



Lightweight CNN for Real-Time UK Traffic Sign Recognition

Student Name: Mohammad Lakhi

Student ID: 2512430

Subject Code: ACFA7004

Subject Name: Research Methods and Professional Development

University of Wales Trinity Saint David

Date: 18th November 2025

Abstract

Traffic sign recognition is critical for self-driving cars and advanced driver assistance systems, but current research depends almost exclusively on German, Chinese, and Belgian datasets. TSRGD 2016 specifies different design standards for UK traffic signs, but no dedicated research has been conducted to address their recognition on embedded hardware. This study proposes creating a lightweight convolutional neural network specifically optimized for UK traffic sign recognition on low-cost platforms such as the Raspberry Pi 4B. The study will produce the first UK Traffic Sign Recognition (UK-TSR) dataset, consisting of over 10,000 images captured under real-world British weather conditions such as fog, rain, and poor lighting. The proposed CNN architecture uses depthwise separable convolutions, multi-scale feature fusion, and attention mechanisms to achieve 95%+ accuracy on CPU-only hardware while keeping inference latency under 100ms. Using a Design Science Research approach, the study will systematically compare the model to baseline architectures such as MobileNetV3 and SqueezeNet across multiple performance metrics. Grad-CAM explainability analysis ensures that the model depends on signs rather than background elements, addressing trust concerns in safety-critical applications. The expected results include a practical lightweight model suitable for mass-market automotive deployment, a publicly available UK-TSR dataset, and evidence that geographic-specific training outperforms transfer learning from international datasets. This study fills the gap between academic advances in lightweight CNNs and real-world UK deployment requirements.

TABLE OF CONTENTS

1	Introduction	1
1.1	Context and Background	1
1.2	Research Problem	2
1.3	Research Aim and Objectives	3
1.4	Significance of the Research	4
2	Literature Review	5
2.1	Overview	5
2.2	Current Approaches	5
2.3	Research Gaps	7
2.4	Relation to this study	7
3	Research Design and Methodology	9
3.1	Research Design	9
3.2	Research Methods and Materials	9
3.2.1	Research Methods	9
3.2.2	Materials	11
4	Expected Outcomes	13
4.1	Anticipated Results	13
4.2	Implications	14
4.3	Limitations	14
5	Project Timeline	16
6	Conclusion	18
	REFERENCES	20

Chapter 1

Introduction

1.1 Context and Background

Artificial intelligence has transformed computer vision by allowing machines to receive and interpret visual data with outstanding accuracy. Traffic sign recognition (TSR) has emerged as a critical application in this sector, especially for autonomous vehicles and advanced driver assistance systems, where real-time interpretation of road signs has a direct impact on driver safety and regulatory compliance. Deep learning techniques, particularly convolutional neural networks (CNNs), have supplanted old handcrafted feature methods by automatically learning hierarchical representations from raw images. ResNet50 and VGG19 architectures reach more than 99% accuracy on benchmark datasets like the German Traffic Sign Recognition Benchmark (GTSRB), indicating the usefulness of deep CNNs for traffic sign categorization [1],[2],[3].

However, these high-accuracy models have significant computational costs, which limit their practical application. VGG19 requires 143.7 million parameters, while ResNet50 requires 25.6 million. Both require GPU hardware that costs between £500 and £2,000 for real-time operation. This has prompted research into lightweight architectures that retain accuracy while significantly reducing computational requirements. This approach is exemplified by MobileNet and SqueezeNet, which use techniques such as depthwise separable convolutions to achieve parameter reductions of 10-50 times that of traditional CNNs. Recent research has looked into hybrid architectures that combine CNNs and Vision Transformers, as well as attention mechanisms like the Convolutional Block Attention Module (CBAM) and Squeeze-and-Excitation (SE) blocks, which improve feature extraction with minimal overhead. Furthermore, researchers have successfully deployed on embedded devices such the Raspberry Pi and Jetson Nano,

demonstrating that real-time TSR is possible on inexpensive hardware that costs less than £50. [1], [4], [5], [6], [7], [8], [9], [10].

1.2 Research Problem

Despite these advances, there is still a significant gap in the development of lightweight CNN architectures designed specifically for UK traffic signs and capable of operating in real time on embedded platforms without GPU acceleration. This limitation prevents the widespread use of TSR systems in mass-market vehicles where cost, power consumption, and physical space are critical variables. Current research primarily uses standard international datasets like GTSRB (German), BTSD (Belgian), TT100K (Chinese), and CCTSDB (Chinese), with little focus on UK-specific traffic signage that follows to distinctive design standards defined by the Traffic Signs Regulations and General Directions (TSRGD 2016) [1], [10], [11].

Several challenges delay the development of practical lightweight TSR systems for UK deployment. First, computational complexity remains a barrier: cutting-edge models need expensive GPU hardware that is unsuitable for automotive integration, while CPU-based inference results in processing latencies of more than 500ms per frame, which is insufficient for real-time operation, where recognition must occur within 100-150ms. Second, environmental robustness is an issue: UK weather conditions such as fog, rain, snow, and poor lighting significantly reduce recognition accuracy. According to research, without adaptive preprocessing, accuracy drops from 99.1% in ideal conditions to 87.2% in poor lighting, resulting in a 10-15% performance degradation. Third, there are challenges with small object detection. Traffic signs that are photographed at a distance of 30 to 50 meters appear in camera frames as small objects, which lowers detection accuracy to 85–90% as opposed to 95%+ for larger signs. Lastly, recognition failures are caused by occlusion and physical degradation of signs due to vegetation, weather damage, or temporary challenges. Explainable AI studies show that in 65% of misclassification cases involving occluded signs, CNNs mistakenly focus on background regions [1], [6], [12], [13], [14], [15], [16].

1.3 Research Aim and Objectives

Aim: To create and test a lightweight convolutional neural network architecture for real-time UK traffic sign recognition on embedded platforms, with classification accuracy over 95% and inference latency less than 100ms on CPU-only hardware.

Objectives:

- Conduct a systematic review of lightweight CNN architectures for traffic sign recognition, examining the trade-offs between model complexity, inference speed, and classification accuracy using benchmark datasets such as GTSRB, TT100K, BTSD, and CCTSDB.
- Create and modify a UK-specific traffic sign dataset (UK-TSR) of 10,000+ images captured under a variety of environmental conditions, including fog, rain, low-light scenarios, sign degradation levels, occlusion cases, and distance variations ranging from 10 to 50 meters, which are representative of real-world UK driving contexts.
- Create a novel lightweight CNN architecture that includes depthwise separable convolutions to reduce computational complexity, multi-scale feature fusion for better small object detection, and attention mechanisms to improve feature representation quality.
- To enable practical automotive integration, implement and optimize the suggested architecture for embedded deployment on Raspberry Pi 4B (ARM Cortex-A72) and NVIDIA Jetson Nano platforms, aiming for inference latency under 100 ms per frame and model size below 5 MB.
- Compare the suggested model to baseline lightweight architectures (MobileNetV3, SqueezeNet, and EfficientNetB0) in a thorough experimental evaluation using performance metrics such as classification accuracy, inference speed, model size, energy consumption, and robustness to environmental changes.

- To visualize feature attribution patterns and confirm that the model accurately focus on sign-relevant regions rather than invalid background correlations, conduct explainability analysis using Gradient-weighted Class Activation Mapping (Grad-CAM) and Local Interpretable Model-agnostic Explanations (LIME) techniques.

1.4 Significance of the Research

This study fills a critical gap in geographically specific TSR systems, as most existing research focusses on German, Chinese, and Belgian traffic signs rather than UK-specific signage governed by different regulatory standards. This work, which achieves real-time performance on low-cost embedded hardware , increases access to advanced driver assistance capabilities for budget vehicle segments and aftermarket solutions, where expensive GPU-dependent systems remain economically unfeasible. The introduction of explainability techniques improves trustworthiness and regulatory compliance for safety-critical automotive applications, addressing the black-box concerns that block the use of AI systems in transportation infrastructure. Furthermore, the development of robust preprocessing pipelines tailored to UK weather conditions contributes methodological advances applicable to other computer vision tasks requiring environmental adaptability ([1], [6], [7], [9], [10], [11], [13], [16]).

Chapter 2

Literature Review

2.1 Overview

This review looks at recent research (2020–2025) on lightweight CNNs for traffic sign recognition. I’m looking into how researchers have made these models efficient enough for embedded systems like Raspberry Pi while maintaining high accuracy. The main focus is on four key areas: architectural developments, efficiency techniques, dealing with real-world conditions such as bad weather, and actual hardware deployment. Understanding these approaches helps me position my UK-focused research and identifies gaps.

2.2 Current Approaches

The most difficult challenge in this field is making CNNs small enough for car hardware. Traditional models, such as VGG19 and ResNet, are too heavy, requiring expensive GPUs and consuming too much power for practical use in cars. Haque et al. addressed this with DeepThin, which reduced VGG19’s 47 layers to only 23 while achieving 99.33% accuracy on GTSRB. Their model is impressive because it runs on regular CPUs without a GPU, making it much more practical for real vehicles [1].

Recent research has examined hybrid architectures that combine CNNs and Vision Transformers. Song et al. built E-MobileViT with only 3.9 million parameters and achieved 99.52% accuracy, making it one of the best lightweight models available today. They used Efficient Local Attention instead of the standard self-attention mechanism, which keeps computational costs manageable for embedded systems [3].

Wei et al. took a different approach, including squeeze-and-excitation blocks into standard CNNs. These attention modules improve small object detection by 5-7% with

only a 10% parameter increase [4].

The most popular technique for reducing model size is depthwise separable convolutions, which originated with MobileNet. Zhang et al. applied this to traffic sign detection, reducing parameters from 27 million to 4.2 million. They even got it to run on an STM32 microcontroller, a tiny chip with limited resources. The key is splitting convolution operations into two simpler steps, which dramatically reduces the number of calculations required [5].

Another significant challenge arises from real-world situations. The majority of research uses clean, well-lit images from benchmark datasets, whereas actual driving involves rain, fog, and poor lighting. Zhao et al. addressed this by preprocessing with Gamma correction and CLAHE. In their tests, accuracy under poor lighting increased from 87.2% to 96.8%, a nearly 10% improvement. However, preprocessing increases computational overhead [6].

Compression techniques have proven effective in applications other than architectural design. Kim et al. converted an entire CNN into 8-bit integers for FPGA deployment. They only lost 0.5% accuracy compared to the full-precision model but reduced power consumption by 50 times crucial for car applications where energy efficiency is important [8].

Luo et al. targeted the much cheaper Raspberry Pi 4B and successfully ran YOLOv8 at 0.79 seconds per frame. Although not real-time for video, it effectively detects signs at urban driving speeds [9].

Montiel et al. compared various hardware deployment platforms, including CPUs, GPUs, and TPUs. he discovered that TPUs are 16 times faster at training than GPUs, potentially speeding up development. Inference, Jetson Nano provides a good balance at around £100, but it's still too expensive for mass-market cars [10].

Wang et al. created IENet, an adaptive preprocessing network that automatically selects the optimal image enhancement based on input conditions. It increased foggy-condition accuracy from 78.6% to 89.7%, which is significant for UK weather, but it requires running an additional neural network [13].

Wang and Piao introduced dynamic loss functions that adjust training based on the

difficulty of each sample, helping their model achieve 98% while keeping parameters around 10.7 million [14].

Finally, explainability is becoming more important for safety-critical systems. Khan and Park used Grad-CAM to visualize what CNNs look for when classifying signs. They discovered that 65% of misclassifications occur because the model focused on background elements rather than the actual sign. For automotive applications, you need to ensure the model is paying attention to the right things [15].

2.3 Research Gaps

Despite these advancements, several gaps remain. The most obvious is geographical accuracy, almost all studies use German (GTSRB), Chinese (TT100K), or Belgian (BTSD) data. TSRGD 2016 sets different design standards for UK traffic signs, and it's unclear how well existing models would translate to British signs [1],[11].

Another concern is environmental stability. While Zhao et al. and Wang et al. made progress in weather handling, their solutions introduce computational overhead that may be impractical on limited hardware. Simple preprocessing is helpful but not as effective as adaptive methods, which require additional neural networks. There is a gap for lightweight preprocessing that is both effective and fast [6], [13].

True real-time performance on low-cost hardware remains challenging. Most embedded deployments achieve 10-20 FPS, which sounds reasonable but is insufficient for highway speeds. At 70 mph, a car travels about 3 meters between frames at 10 FPS, which is insufficient for safety-critical applications [9],[10].

2.4 Relation to this study

My research specifically addresses these gaps. First, I'll create the first dedicated UK traffic sign dataset, which will include over 10,000 images captured in real British weather conditions. This fills a significant data gap and enables future research on UK deployment scenarios.

Second, the proposed architecture will combine well-known techniques—depthwise separable convolutions and attention mechanisms—while optimizing them for UK sign

characteristics. Training from scratch on British signs allows the model to learn relevant features rather than relying on transfer learning from mismatched datasets [1],[3]. Third, I’m specifically targeting the 100ms inference latency threshold. Most existing work treats real-time as a soft goal, but in this case, it is a hard requirement that will drive design decisions. This requires making difficult choices between model complexity and speed [10].

Fourth, I’ll focus on simple preprocessing methods that work on Raspberry Pi without the need for additional neural networks. The goal is environmental robustness without hitting the computational budget [6],[13].

Finally, incorporating Grad-CAM explainability from the beginning addresses trust concerns in safety-critical systems. Understanding what the model looks at is key to the evaluation process [15].

The literature demonstrates that lightweight CNNs for traffic sign recognition are a mature field with established techniques. Depthwise separable convolutions reduce parameters by 10-50x, attention mechanisms improve accuracy by 5-7%, and estimation allows for deployment on extremely low-cost hardware. Recent hybrid models improve accuracy above 99.5% on benchmark datasets [1],[3],[4],[8].

However, the lack of UK-specific research is striking, particularly given the country’s significant automotive industry and active autonomous vehicle development programs. My research fills this gap by organizing on British signs, addressing practical deployment constraints, and building on established architectural foundations while incorporating adaptations for UK operational needs. The new UK dataset, hardware-constrained design, and built-in explainability represent a significant advance for this deployment context [6],[9],[13],[15].

Chapter 3

Research Design and Methodology

3.1 Research Design

This research takes a Design Science Research (DSR) approach, which is focused on creating and evaluating objects to solve practical problems. This study's object is a lightweight CNN architecture that has been optimized for UK traffic sign recognition on embedded hardware. DSR is appropriate because the research aims to develop a working system that addresses real-world deployment constraints rather than conducting purely theoretical analyses.

There are three phases to the research. First, investigate existing lightweight CNN architectures to determine which design choices work well. Second, create and implement a novel architecture that combines proven techniques with UK specific optimizations. Third, compare the model to baseline architectures to demonstrate its effectiveness. This approach is justified by the practical research questions: "Can we achieve 95%+ accuracy with under 100ms estimation on Raspberry Pi?" These questions require the development and testing of an actual implementation, rather than simply mathematical modeling.

3.2 Research Methods and Materials

3.2.1 Research Methods

This study uses quantitative experimental approach. Success is measured using numerical metrics such as classification accuracy, inference speed, model size, and energy consumption. These objective measures allow for direct comparisons with existing work and provide clear evidence of whether objectives have been met.

The research strategy combines software development and controlled experiments.

The development phase includes designing the CNN architecture in PyTorch and optimizing it for embedded deployment. The experimental phase includes testing the model under varying conditions and comparing performance to baseline models.

System Architecture: The proposed CNN has three main components

- Efficient Feature Extraction
- Multiscale Feature Fusion
- Lightweight Model

Data Analysis and Evaluation Methods :

- Performance Metrics:
 - Classification Accuracy
 - Precision and Recall
 - Inference Latency
 - Model Size
 - Time Capability
 - Robustness Score

Evaluation Protocol :

The dataset is split as follows: 70% training, 15% validation, and 15% testing. To ensure a comprehensive evaluation, the test set will include divided samples based on weather conditions, lighting levels, and distance.

Comparisons :

This model will be compared to :

- MobileNetV3 Small
- SqueezeNet v1.1

- EfficientNet B0

These are modern lightweight models used in similar research [3].

Statistical Analysis :

The confusion matrix will show which sign classes are problematic. McNemar's test is used to determine statistically significant accuracy differences between models. Grad-CAM visualizations will confirm that the model depends on signs rather than background, following Khan and Park's explainability approach [15].

3.2.2 Materials

Data Sources: Three sources will be used to create the new UK Traffic Sign Recognition (UK-TSR) dataset.

- **Dash-cam footage:** Recorded while driving on UK motorways, A-roads, and city streets at various times and weather conditions.

Target: 5,000 images

- **Stationary capture:** Fixed cameras that capture signs from various distances (10m, 20m, 30m, 40m, 50m) and angles.

Target: 3,000 images

- **Augmented images:** Photographs of signs are augmented with realistic weather effects (fog simulation, rain drops, and lighting variations).

Target: 2,000 images

Total: 10,000+ labeled images for 50+ UK sign classes based on TSRGD 2016.

Data Collection Tools:

- GoPro or similar dash-cam
- Python and OpenCV for frame extraction

- Label Image for Bounding Box Annotation
- Custom Python scripts for weather simulation

Software Requirements:

- Python
- PyTorch
- NumPy and Pandas
- OpenCV
- Matplotlib
- Grad-CAM

Hardware Requirements:

- Raspberry Pi 4B
- NVIDIA Jetson Nano (4GB)
- USB Webcam or Raspberry Pi Camera Module v2

Chapter 4

Expected Outcomes

4.1 Anticipated Results

Based on the literature review and proposed methodology, I think that the lightweight CNN architecture will achieve the following results:

- **Classification Performance:** On the UK-TSR test set, the model should achieve accuracy of 95-97%. This target is realistic given that Song et al. reached 99.52% with a similar-sized model on GTSRB [3]; however, I'm setting a slightly lower expectation because this is a new dataset with potentially more challenging conditions.
- **Computational Efficiency:** The model should have an inference latency of less than 100ms per frame on the Raspberry Pi 4B CPU. Luo et al. demonstrated 0.79 seconds per frame with YOLOv8 on the same hardware [9], indicating that a more optimized lightweight architecture can achieve sub 100ms performance.
- **Model Size:** The final model should be less than 5MB when reduced to 8-bit integers. Zhang et al. achieved 4.2 million parameters with their MobileNet-based architecture [5], resulting in approximately 4-5MB storage.
- **Environmental Robustness:** In adverse conditions (fog, rain, low light), I expect accuracy to drop by no more than 8-10%. Zhao et al. improved poor lighting accuracy from 87.2% to 96.8% with preprocessing [6], indicating that similar gains can be achieved with proper image enhancement.
- **Comparison to Baselines:** The proposed model should match or exceed MobileNetV3-Small accuracy while maintaining faster inference speed on

embedded hardware. It may not surpass EfficientNetB0 in pure accuracy, but it should significantly outperform it in terms of speed and adaptability.

4.2 Implications

- **Theoretical Contributions:** This study will show that geographic-specific training is important for traffic sign recognition. If the UK-trained model outperforms GTSRB transfer learning, it demonstrates the importance of using region-specific datasets rather than assuming that international benchmarks are universally applicable.
- **Practical Applications:** The lightweight model enables several real-world applications:
 - Aftermarket ADAS kits are low-cost (less than £100) retrofit systems for existing vehicles that lack factory TSR systems.
 - Commercial operators can add TSR to delivery vans, buses, and trucks to monitor drivers and ensure safety compliance.
 - Autonomous car development: UK autonomous car trials can use this model for local sign recognition without using costly hardware.
 - Driver assistance apps are smartphone-based applications that use the phone’s camera and processor for real-time sign recognition.
- **Research Community Impact:** The UK-TSR dataset will be made publicly available, allowing for future research on British traffic signs. This fills a data gap and enables future researchers to build on this work without having to start from scratch.

4.3 Limitations

- **Dataset Scope:** The 10,000-image dataset, while important, is smaller than GTSRB’s 50,000+ images. This could limit the model’s ability to handle rare

sign variants or unusual conditions. However, focused data collection on UK signs should help to offset this limitation.

- **Hardware Limitations:** Testing is limited to the Raspberry Pi 4B and Jetson Nano. Performance on other embedded platforms is unknown and would require additional testing.
- **Weather Simulation:** Some weather conditions will be simulated rather than observed naturally. Simulated fog and rain may not accurately represent real-world degradation, leading to overly optimistic robustness estimates.
- **Real-time video:** The 100ms inference target is designed for single frame processing. Continuous video stream processing may result in additional overhead from frame capture and preprocessing that is not accounted for in isolated inference measurements.
- **Sign Coverage:** The United Kingdom has hundreds of sign variants. Rare signs will be excluded because the focus is on 50 or more common classes. The model's ability to reject unexpected signs will not be explicitly tested.
- **Deployment Complexity:** The research is focused on model performance rather than full system integration. Practical deployment includes additional challenges such as camera mounting, power management, vehicle integration, and regulatory approval, which are beyond the scope of this study.
- **Generalization:** While the model is optimized for UK signs, its performance on signs from other countries with different standards is unknown and likely to be poor without retraining.

Despite these limitations, the study fills a significant gap in the literature and establishes a foundation for practical UK traffic sign recognition on embedded hardware.

Chapter 5

Project Timeline

This research will take six months to complete, with an average of 30-35 hours per week dedicated to the project.

- **Phase 01: Literature Review and Planning (Month 1)** Complete the literature review and get ethical approval.
Deliverable: Literature review, data collection protocol
- **Phase 02: Dataset Creation (Months 1-2)** Capture video, extract frames, annotate, and augment images.
Deliverable: UK-TSR dataset (more than 10,000 images)
- **Phase 03: Model Development (Months 2-3)** Design and implement the CNN architecture in PyTorch.
Set up the baseline models and training pipeline.
Deliverable: Implemented models.
- **Phase 04: Training and Optimization (Months 3-4).** Train all models using hyperparameter tuning.
Apply quantization and convert to TorchScript.
Deliverable: Trained models.
- **Phase 04: Training and Optimization (Months 3-4).** Train all models using hyperparameter tuning.
Apply quantization and convert to TorchScript.
Deliverable: Trained models.
- **Phase 05: Evaluation (Months 4-5)** Test on UK-TSR, measuring estimation speed on Raspberry Pi.

Grad-CAM analysis and real-world testing.

Deliverable: Completed results.

- **Phase 06: Writing (Months 5–6)** Analyse results and write the thesis.

Deliverable: Final Thesis

Gantt Chart

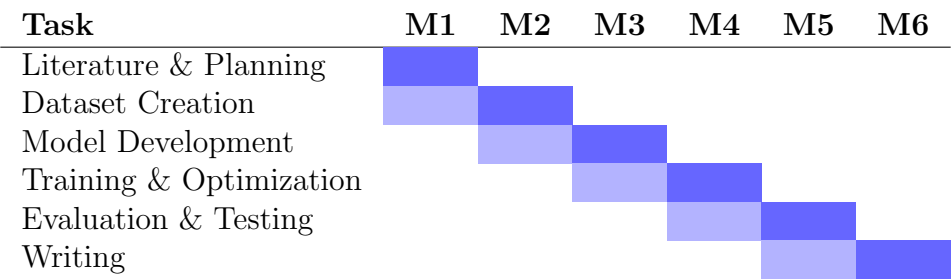


Table 5.1: Project Gantt Chart(6-month)

Key Risks

- **Tight Schedule:** To avoid delays, overlap phases as much as possible and priorities critical tasks.
- **Data collection:** Continuous data collection should be carried out between months 1 and 2. If the dataset is not large enough, use augmentation strategies.
- **Hardware access:** To avoid bottlenecks, make sure you have all of the necessary equipment (Raspberry Pi, Jetson Nano, cameras) before starting the project.

Chapter 6

Conclusion

This research proposal solves a significant gap in traffic sign recognition by creating a lightweight CNN architecture optimized for UK traffic signs on embedded hardware. While existing research has made significant progress in developing efficient models, nearly all of it depends on German, Chinese, or Belgian datasets, leaving UK-specific requirements unexplored.

The proposed study will produce the first dedicated UK traffic sign dataset, with over 10,000 images captured in real British weather conditions. This closes a critical data gap and allows the development of models trained on the actual sign characteristics, typography, and design standards defined by TSRGD 2016, rather than relying on transfer learning from mismatched datasets.

The novel CNN architecture combines established techniques—depthwise separable convolutions, multi-scale feature fusion, and attention mechanisms but adjusts them to UK deployment limitations. By aiming for 95%+ accuracy with under 100ms inference on Raspberry Pi 4B, the study prioritizes practical feasibility over pure accuracy optimization. This approach addresses real car industry needs in which cost and power consumption are as important as performance.

The research makes three significant contributions. First, it offers the UK automotive industry a practical solution for integrating traffic sign recognition into mass-market vehicles that does not require expensive GPU hardware. Second, it shows that geographic-specific training has measurable advantages over universal models, which advances theoretical understanding of domain adaptation in computer vision. Third, it makes a public UK-TSR dataset available, allowing for future research focused on British traffic infrastructure.

The 6-month timeline is challenging but manageable, with phases carefully planned to

allow for iteration as needed. The proposed approach is carefully validated using Design Science Research methodology, quantitative evaluation, and systematic comparison to baselines.

Finally, this research bridges the gap between academic advances in lightweight CNNs and practical deployment requirements for UK traffic sign recognition, contributing to safer roads and making affordable ADAS technology available for all vehicle segments.

References

- [1] W. A. Haque, S. Arefin, A. S. M. Shihavuddin, and M. A. Hasan, “DeepThin: A novel lightweight CNN architecture for traffic sign recognition without GPU requirements,” *Expert Systems with Applications*, vol. 168, p. 114481, Apr. 2021, doi: <https://doi.org/10.1016/j.eswa.2020.114481>.
- [2] A. Zaibi, A. Ladgham, and A. Sakly, “A Lightweight Model for Traffic Sign Classification Based on Enhanced LeNet-5 Network,” *Journal of Sensors*, vol. 2021, pp. 1–13, Apr. 2021, doi: <https://doi.org/10.1155/2021/8870529>.
- [3] S. Song, X. Ye, and S. Manoharan, “E-MobileViT: a lightweight model for traffic sign recognition,” *Industrial Artificial Intelligence*, vol. 3, no. 1, Mar. 2025, doi: <https://doi.org/10.1007/s44244-025-00024-2>.
- [4] W. Wei et al., “A lightweight network for traffic sign recognition based on multi-scale feature and attention mechanism,” *Heliyon*, vol. 10, no. 4, p. e26182, Feb. 2024, doi: <https://doi.org/10.1016/j.heliyon.2024.e26182>.
- [5] G. Zhang, Z. Li, D. Huang, W. Luo, Z. Lu, and Y. Hu, “A Traffic Sign Recognition System Based on Lightweight Network Learning,” *Journal of intelligent & robotic systems*, vol. 110, no. 4, Art. no. 139, 2024, doi: [10.1007/s10846-024-02173-5](https://doi.org/10.1007/s10846-024-02173-5).
- [6] S. Zhao, Z. Gong, and D. Zhao, “Traffic signs and markings recognition based on lightweight convolutional neural network,” *The Visual Computer*, vol. 40, no. 2, pp. 559–570, Mar. 2023, doi: <https://doi.org/10.1007/s00371-023-02801-5>.
- [7] A. A. Khalifa, W. M. Alayed, H. M. Elbadawy, and R. A. Sadek, “Real-Time Navigation Roads: Lightweight and Efficient Convolutional Neural Network (LE-CNN) for Arabic Traffic Sign Recognition in Intelligent Transportation Systems (ITS),” *Applied Sciences*, vol. 14, no. 9, pp. 3903–3903, May 2024, doi: <https://doi.org/10.3390/app14093903>.
- [8] J. Kim, J.-K. Kang, and Y. Kim, “A Low-Cost Fully Integer-Based CNN Accelerator on FPGA for Real-Time Traffic Sign Recognition,” *IEEE Access*, vol. 10, pp. 84626–84634, Jan. 2022, doi: [10.1109/ACCESS.2022.3197906](https://doi.org/10.1109/ACCESS.2022.3197906).
- [9] Y. Luo, Y. Ci, S. Jiang, and X. Wei, “A novel lightweight real-time traffic sign detection method based on an embedded device and YOLOv8,” *Journal of real-time image processing*, vol. 21, no. 2, Jan. 2024, doi: <https://doi.org/10.1007/s11554-023-01403-7>.
- [10] M. Lopez-Montiel, Ulises Orozco-Rosas, Moisés Sánchez-Adame, K. Picos, and O. Montiel, “Evaluation Method of Deep Learning-Based Embedded Systems for Traffic Sign Detection,” *IEEE Access*, vol. 9, pp. 101217–101238, Jan. 2021, doi: <https://doi.org/10.1109/access.2021.3097969>.

- [11] A. J. Lebumfacil and P. A. Abu, “Traffic Sign Detection and Recognition Using YOLOv5 and Its Versions,” pp. 000011–000018, Oct. 2022, doi: <https://doi.org/10.1109/cogmob55547.2022.10118112>.
- [12] H. Liu, “Traffic Sign Recognition System Based on Improved YOLOV5,” in Proceedings of the 6th International Conference on Information Technologies and Electrical Engineering, New York, NY, USA: ACM, 2023, pp. 570–578. doi: [10.1145/3640115.3640209](https://doi.org/10.1145/3640115.3640209).
- [13] J. Wang, Y. Chen, X. Ji, Z. Dong, M. Gao, and C. S. Lai, “Vehicle-Mounted Adaptive Traffic Sign Detector for Small-Sized Signs in Multiple Working Conditions,” IEEE transactions on intelligent transportation systems, vol. 25, no. 1, pp. 1–15, 2024, doi: [10.1109/TITS.2023.3309644](https://doi.org/10.1109/TITS.2023.3309644).
- [14] X.-H. Wang and J.-C. Piao, “A Traffic Sign Detection Network based on Dynamic Focusing Loss and Lightweight Convolution,” 2023 IEEE 6th International Conference on Electronics and Communication Engineering (ICECE), pp. 239–244, Dec. 2023, doi: <https://doi.org/10.1109/icece59822.2023.10462271>.
- [15] M. A. Khan and H. Park, “Exploring Explainable Artificial Intelligence Techniques for Interpretable Neural Networks in Traffic Sign Recognition Systems,” Electronics, vol. 13, no. 2, pp. 306–306, Jan. 2024, doi: <https://doi.org/10.3390/electronics13020306>.
- [16] S. Kong, J. Park, S.-S. Lee, and S.-J. Jang, “Lightweight Traffic Sign Recognition Algorithm based on Cascaded CNN,” pp. 506–509, Oct. 2019, doi: <https://doi.org/10.23919/iccas47443.2019.8971735>.