IET Intelligent Transport Systems

WILEY

The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

PGR-Net: A Pedestrian Gesture Recognition Model for Effective AV-Human Interactions in Autonomous Vehicles

Alif Rizqullah Mahdi | Mahdi Rezaei | Natasha Merat

University of Leeds, Institute for Transport Studies, Leeds, UK

Correspondence: Mahdi Rezaei (m.rezaei@leeds.ac.uk)

Received: 29 October 2024 | **Revised:** 18 November 2025 | **Accepted:** 7 March 2026

ABSTRACT

Autonomous vehicles (AVs) and advanced driver assistance systems (ADAS) continue to advance, yet effective social coordination with human road users (HRUs) remains a key challenge. This study introduces the PGR-Net model, a spatiotemporal deep learning (DL) approach for pedestrian gesture recognition (PGR) to bridge the gap in AV-pedestrian communication. We created the PGR-Net v1.0 dataset by remapping Jester gesture labels to AV-relevant classes: Stop, Go, and Greeting/Thanking. Furthermore, a No Gesture class is defined via a sequential hand-presence rule. The PGR-Net fuses an R(2+1)D, a three-dimensional convolutional neural network (3D-CNN) architecture, and a spatiotemporal stream with hand-pose landmarks, followed by recurrent neural network (RNN) encoders and self-attention layers to emphasise gesture-relevant frames. On the PGR-Net v1.0 dataset, the PGR-Netv2 achieves 88.29% accuracy and an absolute 12.56% improvement from the baseline R(2+1)D model. Qualitative tests on single images beyond the dataset indicate sensible generalisation and highlight the importance of short spatiotemporal context for PGR. These results suggest that hand-augmented spatiotemporal modelling is a viable path toward a robust and AV-relevant PGR for various traffic scenarios. We discuss current limitations due to the limited availability of PGR-specific datasets and outline directions for broader in-the-wild data and context-aware modelling to improve applicability.

1 | Introduction

The widespread usage of autonomous vehicles (AVs) within intelligent transport systems marks a significant milestone in mobility innovation. Despite rapid advancements in advanced driver assistance systems (ADAS), fundamental limitations remain in perception-reaction timing, behaviour understanding, and planning under uncertainty [1–3]. These limitations become especially visible in socially negotiated traffic scenarios, where road users must interpret human signals and respond appropriately. For example, when a traffic officer gestures for an AV to proceed, as shown in Figure 1, a human driver could easily infer the officer's intent, yet the vehicle stalls and behaves conservatively due to an inability to understand the gesture and its context [4]. Interactions with human road users (HRUs), particularly with pedestrians, therefore demand not only robust perception but also social coordination capabilities that current AV stacks largely lack [5–7].

Pedestrian gestures are inherently spatiotemporal, i.e., they unfold over short intervals and draw meaning from body dynamics and scene context [8, 9]. Deep learning (DL) is a state-of-the-art (SOTA) solution for modelling such temporal visual patterns [10, 11], with representative architectures including R(2+1)D, two-stream networks, and temporal segment networks [12–14], typically benchmarked on general action datasets such as UCF101, HMDB51, and Kinetics [15–17]. Although several gesture-focused datasets exist (e.g., EgoGesture, IPN Hand, and Jester), they were primarily designed for human-computer interaction, egocentric viewpoints, or continuous hand gestures rather than traffic-orientated pedestrian-to-vehicle communication [18–20]. This leaves a gap for pedestrian gesture recognition (PGR)

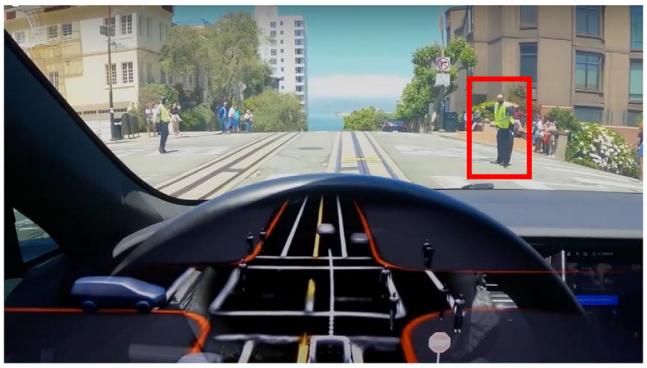

FIGURE 1 | An example of a decision-making conflict that an AV may encounter due to a lack of understanding of the police officer's gesture [4].

specifically tailored to pedestrian-AV exchanges, where identical hand motions may carry different meanings depending on the road context and interacting agent.

This study focuses on front-view interactions from forward-facing vehicle cameras [21] and on upper-body pedestrian gestures most relevant for pedestrian-to-vehicle coordination, while acknowledging that head and lower-body cues also influence the overall gesture meaning and perceived intent [22, 23]. Based on prior taxonomies and gestural motion descriptors, we group gestures into four categories for AV-pedestrian interaction: Stop, Go, Greeting/Thanking, and No Gesture. A more detailed discussion of this grouping is provided in Section 2. These categories cover common pedestrian gestures found near crosswalks and curbside interactions and ground the task in AV-relevant semantics rather than generic actions.

In response to the gap mentioned previously, we introduce the PGR-Net model, a spatiotemporal DL approach designed for PGR in AV-pedestrian interaction. The aim is to reliably classify gestures into the four gesture categories above using short video snippets. This study provides three contributions: first, we construct a PGR dataset by mapping Jester dataset gesture labels to AV-relevant classes (Stop, Go, Greeting/Thanking) and introduce a no-gesture handling strategy based on hand-presence detection; second, we propose a DL architecture that couples an R(2+1)D backbone with hand-pose features, recurrent temporal encoding, and self-attention as a gesture classification network; third, we demonstrate performance improvements over an R(2+1)D baseline on our PGR data and discuss generalisation to images outside the dataset. Together, these steps advance AV-pedestrian social coordination by aligning recognition targets with commonly found pedestrian gestures and by integrating gesture-centric cues for AV decision-making.

2 | Pedestrian Gesture Recognition

2.1 | Gesture Semantics and Task Definition

Pedestrian gestures in road traffic are spatiotemporal signals in which their meaning depends on both body dynamics and situational context. Empirical studies show that a limited set of gestures is commonly understood by human drivers and can measurably influence yielding behaviour [22, 24, 25]. In ambiguous traffic scenarios, gestures play a critical role in solving the right-of-way predicament between drivers and pedestrians [26]. A virtual reality (VR) AV-pedestrian interaction investigation shows that pedestrians use gestures to communicate and negotiate with AVs [27]. There are certain rules or a moral code that determine the meaning of gestures. For instance, waving hands could mean "hello" for human-to-human interaction, and it could mean "please stop" for human-to-bus interaction. Therefore, gesture recognition for AVs should not only determine the gesture type but also give meaning and context to the gesture.

Across different studies and contexts, several pedestrian gestures recur with similarly consistent meanings for coordinating with human drivers. Zhuang and Wu catalogued 11 gestures in Beijing and found that a small subset was widely understood by drivers, with the L-bent-level stop signal significantly increasing yielding rates [24]. Related work on traffic officer signalling identified eight core gestures, several of which mirror pedestrian cues such as Stop and Move Straight [25]. Simulation-based work also converges on a compact gesture set for pedestrian-AV interaction in Unreal Engine scenarios, for example, 'Go Right/Left', 'Stop', 'Come', and 'No Gesture' [28].

Complementing these findings, recent observational and VR studies report that commonly found traffic gestures (e.g., 'Thank You', 'Thumbs Up', 'Wave to Greet/Reject', 'Stop', 'Drive On') observed in VR and real-world settings are largely consistent [26]. Building on these findings, Figure 2 lists the gestures that we have summarised from the literature, which we organised into four classes used throughout this paper: Stop, Go, Greeting/Thanking, and No Gesture. We combined 'Greeting' and 'Thanking' into a single class because they serve the same functional role: to acknowledge drivers that signal recognition or agreement. However, because some gestures are context-dependent, such as raising a hand for greetings versus stop requests for buses/taxis, our present labels are context-invariant. We are only labelling the gestures based on their motion and independent of their background context, due to the Jester only having indoor-only video data.

2.2 | PGR-Relevant Dataset and Label Mapping

A pedestrian gesture is a composition of basic hand/arm motion (e.g., waving, thumbs up, hand extension) that provides communicative meaning in traffic contexts. Several publicly available gesture datasets already contain these motions, most notably the Jester dataset for front-facing hand gestures [20], EgoGesture for egocentric hand sequences [18], and IsoGD for isolated and continuous gestures [29]. While these datasets were not collected for PGR, they provide the raw motion patterns required for deriving PGR-relevant gesture labels. Therefore, we created the PGR-Net v1.0 dataset by remapping source labels to the four gesture classes mentioned in the previous subsection.

In our dataset construction, Jester categories are mapped to three gesture-positive classes: Stop, Go, and Greeting/Thanking, as shown in Table 1. The resulting PGR-Net v1.0 dataset contains short image sequences that emphasise upper-body motion. Each

FIGURE 2 | Six common pedestrian gestures were explored in this research.

TABLE 1 | Jester Label and PGR Label.

Original (Jester) Label	New Label
Stop sign	Stop
Pushing hand away	Stop
Swiping right	Go
Swiping left	Go
Shaking hand	Greeting/Thanking
Thumb up	Greeting/Thanking

sequence comprises 26–30 frames at a fixed resolution of 100 × 176 pixels. Figure 3 illustrates typical examples across classes. For quick ablation and prototyping, we also define the PGR-Net mini v1.0 (~20% of the PGR-Net v1.0 dataset) with the same label distribution. A detailed distribution for both datasets is shown in Table 2.

For the four-class configuration mentioned in the previous subsection, we added a No Gesture class, implemented via a hand-presence check over the sequence. If no hands are detected across frames, the sequence is treated as 'No Gesture' (see Section 3 for more details). This design keeps the PGR-Net v1.0 dataset's three sourced gesture classes intact while enabling a four-class evaluation that distinguishes gesture-positive frames from non-gesture-positive frames.

2.3 | Gesture Recognition

Determining pedestrian gesture from short video sequences is fundamentally a spatiotemporal recognition problem closely related to human action recognition. Among the DL models designed for such cases, the R(2+1)D is a strong and widely used baseline. The model divides 3D convolution into a 2D spatial stage followed by a 1D temporal stage over each frame, improving feature extraction and increasing precision compared to standard 3D kernels [12, 30]. In this work, we adopt R(2+1)D as the primary backbone to extract motion-aware features from video sequences, consistent with prior evidence that convolutional spatiotemporal encoders provide robust performance for short, fine-grained motions.

Purely visual-based encoders can still struggle to separate gesture meanings that share similar motion (e.g., raising the hand for 'Stop' and 'greet'). A recent intention-prediction work has shown that integrating task-specific cues with video features improves classification performance; for example, PIP-Net (a pedestrian

FIGURE 3 | Gesture examples in the PGR dataset.

TABLE 2 | Train and validation distribution.

Class	PGR-Net v1.0 Dataset		PGR-Net mini v1.0 Dataset	
	Train	Validation	Train	Validation
Go	8098	973	1582	159
Greeting/Thanking	8463	1061	1655	177
Stop	8527	1068	1668	178
No Gesture[a]	N/A	N/A	N/A	N/A
Total	**25088**	**3102**	**4905**	**514**

[a]No Gesture is created at the sequence level via a hand-presence rule, detailed in Section 3.2.

FIGURE 4 | Hand pose estimation example from 'Stop' image.

intent prediction model) fuses multiple spatiotemporal features to predict pedestrian crossing intention in the wild [31]. Inspired by this, we complement the R(2+1)D feature stream with hand-centric cues that are used for communication through gestures in traffic scenarios.

For hand cues, we employ MediaPipe to estimate 21 landmarks per hand for each frame, yielding (x, y) coordinates that capture finger and palm configurations over time [32, 33]. The per-frame landmarks are concatenated into a sequence that further emphasises hand dynamics, as shown in Figure 4. In addition to helping enrich gesture discrimination, the hand-pose feature stream provides support in checking for non-gesture-positive frames, which we use for the four-class configuration. This approach enables the model to distinguish gestures from a global context through the spatiotemporal stream and differentiate semantically close gestures through the hand-pose stream. The architecture details and training process for the model are detailed in the next section.

3 | PGR-Net Model

This research presents the PGR-Net model, a novel spatiotemporal architecture tailored to recognise and classify pedestrian gestures from short video sequences, illustrated by Figure 5. The network processes two inputs per sequence: a spatiotemporal stream that captures global motion and context using an R(2+1)D backbone and a hand-pose stream that encodes fine-grained hand dynamics over time. Each stream is processed with recurrent units and refined with Self-Attention to emphasise gesture-critical frames before a final fusion stage to produce the final prediction over the target classes.

3.1 | Spatiotemporal Stream

Sequences are sampled at 12 *fps* from the videos of the PGR-Net v1.0 dataset, and the model processes n frames per sample (with $n = 10$, resulting in ~0.8 s of sampling time). Each image stack is represented as $b \times n \times h \times w \times c$, where b is the batch size, n is the number of frames, h is the height of the image, w is the

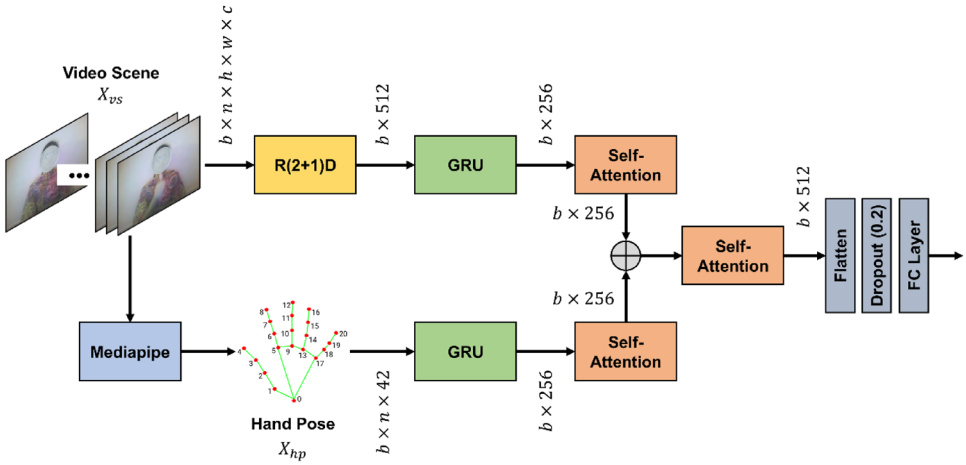

FIGURE 5 | The PGR-Net model architecture.

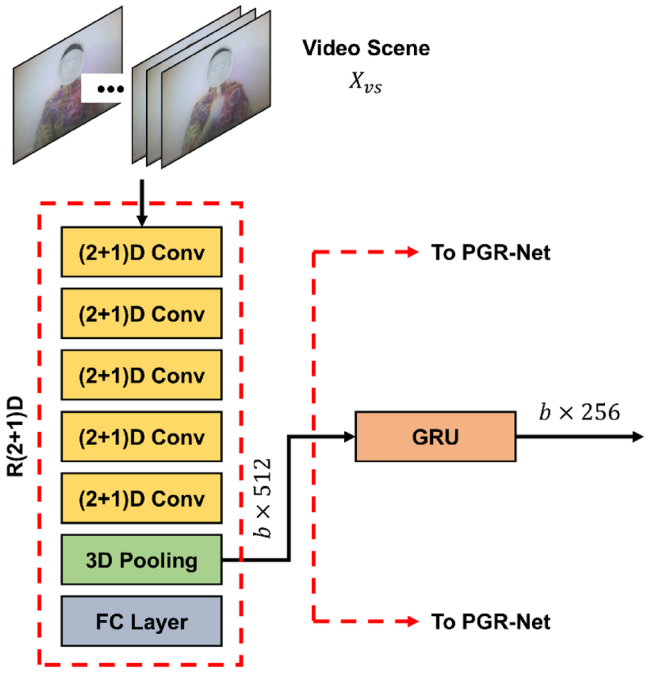

FIGURE 6 | R(2+1)D baseline model used in the PGR-Net model.

width of the image, and c is the number of channels. The image in frame k (im_k) are preprocessed by resizing the frames as $40 \times 40 \times 3$ RGB images for the spatiotemporal encoder. These choices are reflected in Equation (1), where the tensor for each sequence X_k^{im} at frame k is constructed of images from frame $n-k$ until n.

$$X_k^{im} = [im_{n-k}, im_{n-k+1}, \ldots, im_n] \tag{1}$$

The R(2+1)D is utilised as the backbone for the spatiotemporal stream. In our solution, we take the output of the 3D pooling layer (with a size of $b \times 512$ per sequence) and pass it through a gated recurrent unit (GRU) stack with three layers and 256 hidden units. This configuration, illustrated by Figure 6, helps encode spatial and temporal dependencies compactly. Moreover, GRUs offer better accuracy and lower computational load relative to LSTMs in comparable settings [34]. The resulting output from the GRU block is a $(b, 256)$ embedding that will be later combined with the hand-pose stream.

3.2 | Hand-Pose Stream

We complement the spatiotemporal stream with a hand-pose stream, by extracting 21 (x, y) hand landmarks using MediaPipe for each time step [32, 33]. Identical to the spatiotemporal stream, landmarks collected from n frames are vectorised into a 1D vector, denoted by hp, resulting in an $b \times n \times 42$ vector that is fed onto another GRU layer (3 layers, 256 hidden units) to preserve the temporal dynamics. The vector representation for the hand-pose landmark at frame k (X_k^{hp}) is represented below:

$$X_k^{hp} = [hp_{n-k}, hp_{n-k+1}, \ldots, hp_n] \tag{2}$$

Another process derived from the hand-pose feature is constructing a hand-presence rule to be used in determining the fourth class: 'No Gesture'. As shown in Figure 7, each sequence of m image samples are checked for hand-pose presence. This process is similar to the works of Kopuklu et al., where their network contains a detector queue that checks the presence of gestures before classifying them [35].

This procedure, described in detail in Algorithm 1, prevents false positives on non-gesture-positive motions and defines "No Gesture" without altering the three gesture-positive classes.

The 'no' gesture classifier receives the vectors in X_{hp} and the label of the hand sequence y_{hp}. If any of the vectors are non-zero, the algorithm increases the count of detected hands (n_{hands}), which will then determine the class of the sequence. If the number of detected hands is less than one, that is, no hands are detected, the algorithm will relabel y_{hp} to the "No Gesture" class. Conversely, if at least one hand is detected in the sequence ($n_{hands} \geq 1$), the label is set according to the corresponding gesture-positive label from the PGR-Net v1.0 dataset.

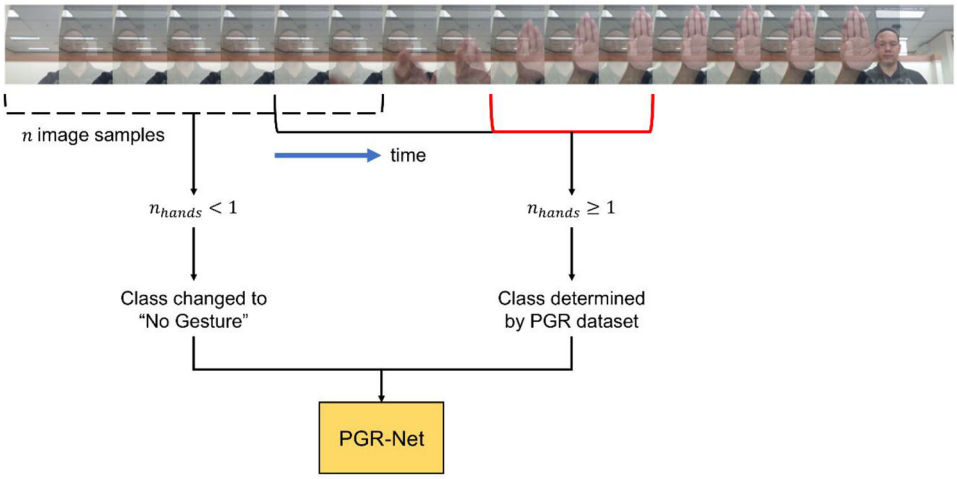

FIGURE 7 | Hand pose extraction.

ALGORITHM 1 | No gesture classifier.

Input: Hand pose sequence and label
Data: X_{hp}, y_{hp}
// y_{hp} = Label of the hand pose sequence
$n_{hands} = 0$
FOR hp **IN** X_{hp} **DO**
  IF hp is NOT zero **THEN**
    $n_{hands} = n_{hands} + 1$
  END
IF $n_{hands} < 1$ **THEN**
  $y_{hp} \leftarrow 3$ // No Gesture
ELSE
  $y_{hp} \leftarrow 0$ *or* 1 *or* 2 // Determined based on PGR Dataset
END
RETURN X_{hp}, y_{hp}

3.3 | Attention and Fusion

The temporally encoded outputs from both streams are then weighted by their relevance using Self-Attention layers. As mentioned in the original paper [36], Self-Attention layers help to capture dependencies in sequential data. This helps the model in capturing the relevance of a particular motion at a certain time step compared to the whole sequence. The weights are calculated by using Equation (3).

$$\text{Attention}\ (Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

where Q is the matrix of queries, K is the matrix of keys, V is the matrix of values, and d_k is the dimensionality of the queries/keys. The resulting outputs from the Self-Attention layers are concatenated to form an $n \times 512$ fused representation, followed by a third Self-Attention block that further discriminates important features that contribute to a specific gesture. The output from this layer is flattened and regularised with a dropout layer (p = 0.2) to reduce overfitting [37]. The result of this process is then passed to a fully connected classifier that outputs three classes (Stop, Go, Greeting/Thanking) for the PGR-Netv1 and four classes (Stop, Go, Greeting/Thanking, No Gesture) for the PGR-Netv2, which is discussed in detail in the next section.

TABLE 3 | Training configuration.

Parameters	Value
Epochs	100
Optimizer	Adam
Learning Rate	0.001
Weight Decay	0.0001
Batch size	24
Sampled frames (m)	10

4 | Experimental Results

We evaluated three model configurations: a baseline R(2+1)D model that uses only the spatiotemporal stream; the PGR-Netv1; which fuses spatiotemporal and hand-pose streams with a three-class output; and the PGR-Netv2; which adds the 'No Gesture' handling and a four-class output. Apart from the output layer and the no-gesture detection functionality, v1 and v2 share the same architecture, temporal encoders, and attention layers. These models are trained with the PGR-Net v1.0 dataset and the PGR-Net mini v1.0 dataset. The configuration for the model training is listed in Table 3. Each run uses the same training and validation split given in Table 2, while a strictly held-out single-image test set is reserved for final investigation.

Initial explorations and ablations use the PGR mini v1.0 to reduce the computation time while probing the model configuration. Full-scale training runs on the PGR-Net v1.0 dataset; the settings remain identical except for the number of epochs, which is set

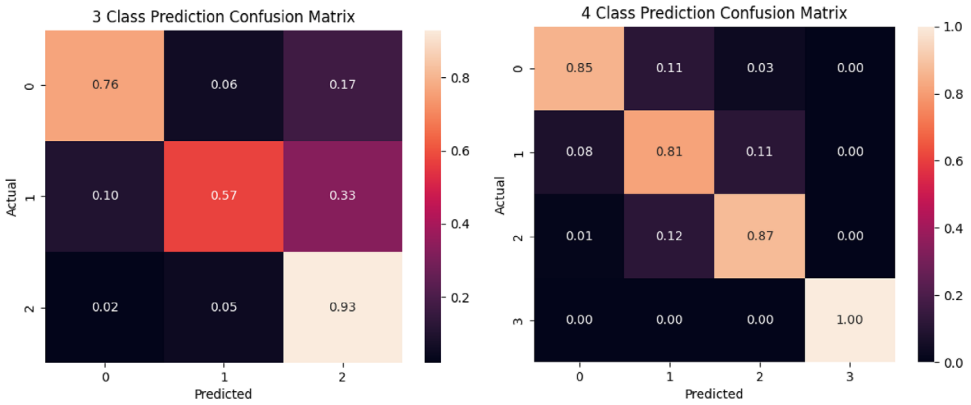

FIGURE 8 | Confusion matrix for PGR-Netv1 (left) and PGR-Netv2 (right).

TABLE 4 | Training result metrics.

Metric	R(2+1)D	PGR-Netv1	PGR-Netv2
Precision	75.92%	78.23%	**89.09%**
Recall	75.80%	74.87%	**88.81%**
F1	0.757	0.746	**0.889**
Accuracy	75.73%	75.87%	**88.29%**

to 30. This change reflects the convergence rate obtained from the mini dataset experiments and aims to limit overfitting and training time on the full-scale dataset.

On the PGR mini v1.0, the PGR-Netv1 achieves a total accuracy of 75.87%, while PGR-Netv2 reaches 88.29%. As shown in Figure 8, the largest relative improvement occurs in class 1 (Greeting/Thanking), improving from 57% to 81%.

This indicates that hand-pose cues help to distinguish the communicative meaning of hand gestures from visually similar motions. However, there is a slight decrease in accuracy on class 2 (Stop) and perfect accuracy for class 3 (No Gesture). This shows that most of the 'No Gesture' class are predicted as 'Stop' when using the three-class configuration.

As shown in Table 4, the PGR-Netv2 outperforms the PGR-Netv1 and the baseline R(2+1)D across precision, recall, F1, and overall accuracy. The R(2+1)D has difficulty in achieving higher accuracy, likely caused by the variations of gesture movements within a single class. The PGR-Netv1 has similar results to the baseline R(2+1)D, meaning that there were not enough improvements by using GRU and attention layers. In contrast, adding the No Gesture class in the PGR-Netv2 improves the overall performance of the model in distinguishing the gestures. This suggests that incorporating hand pose as an additional feature helps the model in recognising hand presence and differentiating gestures with similar motion, leading to better results.

Training the PGR-Netv2 on the PGR-Net v1.0 dataset with 30 epochs maintains performance parity with the PGR mini v1.0, achieving a total accuracy of 88.29% with corresponding precision, recall, and F1 of 88.29%, 88.23%, and 0.883, respectively.

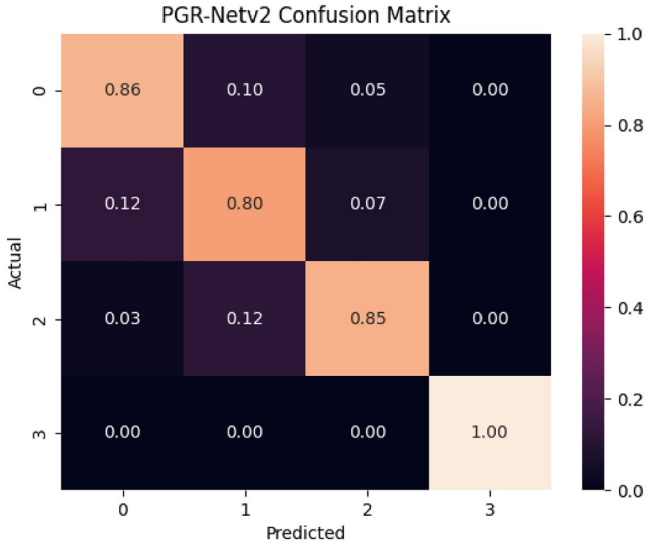

FIGURE 9 | Result of the PGR-Netv2 training with the full PGR dataset.

The confusion matrix structure in Figure 9 mirrors that of the mini dataset experiments, suggesting that the model is able to maintain stable performance as data scale increases.

We further tested the generalisability of the model on single images outside of the PGR dataset, as shown in Figure 10. The model is able to classify images (a) to (e) correctly; for example, images (b) and (d) show a person holding their hands out to show a 'Stop' signal. However, the person in image (f) is raising their hand to greet, and the model inaccurately classifies the gesture as 'Stop'. The misclassification may occur due to the single-frame input of the test images. In contrast, the models are trained using image sequences. This shows the importance of spatiotemporal context, where both spatial and temporal features offer more reliable cues compared to single-frame data. Based on the experimental results, the PGR-Netv2 is quite successful in recognising pedestrian gestures with different backgrounds and settings from the PGR-Net v1.0 dataset.

We measured the model's computational cost on a workstation running Ubuntu 20.04, equipped with an Intel Xeon W-2225 4.10 GHz CPU, an NVIDIA Quadro RTX 5000 GPU with 16 GB

FIGURE 10 | Inference results from external data with real images.

TABLE 5 | The PGR-Netv2 inference performance.

Configuration	Input resolution	Processing time (*ms*)	Effective *fps*
PGR-Netv2	176 × 100 p	21.0	32.4
YOLO-n + PGR-Netv2	1280 × 720 p	35.5	22.3
YOLO-s + PGR-Netv2	1280 × 720 p	37.7	20.2
YOLO-m + PGR-Netv2	1280 × 720 p	41.3	19.1

VRAM, and 62 GB of RAM. We compared the processing time of an input sequence from the PGR-Net v1.0 dataset (100 × 176 pixels, 10 *fps*) against an HD sequence (1280 × 720 pixels, 10 *fps*).

Since the training process uses a detector queue to process gestures, we adopted a similar deployment configuration based on a sliding window with a 10-frame queue. For HD input, we additionally evaluated three variants of an integrated detection–recognition pipeline, where a YOLO-nano, YOLO-small, or YOLO-medium model first detects the pedestrians in the scene and then crops and passes them as the region of interest to the PGR-Netv2 model for gesture recognition.

Table 5 summarises the inference performance of these configurations. The first row represents the processing time on the original PGR-Net v1.0 data set (176 × 100p), and the other rows demonstrate the model performance on the HD resolution input. Each configuration was evaluated on a gesture sequence consisting of 40 consecutive frames, similar to the longest gesture sequences in the PGR-Net v1.0 dataset (~3.3 s at 12 *fps*). For the PGR-Net v1.0 dataset input, the average processing time was 21.0 *ms*, corresponding to an effective rate of 32.4 *fps*, which satisfies both real-time targets of 12 *fps* (corresponding to the PGR-Net v1.0 dataset sampling rate) and the commonly adopted benchmark of 30 *fps* for real-time video processing [38, 39].

In contrast, the HD configurations with YOLO-n, YOLO-s, and YOLO-m yield effective frame rates of 22.3, 20.2, and 19.1 *fps*, respectively. These are sufficient to keep up with a 12 *fps* stream. Since a gesture is a 2–4 s long action, the model's performance on the HD input can also be deemed as real-time. However, going beyond HD resolution or higher frame rates requires further computational resources or model optimisations.

Overall, the PGR-Netv2 consistently outperforms both the R(2+1)D and the PGR-Netv1 on the PGR dataset benchmarks. The results demonstrate that hand-pose spatiotemporal modelling is a viable path for PGR. However, the current data only contains indoor conditions and single-pedestrian, upper-body-centric gestures. This may impact the model's performance on adverse weather conditions, moving pedestrians, and multi-pedestrian scenes with multiple gesture signals. Additionally, the No Gesture detector relies on hand-presence detection, therefore; extreme occlusion or very small-scale images may result in misclassification.

5 | Conclusion and Future Works

This work introduced the PGR-Net model, a spatiotemporal approach that fuses an R(2+1)D spatiotemporal stream with hand-pose dynamics and attention to classify AV-relevant pedestrian gestures. Using the PGR-Net v1.0 dataset, the PGR-Netv2 achieved consistent performance over both the baseline R(2+1)D and the PGR-Netv1 across accuracy, precision, recall, and F1, as shown in Table 4. The PGR-Netv2 achieved a total accuracy score of almost 90% and an F1 score of almost 0.9, signifying that the model has the ability to generally understand and classify pedestrian gestures, shown by the model's sensible results on real images beyond the dataset.

In practical AV perception stacks, the PGR-Net model can operate to detect gestures alongside pedestrian detection and tracking, using front-view clips from existing cameras and producing gesture classification for decision-making modules. Outputs can be consumed by high-level planners (e.g., moving forward when given a 'Go' gesture) and, where appropriate, inform external HMI to give cues and reinforce mutual understanding with pedestrians. Moreover, the model can complement pedestrian intention prediction by introducing gesture and motion recognition as an added feature stream. Our runtime analysis shows that the proposed PGR-Netv2 supports real-time operation at the PGR v1.0 dataset resolution and can meet a 30 *fps* requirement, whereas HD inputs with the integrated YOLO–PGR pipeline are currently limited to around 19–22 *fps*. Practical deployments should therefore balance input resolution and frame rate and consider further optimisation of the detection-recognition pipeline.

Future work may prioritise broader dataset expansion and human-centred labelling to increase robustness and social validity. In particular, assembling a larger in-the-wild PGR dataset with varied environmental conditions will help to improve the model's performance. Complementing the dataset with social perception studies or surveys can validate a taxonomy of socially acceptable pedestrian gestures across different contexts and cultures. Incorporating this taxonomy into the dataset labelling and training objectives may further establish the practicality of the model in various traffic scenarios and move from context-invariant to context-aware labelling.

Author Contributions

Alif Rizqullah Mahdi: conceptualisation, data curation, formal analysis, investigation, methodology, software, validation, visualisation, writing – original draft, writing – review and editing. **Mahdi Rezaei:** conceptualisation, formal analysis, funding acquisition, investigations, methodology, project administration, supervision, validation, funding acquisition, writing – original draft, writing – review and editing. **Natasha Merat:** conceptualisation, supervision, writing – original draft, writing – review and editing.

Acknowledgements

The authors would like to thank all partners within the Hi-Drive project and UK Research and Innovation (UKRI) for their cooperation and valuable contribution. This research has received funding from UKRI through the project references EP/W524372/1 and 2887445, and also the European Union's Horizon 2020 research and innovation programme, under grant Agreement No 101006664. The article reflects only the author's view, and neither the European Commission nor CINEA is responsible for any use that may be made of the information this document contains.

Conflicts of Interest

The authors declare no conflicts of interest

Data Availability Statement

The dataset in this study is available on request from the authors.

References

1. Q. Liu, X. Wang, X. Wu, Y. Glaser, and L. He, "Crash Comparison of Autonomous and Conventional Vehicles Using Pre-crash Scenario Typology," *Accident Analysis and Prevention* 159 (2021): 106281, https://doi.org/10.1016/j.aap.2021.106281.

2. J. Wang, H. Huang, K. Li, and J. Li, "Towards the Unified Principles for Level 5 Autonomous Vehicles," *Engineering* 7, no. 9 (2021): 1313–1325, https://doi.org/10.1016/j.eng.2020.10.018.

3. J. de Winter and D. Dodou, "External human–machine Interfaces: Gimmick or Necessity?," *Transportation Research Interdisciplinary Perspectives* 15 (2022): 100643, https://doi.org/10.1016/j.trip.2022.100643.

4. E. Vinkhuyzen, "What a Self-driving Car Still Can't See," video, 18:10, February 4, 2023, https://www.youtube.com/watch?v=GoTXCvryiW4.

5. S. C. Stanciu, D. W. Eby, L. J. Molnar, R. M. St Louis, N. Zanier, and L. P. Kostyniuk, "Pedestrians/Bicyclists and Autonomous Vehicles: How Will They Communicate?," *Transportation Research Record* 2672, no. 22 (2018): 58–66, https://doi.org/10.1177/0361198118777091.

6. A. Rasouli and J. K. Tsotsos, "Autonomous Vehicles That Interact With Pedestrians: A Survey of Theory and Practice," *IEEE Transactions on Intelligent Transportation Systems* 21, no. 3 (2020): 900–918, https://doi.org/10.1109/TITS.2019.2901817.

7. M. Deepti, K. Mohamed, A. Wael, and A.-S. Mohammed, "Pedestrians' Crossing Behavior at Marked Crosswalks on Channelized Right-Turn Lanes at Intersections," *Procedia Computer Science* 109 (2017): 233–240.

8. S. Kang and B. Tversky, "From Hands to Minds: Gestures Promote Understanding," *Cognitive Research: Principles and Implications* 1, no. 4 (2016).

9. Y. Goutsu, W. Takano, and Y. Nakamura, "Classification of Multi-class Daily Human Motion Using Discriminative Body Parts and Sentence Descriptions," *International Journal of Computer Vision* 126, no. 5 (2018): 495–514, https://doi.org/10.1007/s11263-017-1053-3.

10. H. H. Pham, L. Khoudour, A. Crouzil, P. Zegers, and S. A. Velastin, "Video-based Human Action Recognition Using Deep Learning: A Review," preprint, arXiv:2208.03775, August 7, 2022.

11. S. A. Mahmoudi, O. Amel, S. Stassin, M. Liagre, M. Benkedadra, and M. Mancas, "A Review and Comparative Study of Explainable Deep Learning Models Applied on Action Recognition in Real Time," *Electronics* 12, no. 9 (2023): 2027.

12. M. Huang, H. Qian, Y. Han, and W. Xiang, "R(2+1)D-based Two-stream CNN for Human Activities Recognition in Videos," in *Proceedings of the 40th Chinese Control Conference*, Technical Committee on Control Theory, Chinese Association of Automation, (IEEE, 2021), 7932–7937.

13. L. Wang, Y. Xiong, and Z. Wang, *Temporal Segment Networks: Towards Good Practices for Deep Action Recognition* (Springer International Publishing, 2016).

14. T. Liu, Y. Ma, W. Yang, W. Ji, R. Wang, and P. Jiang, "Spatial-temporal Interaction Learning Based Two-stream Network for Action Recognition," *Information Sciences* 606 (2022): 864–876, https://doi.org/10.1016/j.ins.2022.05.092.

15. K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A Dataset of 101 Human Actions Classes From Videos in the Wild," (CRCV-TR-12-01, University of Central Florida, 2012).

16. H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A Large Video Database for Human Motion Recognition," in *Proceedings of the International Conference on Computer Vision (ICCV)*, (IEEE, 2011), 2556–2563, https://doi.org/10.1109/ICCV.2011.6126543.

17. W. Kay, J. Carreira, K. Simonyan, et al., "The Kinetics Human Action Video Dataset," preprint, arXiv:1705.06950, May 19, 2017.

18. Y. Zhang, C. Cao, J. Cheng, and H. Lu, "EgoGesture: A New Dataset and Benchmark for Egocentric Hand Gesture Recognition," *IEEE Transactions on Multimedia* 20, no. 5 (2018): 1038–1050, https://doi.org/10.1109/TMM.2018.2808769.

19. G. Benitez-Garcia, J. Olivares-Mercado, G. Sanchez-Perez, and K. Yanai, "IPN Hand: A Video Dataset and Benchmark for Real-Time Continuous Hand Gesture Recognition," in *Proceedings of the 2020 25th*

International Conference on Pattern Recognition (ICPR 2020), (IEEE, 2020), 4340–4347.

20. J. Materzynska, G. Berger, I. Bax, and R. Memisevic, "The Jester Dataset: A Large-Scale Video Dataset of Human Gestures," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, (IEEE, 2019), 2874–2882, https://doi.org/10.1109/ICCVW.2019.00349.

21. C. Wang, X. Wang, H. Hu, Y. Liang, and G. Shen, "On the Application of Cameras Used in Autonomous Vehicles," *Archives of Computational Methods in Engineering* 29, no. 6 (2022): 4319–4339, https://doi.org/10.1007/s11831-022-09741-8.

22. A. Almukdad, D. Muley, R. Alfahel, F. Alkadour, R. Ismail, and W. K. M. Alhajyaseen, "Assessment of Different Pedestrian Communication Strategies for Improving Driver Behavior at Marked Crosswalks on Free Channelized Right Turns," *Journal of Safety Research* 84 (2023): 232–242, https://doi.org/10.1016/j.jsr.2022.10.023.

23. M. Lanzer, M. Gieselmann, K. Mühl, and M. Baumann, "Assessing Crossing and Communication Behavior of Pedestrians at Urban Streets," *Transportation Research Part F, Traffic Psychology and Behaviour* 80 (2021): 341–358, https://doi.org/10.1016/j.trf.2021.05.001.

24. X. Zhuang and C. Wu, "Pedestrian Gestures Increase Driver Yielding at Uncontrolled Mid-block Road Crossings," *Accident Analysis and Prevention* 70 (2014): 235–244, https://doi.org/10.1016/j.aap.2013.12.015.

25. Z. Fu, J. Chen, K. Jiang, et al., "Traffic Police 3D Gesture Recognition Based on Spatial–Temporal Fully Adaptive Graph Convolutional Network," *IEEE Transactions on Intelligent Transportation Systems* 24, no. 9 (2023): 9518–9531, https://doi.org/10.1109/TITS.2023.3276345.

26. T. Brand and M. Schmitz, "Identifying Hand Gestures for Pedestrian-driver Communication," *Applied Ergonomics* 125 (2025): 104452, https://doi.org/10.1016/j.apergo.2024.104452.

27. M. R. Epke, L. Kooijman, and J. C. F. Winter, "I See Your Gesture: A VR-Based Study of Bidirectional Communication Between Pedestrians and Automated Vehicles," *Journal of Advanced Transportation* 2021 (2021): 1–10, https://doi.org/10.1155/2021/5573560.

28. E. Shaotran, J. J. Cruz, and V. J. Reddi, "Gesture Learning for Self-Driving Cars," in *Proceedings of the 2021 IEEE International Conference on Autonomous Systems (ICAS)*, (IEEE, 2021), 1–5, https://doi.org/10.1109/ICAS49788.2021.9551186.

29. W. Jun, S. Z. Li, Z. Yibing, Z. Shuai, I. Guyon, and S. Escalera, "ChaLearn Looking at People RGB-D Isolated and Continuous Datasets for Gesture Recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, (IEEE, 2016), 761–769.

30. D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A Closer Look at Spatiotemporal Convolutions for Action Recognition," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (IEEE, 2018), 6450–6459, https://doi.org/10.1109/CVPR.2018.00675.

31. M. Azarmi, M. Rezaei, and H. Wang, "PIP-Net: Pedestrian Intention Prediction in the Wild," *IEEE Transactions on Intelligent Transportation Systems* 26, no. 7 (2025): 9824–9837, https://doi.org/10.1109/TITS.2025.3570794.

32. F. Zhang, V. Bazarevsky, A. Vakunov, et al., "MediaPipe Hands: On-device Real-Time Hand Tracking," preprint, arXiv:abs/2006.10214, June 18, 2020.

33. C. Lugaresi, J. Tang, H. Nash, et al., "MediaPipe: A Framework for Building Perception Pipelines," preprint, arXiv:abs/1906.08172, June 14, 2019.

34. S. Yang, X. Yu, and Y. Zhou, "LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example," in *Proceedings of the 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*, (IEEE, 2020), 98–101, https://doi.org/10.1109/IWECAI50956.2020.00027.

35. O. Kopuklu, A. Gunduz, N. Kose, and G. Rigoll, "Real-time Hand Gesture Detection and Classification Using Convolutional Neural Networks," in *Proceedings of the 14th IEEE International Conference on Automatic Face and Gesture Recognition, (FG 2019)*, (IEEE, 2019), 1–9, https://doi.org/10.1109/FG.2019.8756576.

36. A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention Is All You Need," preprint, arXiv:1706.03762, June 12, 2017.

37. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks From Overfitting," *Journal of Machine Learning Research* 15, no. 1 (2014): 1929–1958.

38. D. G. Lema, R. Usamentiaga, and D. F. García, "Quantitative Comparison and Performance Evaluation of Deep Learning-based Object Detection Models on Edge Computing Devices," *Integration* 95 (2024): 102127, https://doi.org/10.1016/j.vlsi.2023.102127.

39. F. Z. Guerrouj, S. Rodriguez Florez, A. El Ouardi, M. Abouzahir, and M. Ramzi, "Quantized Object Detection for Real-Time Inference on Embedded GPU Architectures," *International Journal of Advanced Computer Science and Applications* 16, no. 5 (2025): 20–29.