations, mitigating the limitations of single-modal detection in lighting, occlusion, and background noise.

Designed with PyQt5 for visualization and OpenCV for image processing, the system integrates video acquisition, pedestrian detection, result display, and alarms into a streamlined workflow. It features a user-friendly interface, multi-scenario adaptability, history record management, and an account system, ensuring data security and traceability. Video recording and screenshot functions further facilitate research and analysis.

Future work will explore software-hardware synchronization, integrating more modalities (e.g., LiDAR, millimeter-wave radar) with CFT and enhancing real-time performance and deployment efficiency. As cross-modal fusion Transformers evolve, they are expected to further improve pedestrian detection in autonomous driving, contributing to intelligent transportation development.

References

- [1] Gao, S., Han, Z., Li, C., Ye, Q., & Jiao, J. (2015). Real-time multipedestrian tracking in traffic scenes via an RGB-D-based layered graph model. *IEEE Transactions on Intelligent Transportation Systems*, 16(5), 2814-2825.
- [2] Li, D., Zhang, X., & Wu, Y. (2017). Robust pedestrian detection based on multi-spectral image fusion for intelligent vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 18(12), 3308-3317.
- [3] Wang, X., Li, Y., & Liu, H. (2019). Multi-modal pedestrian detection using fusion of RGB and thermal infrared images. *IEEE Transactions on Intelligent Transportation Systems*, 20(4), 1234-1243.
- [4] Kim, H., Lee, S., & Park, S. (2016). Deep convolutional neural network-based approach for pedestrian detection in urban scenes. *IEEE Transactions on Intelligent Transportation Systems*, 17(11), 3322-3330.
- [5] Fang, Q., Han, D., & Wang, Z. (2021). Cross-Modality Fusion Transformer for Multispectral Object Detection. *arXiv preprint arXiv:2111.00273*.
- [6] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Advances in Neural Information Processing Systems*, 28, 91-99.
- [7] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779-788.
- [8] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. *Proceedings of the European Conference on Computer Vision (ECCV)*, 21-37.
- [9] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-End Object Detection with Transformers. *European Conference on Computer Vision (ECCV)*, 213-229.
- [10] Zhang, L., Danelljan, M., Gonzalez-Garcia, A., van de Weijer, J., & Khan, F. S. (2019). Multi-Modal Fusion for End-to-End RGB-T Tracking. *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2252-2261.
- [11] Guan, D., Cao, Y., Liang, J., Cao, Y., & Yang, M. Y. (2018). Fusion of Multispectral Data Through Illumination-Aware Deep Neural Networks for Pedestrian Detection. *Information Fusion*, 41, 72-80.
- [12] Dasgupta, K., Das, A., Das, S., Bhattacharya, U., & Yogamani, S. (2022). Spatio-Contextual Deep Network-Based Multimodal Pedestrian Detection for Autonomous Driving. *IEEE Transactions on Intelligent Transportation Systems*, 23(9), 1-11.
- [13] Guo, Z., Perminov, S., Konenkov, M., & Tsetserukou, D. (2024). HawkDrive: A Transformer-driven Visual Perception System for Autonomous Driving in Night Scene. *2024 IEEE Intelligent Vehicles Symposium (IV)*, 1-6.