Received: 20 February 2021 | Revised: 22 June 2021 | Accepted: 15 July 2021

IET Intelligent Transport Systems

DOI: 10.1049/itr2.12101

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

CyclingNet: Detecting cycling near misses from video streams in complex urban scenes with deep learning

Mohamed R. Ibrahim[1] | **James Haworth**[1] | **Nicola Christie**[2] | **Tao Cheng**[1]

[1] SpaceTimeLab, Department of Civil, Environmental and Geomatic Engineering, University College London (UCL), London, UK

[2] Centre for Transport Studies (CTS), Department of Civil, Environmental and Geomatic Engineering, University College London (UCL), London, UK

Correspondence
Mohamed R. Ibrahim, Gower Street, Chadwick building, Department of Civil, Environmental and Geomatic Engineering, University College London, London WC1E 6BT, UK.
Email: mohamed.ibrahim.17@ucl.ac.uk; mibrahim2006@me.com

Funding information
UCL Overseas Research Scholarship (ORS); the Road Safety Trust, Grant/Award Number: RST 38_03_2017

Abstract

Cycling is a promising sustainable mode for commuting and leisure in cities. However, the perception of cycling as a risky activity reduces its wide expansion as a commuting mode. A novel method called CyclingNet has been introduced here for detecting cycling near misses from video streams generated by a mounted frontal camera on a bike regardless of the camera position, the conditions of the built environment, the visual conditions and without any restrictions on the riding behaviour. CyclingNet is a deep computer vision model based on a convolutional structure embedded with self-attention bidirectional long-short term memory (LSTM) blocks that aim to understand near misses from both sequential images of scenes and their optical flows. The model is trained on scenes of both safe rides and near misses. After 42 hours of training on a single GPU, the model shows high accuracy on the training, testing and validation sets. The model is intended to be used for generating information that can draw significant conclusions regarding cycling behaviour in cities and elsewhere, which could help planners and policy-makers to better understand the requirement of safety measures when designing infrastructure or drawing policies. As for future work, the model can be pipelined with other state-of-the-art classifiers and object detectors simultaneously to understand the causality of near misses based on factors related to interactions of road users, the built and the natural environments.

KEYWORDS
action recognition, computer vision, cycling near misses, deep learning, video streams

1 | INTRODUCTION

Cycling for commuting or leisure is a growing transport mode across the globe. Its benefits to health and the natural environment have driven different policies to promote it and build more infrastructure for cycling in cities [1, 2]. However, the modal share of cycling remains low in comparison to other transport modes in part because it is perceived as a dangerous activity, regardless of its benefits [3]. It has been found that in the UK, for instance, this fear of falling or being in a collision with other road users limits the wide expansion of cycling as a transport mode [4–6]. Unfortunately, in many countries, quantitative analysis of cycling safety is difficult because the low mode share of cycling results in few recorded incidents. To address this data gap, scholars have analysed the occurrence of near misses as a proxy for incidents due to their higher frequency, which some studies estimate is as high as 0.172 incidents per mile [4].

Quantitative data on near misses is usually collected in one of three ways; self-reported surveys or questionnaires, site observation (e.g. at an intersection) and naturalistic studies [7]. Of these, naturalistic studies can provide the richest data through bike-mounted sensors such as video cameras, GPS, range sensors and accelerometers. This type of data is also routinely collected by many cyclists, who use action cameras for safety in the same way car drivers use dashcams, reporting incidents to the police. However, analysis of these data, particularly video scenes, is usually done manually, which is a labour-intensive process that limits the broader applicability of the method. Accordingly, finding a method that could automatically detect near misses and their risk factors from naturalistic cycling data would transform

its applicability from small-scale studies to mass applications to crowdsourced video streams and real-time operation. Such a system would be of significant interest to bike riders, planners and policy-makers alike.

The field of artificial intelligence (AI), specifically, the domain of deep learning and computer vision, has the potential to address this gap [8, 9]. Various models have been developed to recognise a wide spectrum of human actions, activities, or body poses in complex settings from untrimmed video streams. Near misses result from unsafe interaction between a number of road-users or obstacles that cause a risky situation for the person on a bike and subsequently an instant action or a group of actions (e.g. swerving, stopping, turning left or right, etc.) needed to be taken to avoid a crash. Therefore, a computer vision system that can detect this type of interaction will need to not only detect actions but understand how the actions of individuals within a scene interact to produce risk.

In this article, we introduce a new method called CyclingNet to detect cycling near misses from untrimmed video streams in complex urban scenes. The model is trained on video streams of a frontal camera on a moving bicycle, either mounted on the helmet of the cyclist or the bicycle handlebar, without any restrictions towards the camera angle, the physical or the visual conditions of the built environment. Our goal is to provide a fast algorithm that can be deployed on a camera to be used in a real-time or near-real-time setting for detecting and evaluating near misses. Our main contributions are:

- Automating the detection of cycling near misses in near real-time.
- A novel end-to-end deep model for recognising cycling near misses from untrimmed video streams in complex urban settings.
- A human-labelled large-scale dataset for classifying video streams of moving bicycles - at a frame level - of near miss and non-near miss.
- A comprehensive set of experiments to evaluate the different architectures of deep models that can be used as a baseline for future research in this study domain.

At this point, it should be noted that the detection of risk factors associated with near miss events is not the topic of this paper, but the task can also be accomplished using computer vision algorithms (see, e.g. [10, 11]).

After the introduction, the paper is structured into six sections. The second section reviews relevant work on current near misses methodologies and the advances of computer vision in action recognition. In Section 3, we explain our method and the materials used. In Section 4, we present our model results, baseline analysis, and evaluation metrics. Afterwards, in Section 5, we discuss our results in the context of the current literature. We also highlight the state-of-the-art of CyclingNet and its limitations. Last, in Section 6, we conclude and give our remarks for future work.

2 | RELATED WORK

2.1 | Limits of the current methods for analysing cycling near misses

The term cycling near miss is subjective and an individual's perception of an event as a near miss may differ based on their experience level, personal characteristics, and perception of risk. 'Near collision' [12], 'perceived crash risk' [13, 14], 'perceived traffic risk' [15], or 'near miss' [4, 16] are all terms often used to describe and address near misses in the literature. In this research, we use the definition from Ibrahim et al. [7, p. 4], which describes a near miss as 'a situation in which a person on a bike was required to act to avoid a crash, such as braking, speeding, swerving or stopping. In some cases, the definition may be extended to include those events that caused the person on the bike to feel unstable or unsafe, such as a close pass or tailgating'. This broader definition is used because it encompasses the range of near-miss events that have been identified in the literature, including (1) close pass, (2) a near left or right hook, (3) someone pulling in or out, (4) a near-dooring, (5) swerve around an obstruction, (6) pedestrian steps out, (7) someone approaching head-on, or (8) tailgating [17], while also allowing for other unforeseen events that involve action to avoid a crash.

2.2 | Current methods for analysing cycling near misses

In the literature, near misses have been analysed using different types of observational studies, which can be categorised according to Ibrahim et al. [7] as (1) self-report studies using surveys or questionnaires [4, 18]; (2) video analyses at specific sites such as intersections [19, 20], and 3) naturalistic studies where video stream data is collected as people cycle [12, 21, 22]. In general, the naturalistic approach has shown the most progress in analysing road conflicts, near misses, and crashes due to the nature of the data collected [23–25]. In this approach, a group of participants carry out their daily activities using bikes equipped with cameras and sensors. Rich data related to the environment, riding behaviour and interaction with other road users can be collected. However, due to the need to manually label video data, current naturalistic studies are labour intensive and are limited in both transferability from one location to another, and scaling up to cover a wider region or larger number of participants. Thus, it is difficult to draw objective conclusions that can allow either a change in cities' policies or road users' behaviours to provide a safer and more inclusive environment.

In summary, the literature on near misses to date focuses on analysing events that have been manually identified as near misses, whether by self-reporting or manual annotation based on a rider's or analyst's inputs. There is currently no method that automates the detection of cycling near misses from routinely collected data such as video streams. To address this gap, the method introduced in this article aims to automate the detection of cycling near misses from video streams in a naturalistic

study. For further information regarding the different methods and their challenges related to detecting cycling near misses, see [7].

2.3 | Action recognition from video streams

While the issue of detecting cycling near misses from moving bicycles in real-world settings has not been addressed in the literature, there is a well-established body of knowledge on action recognition from video streams. Action recognition using CV typically involves two steps: (1) extracting and encoding features and (2) classifying features into action classes [26]. In recent years, convolutional neural networks (CNNs) have been applied to the task with great success. State-of-the-art models are tested on benchmark datasets, with UCF-101 [27] and HMDB-51 [28] being popular for human motion and action recognition tasks and MPII dataset for human activity and pose recognition [29]. UCF-101 is a dataset of 13,320 YouTube videos broken down into 101 action categories and a further 25 groups. HMDB-51 is similar, with 51 actions in 6849 clips, whereas the MPII dataset comprises over 25,000 images that contain over 40,000 humans with labelled body joints for 410 daily human activities.

Different strategies have been adopted in designing the models' architecture to extract features that could enhance the training and inference of the model. Some models, for instance, rely on spatial features to classify actions [26, 30], whereas others include both temporal and spatial aspects of the scene to classify and localise multiple actions [31–34]. For example, Simonyan and Zisserman [30] introduced an action recognition model relying on a two-stream convolution structure, exploiting both RGB data and optical flow[1]. The model is evaluated on UCF-101 and HMDB-51, achieving a top-performance of 87.9% on the UCF-101 dataset. Wang et al. [36] introduced deep convolutional descriptors based on trajectory pooling. The model is structured and trained as a two-stream convolutional structure in which features are extracted based on the RGB image and the tracked trajectories in the sequential frames. The model achieved the best accuracy of 65.9% and 91.5% on UCF-101 and HMDB-51 datasets respectively. Ng et al. [37] introduced a hybrid model of the convolutional structure and long-short term memory (LSTM) blocks to classify actions based on their temporal structure, achieving a result of 88.6% and 82.6% on the UCF-101 dataset, with and without optical flow data, respectively. Most significantly, Wu et al. [38] introduced a spatiotemporal model relying on the two-streams network, utilising both RGB frames and optical flows. The model architecture is based on integrating LSTM on top of the convolution structures for the two streams. The model achieved a top score of 91.3% on the UCF-101 dataset.

Recently, approaches have been introduced to tackle actions in video streams besides the 2D convolution structure and LSTM units. For example, relying on a 3D convolutional structure (where time is the third dimension), Diba et al. [32] introduced a new temporal 3D CNN model relying on a Temporal Transition Layer to recognise human actions in video streams. This spatiotemporal model aimed to capture the variations in the dynamics of video representation. The model achieved a top-score of 93.2% and 63.5% in UCF-101 and HMDB-51 datasets, respectively. Girdhar and Ramanan [39] introduced a new attentional pooling structure that has improved the accuracy of action recognition on various benchmark datasets without any cost in computation intensity or time of inference. For instance, the model achieved better performance on the MPII dataset than the previous methods with a 12.5% relative overall improvement [29].

There are variants of action recognition models that focus mainly on understanding human activities rather than the overall perception of the interaction between different agents or the clue of the scene in the case of the stated issue of near misses ([26]. In summary, not only do the architecture of action recognition models vary, but also the training process and the data fusion approach. Some models have been trained in an end-to-end network, whereas others are designed and trained in a two-stream network with an early or late-stage fusion of data types (RGB frames, optical flow data, etc.). While complex model structures have yielded higher accuracies in given tasks, specifically, the two-streams network, these differences have consequences on the trade-off between model accuracy, complexity, and time needed for inference. All these factors influence whether the model could function in real-time. Video recognition, however, remains a challenging task due to the high variance between the sequential images and inter-classes and the low-resolution of videos [40].

3 | METHODS

3.1 | Model requirements

As stated in section 2.1, we define a near miss as 'a situation in which a person on a bike was required to act to avoid a crash, such as braking, speeding, swerving or stopping. In some cases, the definition may be extended to include those events that caused the person on the bike to feel unstable or unsafe, such as a close pass or tailgating' [7, p. 4]. To identify these events, a computer vision algorithm must be capable of distinguishing such a set of instant actions from normal riding behaviour, which may also include actions similar to those taken during a near miss.

Near misses can be seen as instant actions that take place by other objects in the scene. Accordingly, there are three main elements that the model needs to learn in order to recognise near misses: (1) The relative motions of the elements in the scene, (2) the spatial structure of the scene, and (3) memory to recognise what happened in the past.

Subjectively, understanding the change in motion could lead to a better way of understanding the actions related to both safe and unsafe rides since each object conserves its motion between consecutive frames and neighbouring pixels are more likely to conserve similar motion. Accordingly, combining street-level

[1] Optical flow refers to the perceived motion of objects in a given scene with respect to the relative motion of the observer and the objects in the scene [35].

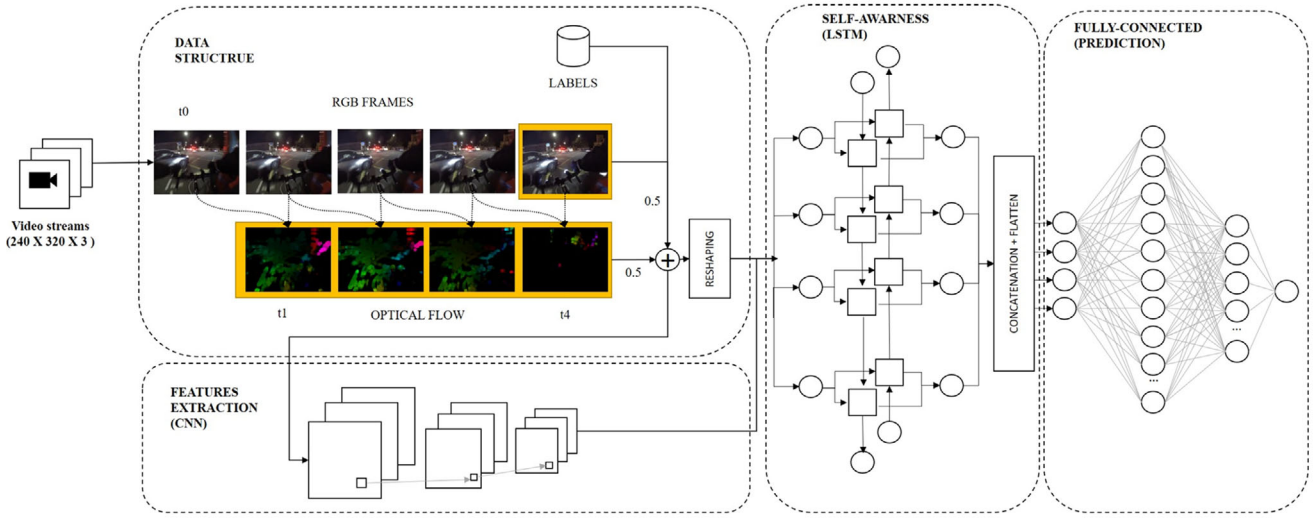

FIGURE 1 The architecture for the proposed CyclingNet

frame images with their optical flow for a number of consecutive frames may lead to a better approach to recognise near misses from video streams.

3.2 | Model architecture

In order to respond to the aforementioned requirements, we propose the CyclingNet model. The CyclingNet is a novel single-stream spatiotemporal deep model that is trained in an end-to-end fashion. It aims to include the features of two-streams networks by including the spatial and temporal aspects of the video stream while providing an inference in near real-time similar to the single-stream networks. Its algorithms comprise four main sections; data structure, feature extraction, self-awareness, and integration and prediction. Figure 1 shows the order of the main algorithms and how the model is structured. It is worth mentioning that the proposed architecture is built incrementally based on trial and error, where a given structure of the architecture has shown enhancement of the overall performance and stability of the trained model. In the upcoming section (Section 4.2), we show a comparison with the outcomes of the different architectures that were built before we reached this architecture and now serve as base models.

3.2.1 | Data structure

As inputs, the model takes two types of data, which are both are resized into 240 × 320 × 3 tensors of single-frame images. The first input comprises video streams typically produced by cameras mounted on a moving bike, which may have varying angles, fields of view, rider speeds and filtering processes (e.g. for stabilisation or extraction of a region of interest). The second input comprises a computed dense optical flow for each pixel in two consecutive frames. It is computed as follows: For a given pixel $P_{(x,y,t)}$ that moves a (d) distance of (dx, dy), the change in P, assuming that P does not change its intensity, can be calculated as:

$$P_{(x,y,t)} = p\,(x + dx,\ y + dy,\ t + dt) \tag{1}$$

By dividing the right side with dt and using the Taylor series approximation, we estimate the optical flow as:

$$f_x u + f_y v + f_t = 0 \tag{2}$$

given that: $f_x = \frac{\partial f}{\partial x}$, $f_y = \frac{\partial f}{\partial y}$, $u = \frac{dx}{dt}$, $v = \frac{dy}{dt}$ where (f_x) and (f_y) are the image gradients, (f_t) is the gradient over time and (u) and (v) are unknown.

In order to solve this equation, with several unknown gradients, we used Gunner Farneback's algorithm [41], in which he approximates each neighbourhood by a quadratic polynomial. Consequently, a new signal can be constructed based on a global displacement, which can be computed based on equating the coefficients of the yields of the quadratic polynomials.

The output optical flow vectors (u, v) are an array of two channels, which can be visualised in a colour image, with a magnitude as the value plane, and direction as the hue value.

After computing the optical flow ($f_{of(t)}$) for a given time (t), the data of the RGB images (f_{rgb}) are truncated for each video file to start with the 4th frame in the frame sequence and wrapped with the four timestamps of the frames of optical flows [t_i, $t_{i\text{-}1}$, $t_{i\text{-}2}$, $t_{i\text{-}3}$] in a proportion of 0.5 to 0.5 respectively. The input ($x_{(t0)}$) is defined as:

$$x_{(t_i)} = \frac{f_{rgb(t_i)}}{2} + \frac{f_{of(t_i)} + f_{of(t_{i-1})} + f_{of(t_{i-2})} + f_{of(t_{i-3})}}{8} \tag{3}$$

There are two reasons for selecting and optimising these hyperparameters: First, to add the time dimension to the spatial structure of each street-level image, and second, to control and

reduce the information and the number of features and textures that are not useful for detecting near misses (i.e. the textures of people, cars, buildings, etc.). We experimented with the values of the combined ratio, based on trial and error to optimise the overall fitness and performance of the model when detecting near misses.

The output data is structured and reshaped into four-dimensional tensors (timestamps, width, height, channels), in addition to embedding the four optical flow steps with the single-frame images. Such an approach means the dimension of time can be utilised and seen either in the spatial structure of the image (fusion with four previous steps of the optical flows) or in the series of the data (the length of timestamps). Both approaches will be utilised and discussed thoroughly in the algorithms of CyclingNet in the two upcoming sections.

3.2.2 | Extracting features

The goal of this part of the model is to extract mainly spatial features from the single-frame images, bearing in mind the fused data of the optical flows of the previous four steps. The architecture of this section comprises three consecutive blocks of convolutional structure, each having different sets of structure and hyperparameters and initialised by the 'He normal' initialisation technique to provide more efficient and faster gradient descent [42]. Generally, the choices of the presented hyperparameters are made based on trials and errors, and the most common practice for training convolutional models. Nevertheless, different models with different hyperparameters will be trained and presented as base models for further evaluating the introduced methods, in the results section (Section 4.2).

Block one consists of two 2D convolution layers of a kernel size of (24 × 5 × 5), (36 × 5 × 5) respectively, and a subsampling size of (2 × 2). They are activated based on a Rectified Linear Unit. These two CNN layers are followed by a 2D Max-Pooling layer of pool size of (2 × 2) and a Batch-normalization layers of the momentum of 0.99 and epsilon of 0.001. It is feed with single-frame images with the embedded optical flow steps.

Similar to block one, block two consists of two 2D Convolution layers, however, a kernel size of (48 × 5 × 5), (64 × 3 × 3) respectively, and a subsampling size of (2 × 2). They are activated based on a Rectified Linear Unit. They are also followed by a 2D Max-Pooling layer of pool size of (2 × 2) and batch-normalization layers of the momentum of 0.99 and epsilon of 0.001.

Block three consists of a single convolution layer of a kernel size (128 × 3 × 3) subsampled with (2 × 2) and activated by a ReLU function. It is also followed by a 2D Max-Pooling layer of pool size of (2 × 2) and a batch-normalization layer of the momentum of 0.99 and epsilon of 0.001.

3.2.3 | Spatial and temporal awareness

If the algorithms for detecting near misses rely only on the features of the previous section (the convolution structure), based on experiments, the results will be sensitive to the changes in the spatial structure of the local context. In other words, the model would not have taken into account the global context of the inputted features that ensure stability and accuracy for training and inference. For this reason, designing the architecture of CyclingNet further to be aware of both local and global spatial and temporal structure is important.

This part of the model comprises one bidirectional LSTM block, followed by a regulated self-attention layer. The LSTM block consists of 128 units, and a dropout regulation of a size of 0.3 to avoid over the fitness of the model. However, the goal is not only to consider the sequence of the defined timestamps but also consider the context for each timestamp. Therefore, a self-attention mechanism is essential to ensure the balance between global and local context when describing a given scene.

Generally, a unidirectional LSTM has shown great progress in extracting features related to sequential data to predict future states [43, 44]. Unlike a traditional recurrent layer, LSTM can learn long-term dependencies without suffering from issues related to vanishing gradient. This internal recurrence, the so-called self-loop, enabled the previous vectors to create paths, in which the gradient can move forward for a long duration without vanishing issues. Nevertheless, most recently, it has also been shown to improve the overall performance of the model when predicting even a given state without timestamps by learning not only the spatial structure of a given vector but also the short-term dependences among the input given vector as the time constants are output by the LSTM itself. Accordingly, this allows the time scale to change based on the input sequence, even if the LSTM units have fixed parameters.

To extract long-term dependences, the self-loops of the LSTM units can be controlled by three gated units: forget gate ($f_i^{(t)}$), external input gate ($g_i^{(t)}$), and an output gate ($q_i^{(t)}$).

First, ($f_i^{(t)}$) can be explained for a given cell (i) and time (t), whereas it is fitted to a scaled value in the interval [0,1] and a sigmoid activation unit (σ) as:

$$f_i^{(t)} = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\right) \tag{4}$$

given that $h^{(t)}$ represents a vector that contains the outputs of all the LSTM cells for the current hidden layer, $x^{(t)}$ represents the current input vector, W^f represents the recurrent weights for the forget gates, U^f represents the input weights and last, b^f represents the biases of the forget gates.

Second, to update the LSTM internal state, a conditioned weight of the self-loop ($f_i^{(t)}$) is computed as:

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma\left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}\right) \tag{5}$$

given that U is the input weights, b is the bias vector, W represents the current weights into the LSTM cell. Similar, to ($f_i^{(t)}$),

the external input gate ($g_i^{(t)}$) is computed, however with it is a parameter:

$$g_i^{(t)} = \sigma\left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\right) \quad (6)$$

Last, the output gate ($q_i^{(t)}$) is used to control and shut off the LSTM cell output ($h_i^{(t)}$) with a sigmoid unit, in which the $h_i^{(t)}$ is defined as:

$$h_i^{(t)} = \tanh(s_i^{(t)}) q_i^{(t)} \quad (7)$$

$$q_i^{(t)} = \sigma\left(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}\right) \quad (8)$$

where b^o is the model biases, U^o is the input weights, $W^o$is the current weight.

Unlike unidirectional LSTM units, a bidirectional LSTM layer allows the current hidden state to rely on two independent hidden states, one computed in a forward direction, named a forward LSTM, and the latter in the opposite direction, named a backward LSTM. This allows the retention of historical and current information simultaneously. This has a direct implication when detecting near misses, in which the predicted output for a given state is smoothed when compared to the previous ones without any post-prediction smoothing techniques.

Moreover, adding a self-attention mechanism to the bidirectional LSTM units allows the model to learn not only from the extracted features – whether spatial or temporal ones- but also to learn from the relations between the input sequences of the RGB image and optical flow ones by allowing the model to relate the position of each sequence and accordingly, learn the representation of its input [43, 45]. Nevertheless, the model can learn which context to consider for a given scene to output the prediction [46]. The context (l_t) can be computed as:

$$l_t = \sum_{t'} a_{t,t'} x_{t'} \quad (9)$$

given that:

$$h_{t,t'} = \tanh(x_t^T W_t + x_{t'}^T W_x + b_t) \quad (10)$$

$$e_{t,t'} = \sigma\left(W_a h_{t,t'} + b_a\right) \quad (11)$$

$$a_t = softmax\,(e_t) \quad (12)$$

where ($h_{t,t'}$) represents the hidden state of the previous step – in a given direction of the bidirectional LSTM- that is fitted to a simple forward neural model ($e_{t,t'}$), (a_t) is the amount of attention that the output at a given state should consider for the previous activation (σ).

TABLE 1 The structure of CyclingNet and its hyperparameters

Block	Layer	Output shape	Number of parameters
Input	$x_{(t_i)}$	(None[b], 240, 320, 3)	0
Block 1	conv2d_1 (Conv2D)	(None, 118, 158, 24)	1824
	conv2d_2 (Conv2D)	(None, 57, 77, 36)	21,636
	max_pooling2d_1 (MaxPooling2)	(None, 28, 38, 36)	0
	batch_normalization_1 (BatchNorm)	(None, 28, 38, 36)	144
Block 2	conv2d_3 (Conv2D)	(None, 12, 17, 48)	43,248
	conv2d_4 (Conv2D)	(None, 10, 15, 64)	27,712
	max_pooling2d_2 (MaxPooling2)	(None, 5, 7, 64)	0
	batch_normalization_2 (BatchNorm)	(None, 5, 7, 64)	256
Block 3	conv2d_5 (Conv2D)	(None, 3, 5, 128)	73,856
	max_pooling2d_3 (MaxPooling2)	(None, 1, 2, 128)	0
	batch_normalization_3 (BatchNorm)	(None, 1, 2, 128)	512
Reshape		(None, 2, 128)	0
Block 4	Attention (SeqSelfAttention-LSTM)	(None, 2, 1024)	10,48,577
	bidirectional_1 (Bidirection-LSTM)	(None, 2, 1024)	26,25,536
	dropout_1 (Dropout)	(None, 2, 1024)	0
Flatten		(None, 2048)	0
Block 5	dense_1 (Dense)	(None, 256)	5,24,544
	dropout_2 (Dropout)	(None, 256)	0
	dense_2 (Dense)	(None, 64)	16,448
	dropout_3 (Dropout)	(None, 64)	0
Output	dense_3 (Dense)	(None, 1)	65

[a]The total trainable parameters: 43,83,902 and the non-trainable parameters: 456.
[b]None values represent the total number of samples in 64 batches in training and validation sets.

3.2.4 | Model initialization and training

After the LSTM block, the output is flattened and fed forward to two fully connected layers of 180 and 64 neurons respectively. Both layers are activated by a ReLU function, in which a dropout mechanism is applied for both layers with a size of 0.3. The final output layers consist of a single neuron and are activated with a sigmoid function.

The model is compiled with stochastic gradient descent, relying on the 'adam' optimiser, with a momentum of 0.9, and an initial learning rate of 0.001. The model is set to be trained for a maximum training cycle (epochs) of 100, with an early stopping technique, monitoring the change in loss with a patience value of 20 epochs.

Table 1 summarises the different layers of the CyclingNet model. It shows the transition of their input shapes and the

number of hyperparameters for each layer. Overall, the model has 4,383,902 trainable parameters.

3.3 | Evaluation metrics

The model is penalised during training, testing and validation based on a cost function of cross-entropy of error. It is defined as:

$$E = -\sum_{i}^{n} t_i \log(y_i) \tag{13}$$

given that t_i represents the target vector, y_i represents the predicted vector, and n represents the binary classes.

For further assessing the model performance, we computed accuracy, precision, recall, false-positive rate, and $F1$-score:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \tag{14}$$

$$Precision = TP / (TP + FP) \tag{15}$$

$$Recall = TP / (TP + FN) \tag{16}$$

$$False\text{-}positive\ rate = FP / (FP + TN) \tag{17}$$

$$F1\text{-}score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{18}$$

where FP represents the predicted false-positive values, TP represents the predicted true-positive values, FN represents the predicted false-negative values, and TN represents the predicted true-negative values.

Last, it remains a challenge to compare our results with other models due to the absence of other models for detecting near misses for a moving cyclist from a street level. We created, however, different architecture to draw a baseline for the performance of the proposed method and to show how the different architectures and hyperparameters could yield different outcomes for a given task with the same material types.

3.4 | Materials and data pre-processing

To the best of our knowledge, there is no benchmark data set of video streams that focus on the different types of cycling near misses that is open-sourced to conduct computer vision research. Therefore, collecting our own dataset becomes the only way to train the model to detect near misses in complex environments. We collected video clips that were made available online by people on bikes on two websites: YouTube, and road.cc. In these clips, near misses are labelled manually in the embedded frames by the sharers. Two aspects make this data a significant one for understanding near misses: First, the variation in the perceptions of near misses as defined by the clips sharers. This could allow the model to extract features related to the common trends instead of being heavily directed or biased by a small group of participants or self-labelling. Second, the variation of equipment, camera position, context, visual, and weather conditions along with the different behaviours and riding styles in these scenes are crucial for the learning process of the model, generalisation, and deployment.

After qualitatively inspecting the quality and ground truth of the embedded information of the selected clips, we collected a dataset of 74,477 sequential frames and we computed their equivalents to optical flow frames (74,469). Of these 8567 sequential frames belong to near miss cases (11.5% of the total sequential frames) which occur at sparse intervals. They represent 209 unique near misses of an average duration of 1.3 s (40.9 sequential frames). We also used an additional dataset of 12,812 sequential frames for further testing, after training and validation. This dataset comprises 81 unique near miss events.

These clips include complex urban settings of different visual and weather conditions and a variety of scene components. For example, 81.9% of the scenes in the dataset are during the daytime, 15.7% at night, and 2.4% at dawn/dusk time. Also, the dataset includes around 93.6% of scenes of clear weather, 5.9% rainy weather, and 0.5 % of snowy weather. Around 2.5% of the dataset includes foggy scenes, 7.9% with glare, and 37.5% are scenes that include a cycling lane. 93.7% of scenes include other humans, 46% include scenes that comprise other cyclists, 67.9 % of scenes include at least one car, 23% with one bus or more, and 12.6% with one truck or more. It is worth mentioning that these data statistics are generated based on visual inspections after using other deep models for extracting labels such as URBAN-i for object detection [10], WeatherNet for weather and visual detections [11].

The clips also consist of variations of near miss types of different temporal scale (the duration of near miss) and various interactions with different road users. The clips, for instance, include near misses such as a close pass, a near left or right hook, someone pulling in or out, swerving around an obstruction, a pedestrian stepping out, and someone approaching head-on. However, there is a lack of clips that include near-dooring and tailgating events. Figure 2 shows a sample of the sequential frames and their corresponding optical flows.

Data augmentation for deep learning has been shown as a strong indicator for enhancing the training process and accuracy of the model [47]. Accordingly, we augmented the collected data by applying several techniques such as normalisation, scaling, and horizontal flipping.

4 | RESULTS

4.1 | CyclingNet evaluation

Figures 3a,b show the losses and accuracies of the training and testing sets respectively for the self-attention bidirectional CNN-LSTM architecture. After 35 training cycles (epochs) of

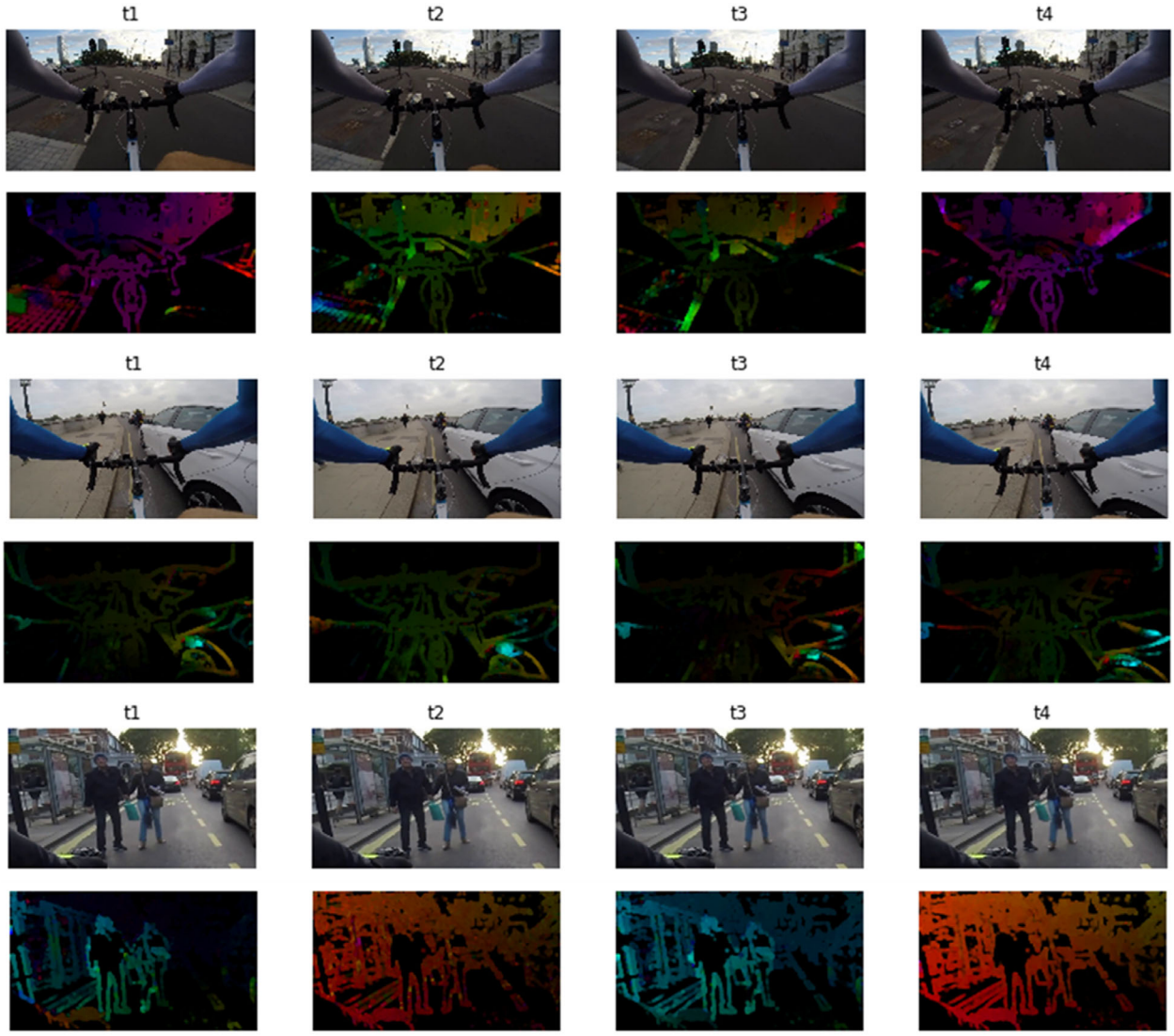

FIGURE 2 A sample of the dataset for the RGB frames and their optical flows

TABLE 2 Classification metrics for CyclingNet

Self-attention Bi CNN-LSTM	Precision	Recall	False-positive rate	F1-score
Validation set	0.994	0.995	0.041	0.994
Test set	0.842	0.927	0.418	0.883

100 epochs, the model has converged and the training has stopped to avoid over-fitness after no significant change in the validation loss. In table 2, we expand further on evaluating the classification of CyclingNet. The table shows high validation in terms of precision, recall, and an *F*1-score, with minimum false-positive rates. The model shows high validation in terms of the true positive of the area under the curve of 0.99 and 0.84 for validation and testing sets, respectively. However, the gap between the values of the validation and testing sets can be explained due to the variations in near miss events, or the limitations of similar events that the model can learn and extract features from for future inference.

4.2 | Baseline evaluation

We experimented with adjusting optical flow to images fusion ratio, model architecture, an optimisation technique, and post-prediction with the classification thresholds aiming to maximise temporal smoothing while reducing the global loss. We found that the global loss can be reduced even by a simple CNN architecture, however, the predicted values are prone to temporal instability. On the contrary, after applying a CNN-LSTM architecture the temporal dependences improved whereas the model outputs a smoothed prediction throughout the clip. We also found that by including bidirectional and self-attention mechanisms in the architecture of the CNN-LSTM model, the losses at the local and global levels of the training and testing datasets have improved in comparison to a CNN-LSTM model. Table 3

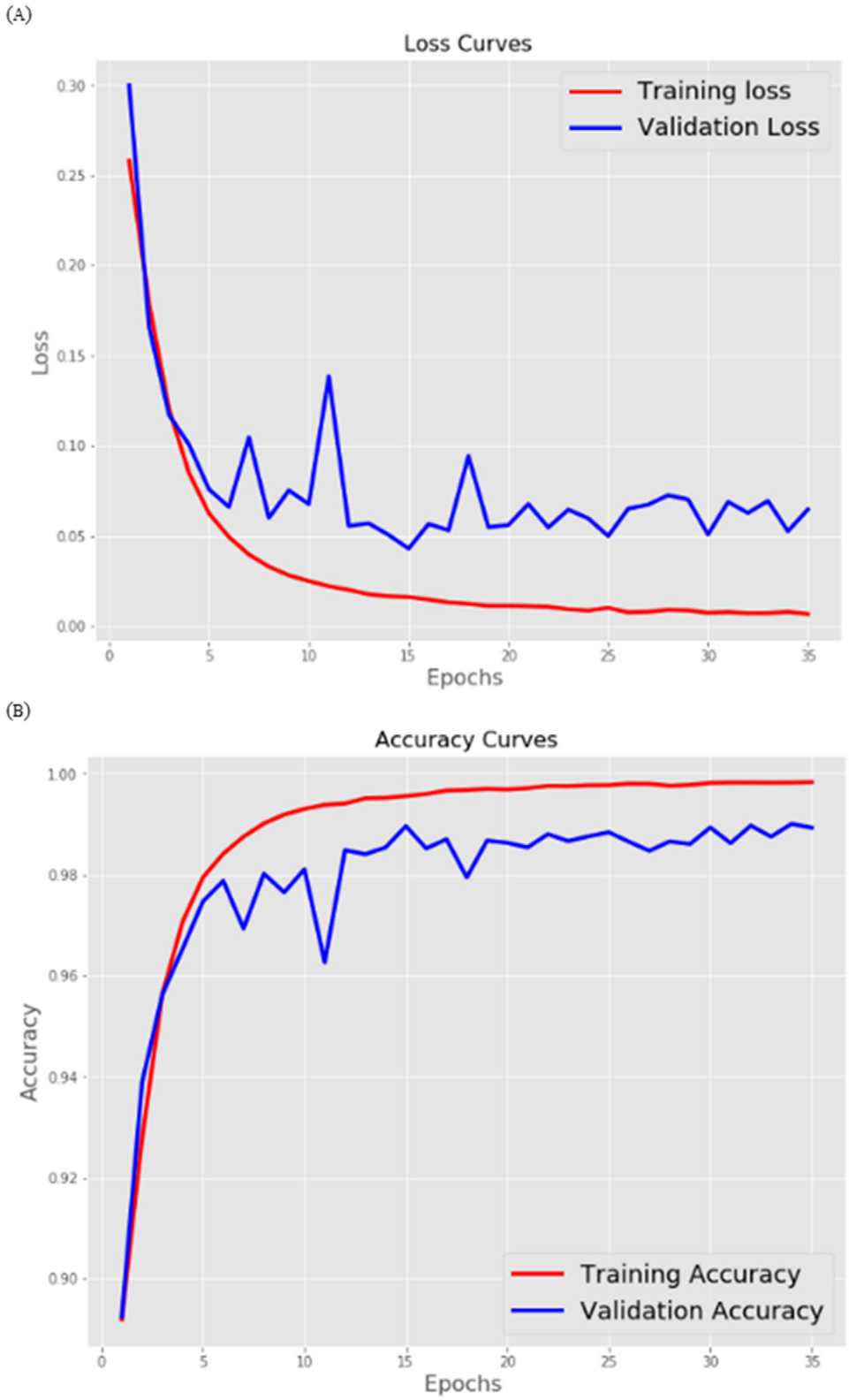

FIGURE 3 Training and evaluation of CyclingNet

TABLE 3 Baseline assessment of CyclingNet

Architecture comparison	Validation Accuracy	Validation Loss
CNN (Block 1-3)	86.5 %	0.73
CNN-LSTM	97%	0.15
Self-attention CNN-LSTM	96 %	0.20
Self-attention Bi CNN-LSTM	**98.9 %**	**0.06**

summarises the outcomes of the different studied architectures on the validation set, with a constant fusion ratio of 50% of the single images and optical flows.

4.3 | Scenes prediction

In figure 4, we show different clips of near misses predicted by CyclingNet. The model shows high accuracy in predicting a wide range of complex urban scenes at different times of the day and in different weather conditions. The model shows high accuracy in predicting near misses including different types of near misses, such as close passes, pedestrian step in, or any risky situation with different road users, including other people on bikes. Similarly, figure 5 shows a variation of urban scenes that has been detected as a safe ride.

5 | DISCUSSION

5.1 | CyclingNet as the state-of-the-art method for detecting cycling near misses

Understanding safety as a clue from the overall scene and the interaction of different road users remains a challenge. In this paper, we introduced CyclingNet as a novel method for detecting cycling near misses from video streams of moving bicycles in a complex urban setting. The model has shown strong performance in detecting near misses, regardless of the complexity of the scene, time of the day, weather, visual conditions, or the placement of the camera on the bike. Due to the absence of other models or benchmark datasets for the stated purpose, it remains a challenge to compare our results to other models, besides the ones we developed as base models. This, however, makes the CyclingNet model a vital and indispensable model for the field of road safety and more specifically, for detecting near misses. Accordingly, this makes it good practice for generalisation, deployment, and transfer learning to detect near misses for other road users or other safety-related domains.

5.2 | Training and inference time

In table 4, we show the training and inference time required to run the introduced model. Training the model on street-level images of both safe rides and near misses took almost 2 days (42 h) on a single GPU (Titan V), but the model can detect cycling near misses at a rate of 15.1 frames per second (FPS), showing a near real-time detection without any post-training quantization. This demonstrates the potential of the method in real-time safety operations such as early warning systems.

TABLE 4 Training and inference time for introduced methods

Method	GPU	Training time	Inference time
CyclingNet	1 (Titan v)	42 hours	15.1 FPS
	1 (RTX 2080 Ti)	-	10.4 FPS

5.3 | Covid-19 pandemic and the increase in the number of people on bikes

Whether temporarily or permanently, it has been debated that there is an increase in the numbers of people on bikes with different profiles and social characteristics due to the Covid-19 pandemic [48]. There is no doubt that this increase in cycling would have direct benefits for health and the environment. With this increase, however, cycling infrastructure needs further preparation to support the increased numbers and more safety-related measures need to be considered. Accordingly, automating the detection of near misses could lead to drawing more significant safety policies for cycling in cities. Nevertheless, we need to understand the capacity of the current cycling infrastructure for a safe ride, and the tipping point for the increase in the number of near misses based on the interaction with other people on bikes or other road users.

5.4 | Risk factors, and their causal inference

Improving transport safety studies, especially for the most vulnerable road users, through automation is still a new domain of research. While detecting cycling near misses is vital for evaluating risky situations and better understanding the experiences of people on bikes, it is still one task towards understanding in depth the reasons for near misses, and their risk factors, and how to avoid them in the future. Accordingly, an AI-embedded system can be developed to capture not only near misses but also to understand the dynamics of the built and natural environments, in addition to detecting and localising either the various road users or the objects that could cause a risky situation for people on bikes. To do so, a pipeline of deep models can be utilised to sense, detect, and extract information of the different layers of the cities (the built environment, natural environment, transport, and infrastructure) to draw significant conclusions [9]. Different models have been proposed that could be utilised for the stated issue such as detecting the state of the environment and counting different road users [10] and recognising weather and visual conditions [11]. After integrating these different deep models, a Bayesian approach for causal inference can be added to understand the effect and influence of each risk factor. We can also understand how such near miss experiences vary with the individual characteristics of the person who cycles such as their age, gender and even modifiable variables such as

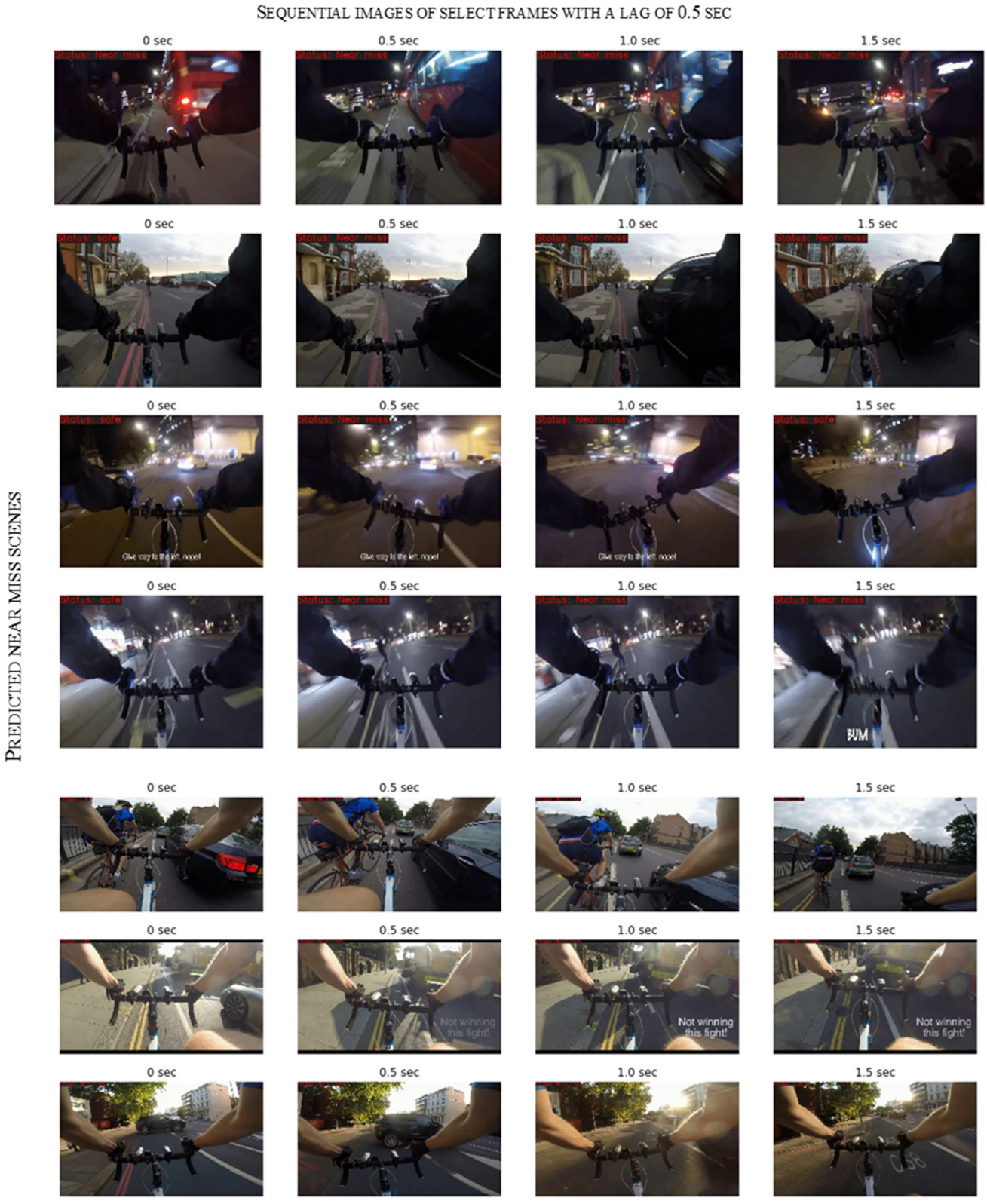

FIGURE 4 Examples of predicted cycling near misses by our model

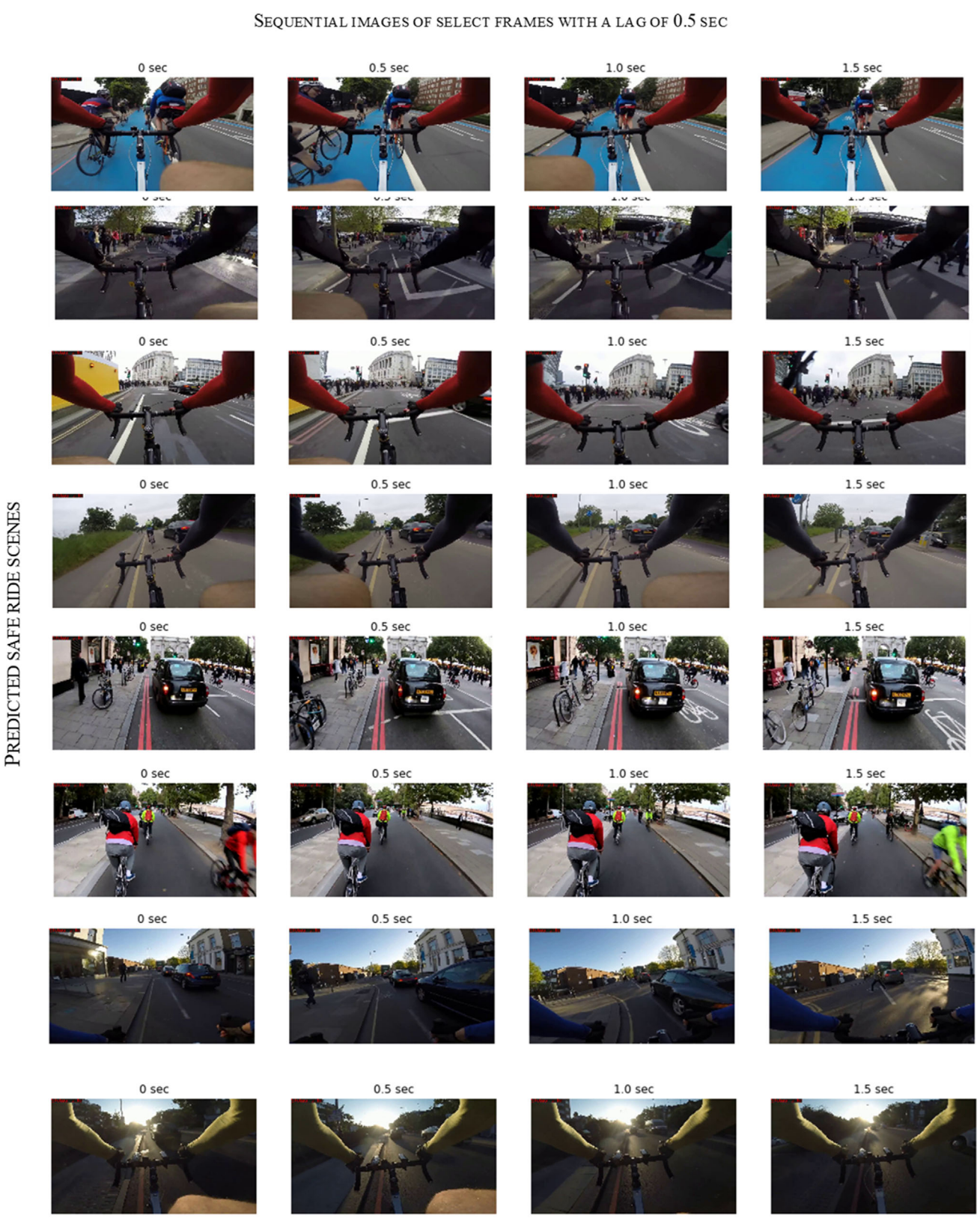

FIGURE 5 Examples of predicted cycling safe ride by our model

the level of training. In this respect, such models represent an efficient way to collect data. Accordingly, new transport safety policies can be taken into consideration, or a guideline for altering individual behaviours while riding bicycles or driving cars can be considered.

5.5 | Model limitations and future work

The model shows high validation for generalisation. However, there are still some limitations that need to be addressed in future work. First, instrumented bikes and groups of volunteered cyclists can be utilised for collecting new datasets which could offer a further verification of the introduced model. Second, introducing a new model to classify the different types of near misses after detection would allow a better understanding of the frequency of the different types of near misses. Third, developing the model to accurately extract the start and end of an event is another domain that needs further research. Finally, applying similar models to detect safety measures and near misses for other road users such as pedestrians and car drivers would allow tailored-made policies or guidelines for the interaction of the different road-users according to the specific type of near misses for a given road-user.

6 | REMARKS

The problem of detecting cycling near misses from video streams of moving bicycles in real-world settings has not been addressed in the current literature. In this paper, we utilised the advances in computer vision and deep learning to detect such events in a near real-time fashion. We introduced the CyclingNet model, a new deep computer vision for detecting cycling near misses from video streams of moving bicycles in complex urban environments. The model is structured as a single stream and trained in an end-to-end fashion, exploiting both single RGB frames, and optical flow data. After training the model with data from both near misses and safe rides, the results show high performance on both training and validation data sets.

The model is intended to be used for generating information that can draw significant conclusions regarding cycling behaviour in cities and elsewhere, which could help planners and policy-makers to better understand the requirement of safety measures when designing infrastructure or drawing policies. As for future work, the model can be pipelined with other state-of-the-art classifiers and object detectors simultaneously to understand the causality of near misses based on factors related to interactions of road users, the built and the natural environments.

ACKNOWLEDGEMENT

This research outcome is a part of a PhD study for the first author at University College London (UCL). This work was supported by UCL Overseas Research Scholarship (ORS) and the Road Safety Trust (RST 38_03_2017). We would like to thank NVIDIA for the GPU grant.

DATA AVAILABILITY STATEMENT

Data available on request from the authors.

ORCID

Mohamed R. Ibrahim https://orcid.org/0000-0001-7733-7777
James Haworth https://orcid.org/0000-0001-9506-4266
Tao Cheng https://orcid.org/0000-0002-5503-9813

REFERENCES

1. De Hartog, J.J., et al.: Do the health benefits of cycling outweigh the risks? Environ. Health Perspect. 118(8), 1109–1116 (2010)
2. Juhra, C., et al.: Bicycle accidents – Do we only see the tip of the iceberg? Injury 43(12), 2026–2034 (2012)
3. Blaizot, S., et al.: Injury incidence rates of cyclists compared to pedestrians, car occupants and powered two-wheeler riders, using a medical registry and mobility data, Rhône County, France. Accid. Anal. Prev. 58, 35–45. (2013)
4. Aldred, R.: Cycling near misses: Their frequency, impact, and prevention. Transp. Res. Part Policy Pract. 90, 69–83 (2016)
5. De Rome, L., et al.: Bicycle crashes in different riding environments in the Australian capital territory. Traffic Inj. Prev. 15(1), 81–88 (2014)
6. Winters, M., Branion-Calles, M.: Cycling safety: Quantifying the under reporting of cycling incidents in Vancouver, British Columbia. J. Transp. Health 7, 48–53 (2017)
7. Ibrahim, M.R., et al.: Cycling near misses: A review of the current methods, challenges and the potential of an AI-embedded system. Transp. Rev. 0(0), 1–25 (2020)
8. Nguyen, H., et al.: Deep learning methods in transportation domain: A review. IET Intell. Transp. Syst. 12(9), 998-1004 (2018)
9. Ibrahim, M.R., Haworth, J., Cheng, T.: Understanding cities with machine eyes: A review of deep computer vision in urban analytics. Cities 96, 102481, (2020)
10. Ibrahim, M.R., Haworth, J., Cheng, T.: URBAN-i: From urban scenes to mapping slums, transport modes, and pedestrians in cities using deep learning and computer vision. Environ. Plan. B Urban Anal. City Sci. 48(1), 76–93 (2019)
11. Ibrahim, M.R., Haworth, J., Cheng, T.: WeatherNet: Recognising weather and visual conditions from street-level images using deep residual learning. ISPRS Int. J. Geo-Inf. 8(12), 549, (2019)
12. Johnson, M., et al.: Naturalistic cycling study: Identifying risk factors for on-road commuter cyclists. Ann. Adv. Automot. Med. Annu. Sci. Conf. 54, 275–283 (2010)
13. Chaurand, N., Delhomme, P.: Cyclists and drivers in road interactions: A comparison of perceived crash risk. Accid. Anal. Prev. 50, 1176–1184 (2013)
14. Strauss, J., Miranda-Moreno, L.F., Morency, P.: Cyclist activity and injury risk analysis at signalized intersections: A Bayesian modelling approach. Accid. Anal. Prev. 59, 9–17 (2013)
15. Sanders, R.L.: Perceived traffic risk for cyclists: The impact of near miss and collision experiences. Accid. Anal. Prev. 75, 26–34 (2015)
16. Poulos, R.G., et al.: Exposure-based cycling crash, near miss and injury rates: The safer cycling prospective cohort study protocol: Figure 1. Inj. Prev. 18(1), e1–e1 (2012)
17. Aldred, R., Goodman, A.: Predictors of the frequency and subjective experience of cycling near misses: Findings from the first two years of the UK near miss project. Accid. Anal. Prev. 110, 161–170 (2018)
18. Paschalidis, E., Basbas, et al: Put the blame on…others!.: The battle of cyclists against pedestrians and car drivers at the urban environment. A cyclists. perception study 18, (2016)
19. Lehtonen, E., et al.: Evaluating bicyclists. risk perception using video clips: Comparison of frequent and infrequent city cyclists. Transp. Res. Part F Traffic Psychol. Behav. 41, 195–203 (2016)
20. Vansteenkiste, P., et al.: A hazard-perception test for cycling children: An exploratory study. Transp. Res. Part F Traffic Psychol. Behav. 41, 182–194 (2016)

21. Dozza, M., Werneke, J.: Introducing naturalistic cycling data: What factors influence bicyclists. safety in the real world?. Transp. Res. Part F Traffic Psychol. Behav. 24, 83–91 (2014)
22. Johnson, M., et al.: Cyclists and open vehicle doors: Crash characteristics and risk factors. Saf. Sci. 59, 135–140 (2013)
23. Dozza, M., Bianchi Piccinini, G.F., Werneke, J.: Using naturalistic data to assess e-cyclist behavior. Transp. Res. Part F Traffic Psychol. Behav. 41, 217–226 (2016)
24. Dozza, M., Werneke, J.: Introducing naturalistic cycling data: What factors influence bicyclists. safety in the real world?. Transp. Res. Part F Traffic Psychol. Behav. 24, 83–91 (2014)
25. Schleinitz, K., et al.: The German naturalistic cycling study – comparing cycling speed of riders of different e-bikes and conventional bicycles. Saf. Sci. 92, 290–297 (2017)
26. Kang, S.M., Wildes, R.P.: Review of action recognition and detection methods. ArXiv161006906 Cs (2016)
27. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes From videos in the wild. arXiv12120402 Cs (2012)
28. Kuehne, H., et al.: HMDB: A large video database for human motion recognition. In: 2011 International Conference on Computer Vision. 2011 IEEE International Conference on Computer Vision (ICCV), Barcelona, Spain, 6–13 November 2011
29. Andriluka, M., et al.: 2D human pose estimation: New benchmark and state of the art analysis. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014
30. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. arXiv14062199 Cs (2014)
31. Buch, S., et al.: SST: Single-stream temporal action proposals. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)' 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017
32. Diba, A., et al.: Temporal 3D ConvNets: New architecture and transfer learning for video classification. arXiv171108200 Cs (2017)
33. Shou, Z., Wang, D., Chang, S.-F.: Temporal action localization in untrimmed videos via multi-stage CNNs. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)' 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016
34. Xue, F., et al.: Attention-based spatial–temporal hierarchical ConvLSTM network for action recognition in videos. IET Comput. Vision 13(8), 708–718 (2019)
35. Gibson, J.J.: The perception of the visual world. Houghton Mifflin, Boston (1950)
36. Wang, L., Qiao, Y., Tang, X.: Action recognition with trajectory-pooled deep-convolutional descriptors. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015
37. Ng, J.Y.-H., et al.: Beyond short snippets: Deep networks for video classification. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)' 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015
38. Wu, Z., et al.: Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. arXiv150401561 Cs (2015)
39. Girdhar, R., Ramanan, D.: Attentional pooling for action recognition. arXiv171101467 Cs (2017)
40. Wang, L., et al.: Towards good practices for very deep two-stream ConvNets. arXiv150702159 Cs (2015)
41. Farnebäck, G.: Two-Frame Motion Estimation Based on Polynomial Expansion. In: Bigun, J., Gustavsson, T. (eds.) Image Analysis, vol. 2749, pp. 363–370. Springer, Berlin Heidelberg (2003)
42. He, K., et al.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: 'Proceedings of the IEEE international conference on computer vision, Santiago, Chile, 7–13 December 2015
43. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning, The MIT Press, (2017)
44. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521(7553), 436–444 (2015)
45. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention Is All You Need. ArXiv170603762 Cs (2017)
46. Xu, K., et al.: Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. arXiv150203044 Cs (2016)
47. Mikolajczyk, A., Grochowski, M.: Data augmentation for improving deep learning in image classification problem. in 2018 International Interdisciplinary PhD Workshop (IIPhDW), IEEE, pp. 117–122 (2018)
48. DfT: Transport use during the coronavirus (COVID-19) pandemic, (2020)

How to cite this article: Ibrahim, M.R., et al.: CyclingNet: Detecting cycling near misses from video streams in complex urban scenes with deep learning. IET Intell. Transp. Syst. 15, 1331–1344 (2021). https://doi.org/10.1049/itr2.12101