Fast Wrong-way Cycling Detection in CCTV Videos: Sparse Sampling is All You Need

Jing Xu[12†], Wentao Shi[1†], Sheng Ren[1], Lijuan Zhang[1], Weikai Yang[2*], Pan Gao[1*] and Jie Qin[1]

***Abstract*—Effective monitoring of unusual transportation behaviors, such as wrong-way cycling (i.e., riding a bicycle or e-bike against designated traffic flow), is crucial for optimizing law enforcement deployment and traffic planning. However, accurately recording all wrong-way cycling events is both unnecessary and infeasible in resource-constrained environments, as it requires high-resolution cameras for evidence collection and event detection. To address this challenge, we propose WWC-Predictor, a novel method for efficiently estimating the wrong-way cycling ratio, defined as the proportion of wrong-way cycling events relative to the total number of cycling movements over a given time period. The core innovation of our method lies in accurately detecting wrong-way cycling events in sparsely sampled frames using a light-weight detector, then estimating the overall ratio using an autoregressive moving average model. To evaluate the effectiveness of our method, we construct a benchmark dataset consisting of 35 minutes of video sequences with minute-level annotations. Our method achieves an average error rate of a mere 1.475% while consuming only 19.12% GPU time required by conventional tracking methods, validating its effectiveness in estimating the wrong-way cycling ratio.**

***Index Terms*—Wrong-way cycling, video analysis, tracking.**

I. Introduction

Wrong-way cycling refers to the behavior of riding a bicycle or e-bike in the opposite direction of the designated traffic flow. This behavior is a serious violation that significantly increases the risk of collisions, posing safety risks for both cyclists and other road users. While robust enforcement systems exist for motor vehicle violations, they typically rely on high-resolution CCTV cameras and significant computational resources for license plate recognition and violation evidence collection [1], [2]. The high cost and resource intensity of such systems restricts their widespread deployment for monitoring non-motorized transport effectively.

Given these constraints, our focus shifts from identifying and penalizing individual offenders to a macro-level safety management and transport system design. Specifically, we focus on a key metric: the wrong-way cycling ratio, i.e., the proportion of wrong-way cycling instances relative to the total number of cycling movements over a given time period. This ratio is crucial for assessing area-specific safety levels, identifying hotspots that require targeted interventions, and providing vital data for urban planning and safety optimization. The primary need, therefore, is to develop an efficient and scalable method to calculate this ratio across entire road networks, leveraging the resource-constrained CCTV feeds already widely deployed for general traffic observation.

To address this need, we propose the **W**rong-**W**ay **C**ycling Predictor (WWC-Predictor), a lightweight method designed to efficiently estimate wrong-way cycling ratio using significantly fewer frames and less computational resources. This is achieved through sparse sampling supported by a Two-Frame WWC Detector, which precisely extracts orientation-based counts from each pair of frames. To mitigate orientation errors caused by detection in sparse sampling (e.g., occlusion ambiguities), we introduce an ensemble method to cross-validate the cycling orientations detected in each pair of frames. Subsequently, a temporal WWC estimator applies an ARMA model to convert validated frame-level counts into a video-level wrong-way cycling ratio. The extreme lightweight design enables real-time inference on edge devices while maintaining robust estimation.

To evaluate our method, and to foster future research on this task, we construct a benchmark containing 405 annotated images for non-motor vehicle detection task, 1199 images for orientation estimation task, and 4 fully annotated CCTV videos (35 minutes in total) for end-to-end validation. The evaluation result on our benchmark demonstrates that WWC-Predictor achieves accuracy comparable to conventional tracking-based methods while requiring 6-10 times fewer frames and 4-6 times less computational resources, confirming the effectiveness of our method.

In summary, the contribution of our work includes

- a two-frame wrong-way cycling detector, which robustly detects orientations of non-motor vehicles in each sparsely sampled frame pair,
- a temporal wrong-way cycling estimator, which forecasts video-level wrong-way cycling ratios utilizing ARMA time-series model, and
- a benchmark dedicated to the wrong-way cycling ratio estimation task.

II. Related Work

A. Detection of Wrong-way Incidents

Wrong-way driving represents a closely related area that has garnered significant attention. Current studies [3]–[8] have

[1]The College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China. {jing.xu, shiwentao, rensheng, lijuan.zhang, pan.gao, jie.qin}@nuaa.edu.cn
[2]The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China. {jing.xu@connect., weikaiyang@}hkust-gz.edu.cn
†Co-first author
*Co-corresponding author

developed frameworks for the detection of wrong-way driving in CCTV footage. These frameworks employ a variety of multi-object tracking (MOT) methods, including FastMOT [9], DeepSORT [10], Kalman filter [11] and centroid tracking [6], to analyze vehicular movements. However, these MOT methods, especially those not based on detection [12]–[14], require a substantial amount of labeled video data for recognizing the same instances across different frames, and annotating those videos is both time-consuming and costly; contemporary open vocabulary models [15], [16] are also not readily applicable in such specialized domains.

The approach detailed in [3], which utilizes FastMOT, segments video footage into one-minute intervals. This segmentation allows for the precise identification of vehicle start and end points through tracking models, with subsequent post-processing to determine vehicle orientation. Contrarily, other strategies involve analyzing entire video sequences, employing detection-based methods followed by post-processing to ascertain the direction of travel. Suttiponpisarn *et al*. [7], [17] developed a system utilizing YOLOv4-tiny [18] to detect object and DeepSORT tracking to track vehicles, in which two main algorithms called Road Lane Boundary detection from CCTV algorithm (RLB-CCTV) and Majority-Based Correct Direction Detection algorithm (MBCDD) are designed and improved to consume less computational time. Choudhari *et al*. [19] introduced a continuous tracking method for monitoring the direction of motorbikes, comparing it against the expected direction of the lane. If the tracked orientation of a motorbike is contrary to the lane direction for at least 80% of the observed time, it is identified as wrong-way riding. In essence, the prevailing strategies for detecting wrong-way incidents predominantly rely on tracking methods. While effective, these methods are computationally intensive [5], [7], often requiring significant GPU resources [2], [3] and processing time [6], which becomes particularly challenging when analyzing extensive CCTV footage from multiple cameras. This computational burden represents a critical limitation in real-world deployments and underscores the need for more efficient approaches that can maintain acceptable accuracy while reducing resource requirements.

It is also popular to utilize GPS information to detect wrong-way cycling. Gu *et al*. [20] proposed BikeMate, a ubiquitous bicycling behavior monitoring system with smartphones. Hayashi *et al*. [21] presented a mobile system that performed vision-based scene analysis to detect potentially dangerous cycling behavior including wrong-way cycling. Dhakal *et al*. [22] used data collected from a smartphone application to explain wrong-way riding behavior of cyclists on one-way segments to help better identify the demographic and network factors influencing the wrong-way riding decision making.

B. Orientation Detection

Originally, detecting the orientation of non-motor vehicles in videos poses a dynamic challenge. However, within our sparse analysis framework, this issue transitions into a more static scenario, prompting us to delve into orientation detection within still images. Although direct analogues—tasks with an image input leading to an orientation output—are scarce, there exists a foundation of architectures addressing orientation in various contexts. Historically, orientation challenges have often manifested within the realm of oriented object detection [23]–[25]. This methodology aims not only to delineate the object with a bounding box but also to ascertain its orientation, thereby enhancing the precision of detection beyond the capabilities of conventional object detection techniques.

A novel contribution to this field is introduced by Yu *et al*. [26] through the development of a differentiable angle coder, termed the phase-shifting coder (PSC). This innovative approach tackles the issue of orientation cyclicity by translating the rotational periodicity across various cycles into the phase of differing frequencies, effectively addressing the rotation continuity problem. In our project, we incorporate the PSC to aid in resolving the challenges associated with orientation detection, leveraging its advanced capability to understand and quantify orientation within images accurately.

C. Mathematical Modeling of Traffic Flow

Given our focus on sparse time sequences, it is essential to establish connections among these sequences rather than treating them as isolated samples. The modeling of traffic flow serves as a pertinent example of this approach.

Tian *et al*. [27] suggested modeling the frequency components of network traffic using the ARMA model, illustrating a method to grasp the temporal dynamics in data. Similarly, Peng *et al*. [28] introduced an ARIMA-SVM combined prediction model to forecast urban short-term traffic flow, demonstrating the efficacy of integrating traditional statistical models with machine learning techniques for enhanced predictive accuracy. These methodologies underscore the significance of training models on a set of pre-measured network traffic data, enabling the ARMA or ARIMA-SVM models to capture the intrinsic characteristics of traffic flows and, subsequently, to forecast network traffic effectively.

Recent advances in Graph Signal Processing (GSP) have demonstrated significant potential for video analysis [29]. Fundamental to GSP is the extension of traditional signal processing concepts to irregular domains [30], with ARMA graph filters [31] providing a particularly relevant framework for our work. These techniques model signals evolving over graph structures, where temporal dynamics can be represented as graph signals along time-vertices [32], and ARMA graph filters capture recursive relationships in signal evolution [31]. While our immediate application focuses on single-camera traffic flow, the GSP perspective offers pathways for future extension to multi-camera networks using graph-based fusion.

In our analysis, although prediction is not our primary aim, we employ ARMA model to distill specific features that elucidate the relationships among samples.

III. Method Overview

Taking a raw video $\mathcal{V}$ as input, our method predicts the wrong-way cycling ratio through two integrated stages: the sparse detection and the temporal estimation. In the sparse detection stage, we uniformly sample $\mathcal{V}$ at fixed intervals

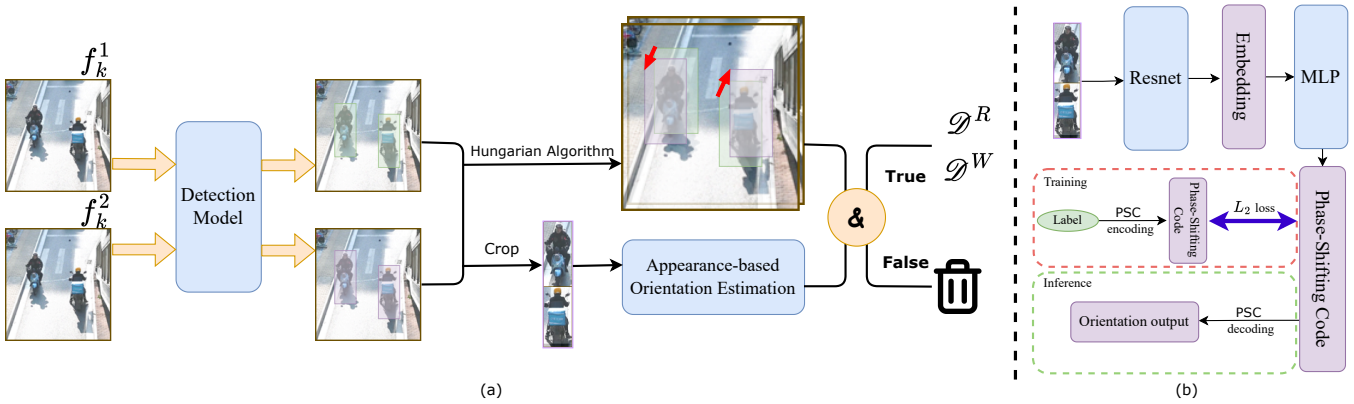

Fig. 1. (a): Overview of Two-Frame Wrong-Way Cycling Detector, which consists of a detection model, an appearance-based orientation model and an And-strategy (shown as & on the figure). (b): Training and inference pipeline of our proposed appearance-based orientation estimation model.

T_{gap} to generate sequential frame pairs $\mathcal{S} = \{\mathcal{S}_0, \mathcal{S}_1, \ldots\}$, where each pair $\mathcal{S}_k$ consists of two consecutive frames. Each pair $\mathcal{S}_k$ is processed by our **two-frame wrong-way cycling detector**, which produces sparse time-stamped counts of right-way cycling events $\mathcal{D}_k^R$ and wrong-way cycling events $\mathcal{D}_k^W$. In the temporal estimation stage, the sparse detection results serve as input to our **temporal wrong-way cycling ratio estimator**, which models the number of events across the entire time period using an autoregressive moving average model to estimate the wrong-way cycling ratio. This two-stage design enables efficient processing by combining targeted detection at key frames with statistical modeling for comprehensive temporal coverage.

IV. Two-Frame Wrong-Way Cycling Detector

Given a two-frame pair $\mathcal{S}_k = (f_k^1, f_k^2)$ from video $\mathcal{V}$, our two-frame wrong-way cycling detector determines vehicle orientations and classifies right-way/wrong-way cycling instances through a three-stage pipeline. As illustrated in Figure 1, our method integrates: 1) motion-based orientation estimation, which analyzes inter-frame object displacement to infer travel direction, 2) appearance-based orientation estimation, which leverages deep learning to predict orientation from vehicle appearance, and 3) ensemble validation, which cross-validates predictions from both methods to ensure robust classification.

A. Motion-Based Orientation Estimation

The motion-based orientation estimation method determines vehicle travel direction by analyzing displacement between consecutive frames. This process involves three steps: vehicle detection, vehicle matching, and orientation calculation.

Vehicle Detection. The first step detects non-motor vehicles in each frame pair $\mathcal{S}_k$ through a detection model, which generates bounding box sets $\mathbf{B}_1 = D(f_k^1)$ and $\mathbf{B}_2 = D(f_k^2)$ for each pair. Here, $D(\cdot)$ represents the detection function applied to frames, and we employ YOLOv5 [33] due to its proven industrial efficacy in real-time object recognition.

Vehicle Matching. To establish correspondences between detected vehicles across paired frames, we compute a similarity matrix using Intersection-over-Union (IoU) metrics. The function F_{IoU} generates $\mathbf{X}_{\text{IoU}} \in \mathbb{R}^{|\mathbf{B}_1| \times |\mathbf{B}_2|}$ where each element $\mathbf{X}_{\text{IoU}}^{i,j}$ quantifies spatial overlap between detected boxes. To exclude stationary objects, we apply a threshold mask:

$$\mathbf{X}_{\text{IoU}} = \mathbf{X}_{\text{IoU}} \odot (\mathbf{X}_{\text{IoU}} < \text{IoU}_{\max}),$$

where $\text{IoU}_{\max} = 0.98$ excludes high-overlap matches that likely represent stationary objects. The optimal bipartite matching is then obtained using the Hungarian algorithm $\mathbf{H}(\cdot)$, which produces matched index pairs $L_{\text{match}} = \{(i_1, j_1), (i_2, j_2), \ldots\}$.

Orientation Calculation. For each valid match (i, j) in L_{match}, we compute the geometric orientation by analyzing the displacement vector between frame centroids:

$$O_{\text{det}} = \arctan(\mathbf{Cen}(\mathbf{B}_2^j) - \mathbf{Cen}(\mathbf{B}_1^i)),$$

where $\mathbf{Cen}(\cdot)$ denotes the centroid of a bounding box.

B. Appearance-Based Orientation Estimation

While motion-based estimation provides reliable orientation cues from inter-frame displacement, it may fail in scenarios with insufficient movement, occlusion, or ambiguous trajectories. To address these limitations, we introduce an appearance-based orientation estimation model that predicts vehicle direction directly from visual appearance.

Figure 1 illustrates the architecture of our appearance-based orientation estimation model, which takes an image of vehicle as input, and output its facing orientation. Since the model output a continuous and periodic variable, we applied Phase-Shifting Coder (PSC) [26] to transform the discontinuous degree system into continuous m-dimension vector. The PSC works as follows:

Encoding:

$$x_i = \cos\left(\varphi + \frac{2i\pi}{m}\right), i = 1, 2, \ldots, m. \tag{1}$$

Decoding:

$$\varphi = -\arctan \frac{\sum_{i=1}^{m} x_i \sin(\frac{2i\pi}{m})}{\sum_{i=1}^{m} x_i \cos(\frac{2i\pi}{m})}. \tag{2}$$

In our work, we set m as 3, which strikes a good balance between representation accuracy and computational efficiency.

Algorithm 1 Two-Frame Wrong-Way Cycling Detector

Require: whole video $\mathcal{V}$, right-way orientation O_{right}

Sparse sampling: $\mathcal{S} = \{\mathcal{S}_0, \mathcal{S}_1, \ldots\}$ from $\mathcal{V}$, where each $\mathcal{S}_k = (f_k^1, f_k^2)$ is a pair of consecutive frames
for (f_k^1, f_k^2) in $\mathcal{S}$ **do**
  Apply detection model: $\mathbf{B}_1 \leftarrow D(f_k^1)$, $\mathbf{B}_2 \leftarrow D(f_k^2)$
  Compute IoU: $\mathbf{X}_{iou} \leftarrow F_{iou}(\mathbf{B}_1, \mathbf{B}_2)$
  $\mathbf{X}_{iou} \leftarrow (\mathbf{X}_{iou} < IoU_{max}) \cdot \mathbf{X}_{iou}$
  Apply Hungarian algorithm: $L_{match} \leftarrow \mathbf{H}(\mathbf{X}_{iou})$
  for i, j in L_{match} **do**
    Compute orientation in two methods:
    $O_{det} \leftarrow \mathbf{Orient}(\mathbf{Cen}(\mathbf{B}_1^i), \mathbf{Cen}(\mathbf{B}_2^j))$
    $O_{model} \leftarrow \mathbf{Ave}(\mathbf{F}_o(\mathbf{B}_1^i, f_k^1), \mathbf{F}_o(\mathbf{B}_2^j, f_k^2))$
    Pick out valid instances:
    if $\mathbf{AndStrategy}(O_{det}, O_{model})$ **then**
      $OrientationList.\mathbf{append}(\mathbf{Ave}(O_{det}, O_{model}))$
    end if
  end for
end for
Count for final number:
$is\_Right \leftarrow \mathbf{Dis}(OrientationList, O_{right}) < \frac{2}{3}\pi$
$\mathcal{D}_R \leftarrow \mathbf{SUM}(is\_Right)$
$\mathcal{D}_W \leftarrow \mathbf{SUM}(\neg is\_Right)$
return $\mathcal{D}^R, \mathcal{D}^W$

More specifically, we apply pretrained backbone, Resnet-101 [34] here, to generate embedding $\mathbf{b} \in \mathbb{R}^n$ for an image, and a linear layer is applied to convert the embedding $\mathbf{b}$ to vector $\hat{\mathbf{x}} \in \mathbb{R}^m$.

During training process, the label $\varphi \in (-\pi, \pi]$ is encoded to $\mathbf{x} \in \mathbb{R}^m$. Then the loss is computed as follows:

$$l = \frac{1}{m}\|\hat{\mathbf{x}} - \mathbf{x}\|^2. \tag{3}$$

During inference, the vector $\hat{\mathbf{x}} \in \mathbb{R}^m$ is decoded to $\hat{\varphi} \in (-\pi, \pi]$ as the final output.

In the orientation prediction task, we define the metric as the distance between the prediction and the label. Since it is a cyclic number, its formula can be expressed as:

$$\begin{aligned} d &= \max(\varphi, \hat{\varphi}) - \min(\varphi, \hat{\varphi}). \\ \text{Error} &= \begin{cases} d, & \text{if } d \leq \pi, \\ 2\pi - d, & \text{otherwise.} \end{cases} \end{aligned} \tag{4}$$

Here, φ represents the label value and $\hat{\varphi}$ represents the predicted value. Both φ and $\hat{\varphi}$ are in the range of $(0, 2\pi]$ and are measured in degrees. The error is computed based on the difference between the maximum and minimum values of φ and $\hat{\varphi}$. If this difference is less than or equal to π, the error is equal to d. Otherwise, if the difference is greater than π, the error is computed as $2\pi - d$. In the experimental part, we utilize this metric to evaluate the performance of our model.

To train our appearance-based orientation estimation model, we initially plan to fine-tune a pretrained ViT [35] on limited manually collected and annotated orientation data. However, manually collected data often exhibits similar orientation patterns, resulting in imbalanced data distribution. Recognizing the challenges associated with obtaining real-world data with precise orientation labels and addressing long-tail attributes, we generate synthetic data to conduct task-specific pretraining on the model and to provide it with a beneficial bias towards orientation information. By leveraging this approach, we aim to enhance the model's ability to understand and utilize orientation cues in real-world scenarios.

Fig. 2. One case reconstructed and generated by instant-ngp.

Firstly, we capture 360-degree videos using a regular camera in real-world settings. We then capture 30-40 frames from the video and utilize a trained COLMAP [36], [37] (Structure-from-Motion and Multi-View Stereo) system to estimate the camera's position and attitude parameters both inside and outside the captured scenes. Using these information, we reconstruct a 3D model using instant-ngp [38].

Next, we leverage pre-designed camera poses to render images with corresponding orientation labels. To simulate real-world conditions, we render images from different heights for each orientation as shown in Figure 2. We capture images at every 10-degree interval with 3 different height, resulting in 72 labeled images for each 3D model.

This process allows us to generate a diverse dataset of labeled images, which serves as valuable task-specific pre-training data for our orientation-aware model.

C. Ensemble Validation

Building on the orientation obtained from both motion-based and appearance-based methods, we employ an ensemble validation strategy to ensure robust and reliable orientation classification.

First, to obtain robust orientation estimates from the two complementary methods, we implement an And-strategy that integrates outputs from both the motion-based estimation (O_{det}) and the appearance-based model (O_{model}). Specifically, predictions are considered valid only when the absolute difference between the two orientation estimates falls below a predefined threshold $|O_{\text{det}} - O_{\text{model}}| < \text{Div}_{\max}$. We set $\text{Div}_{max} = \frac{2}{3}\pi$ in our implementation.

For instances that pass the cross validation step, we compute the final orientation by averaging the two validated predictions. This final orientation is then compared against the expected right-way direction O_{right}. If the angular distance is less than $\frac{2}{3}\pi$, the instance is classified as a right-way cycling instance; otherwise, it is considered wrong-way cycling. This process ultimately yields the counts of right-way cycling ($\mathcal{D}_R$) and wrong-way cycling ($\mathcal{D}_W$) instances.

Mathematically, it can be shown that employing an And-strategy with an ensemble of two models can enhance overall performance compared to relying on a single model. Specif-

ically, when the accuracy of both models exceeds 50%, this strategy guarantees improved performance.

Lemma 1. *Assume that P_1 and P_2 represent the respective posterior probabilities that each of the two models errs, which are in the range of $(0, 0.5]$, $P_w = P_1P_2$ represents the possibility of two models erring simultaneously, and $P_{valid} = (1 - P_1)(1 - P_2) + P_1P_2$ represents the possibility of the sample being valid. The possibility of the ensemble model erring $P_3 = \frac{P_w}{P_{valid}} \leq min\{P_1, P_2\}$.*

Proof. Considering P_3 as a function related to P_1, P_2, we have

$$P_3 = f(P_1, P_2) = \frac{P_1P_2}{(1 - P_1)(1 - P_2) + P_1P_2}. \tag{5}$$

Deriving f with respect to P_1 (or equivalently P_2) yields:

$$\begin{aligned} &\frac{\partial f}{\partial P_1} = \frac{-P_2^2 + P_2}{(1 - P_1 - P_2 + 2P_1P_2)^2}, \\ &\frac{\partial f}{\partial P_1} > 0, \text{ for } P_1, P_2 \in (0, 0.5]. \end{aligned} \tag{6}$$

This establishes P_3 as a monotonically increasing function relative to P_1 and P_2. As a result,

$$P_3 \leq f(P_1, 0.5) = P_1, P_3 \leq f(0.5, P_2) = P_2. \tag{7}$$

□

V. Temporal Wrong-Way Cycling Ratio Estimator

Using the Two-Frame Wrong-Way Cycling Detector, we can determine the orientations of non-motor vehicles at discrete time points. However, reconstructing temporal information from such sparse data remains challenging, since each instance may appear at multiple time points. Our Temporal WWC Estimator addresses this by estimating the number of instances (N_R and N_W) passing through each time interval T_{gap} using the counts of instances ($\mathcal{D}_R$ and $\mathcal{D}_W$).

Formally, let N_k be the number of vehicles passed the time interval T_{gap} between t_{k-1} and t_k, we have the estimation $\hat{N}_k = \mathcal{D}_k - \varphi\mathcal{D}_{k-1}$, where the parameter φ quantifies the probability of a vehicle from the previous frame persisting into the subsequent one. To effectively estimate the parameter φ, we employ the Auto Regressive Moving Average (ARMA) model, which can be generally described as follows:

$$\mathcal{D}_k = c + \epsilon_k + \sum_{i=1}^{p} \varphi_i \mathcal{D}_{k-i} + \sum_{j=1}^{q} \theta_j \epsilon_{k-j}, \tag{8}$$

where $\mathcal{D}_k$ represents vehicle counts at time interval k, c is the constant term representing baseline traffic flow, $\epsilon_k \sim \mathcal{N}(0, \sigma^2)$ captures random fluctuations that are not explained by the past values, and φ_i and θ_i are the auto-regressive and moving average coefficients, respectively.

The sum $\Sigma_{i=1}^{p} \varphi_i \mathcal{D}_{k-i}$ is the autoregression (AR) part, where p is the order of the AR process. Each φ_i is a parameter that multiplies the number of vehicles at a previous time point $k-i$, indicating how past values are weighted in the model. In our scenario, p is selected as one, so that this term is simplifed as $\varphi\mathcal{D}_{k-i}$.

The expression $\Sigma_{i=1}^{q} \theta_j \epsilon_{k-j}$ represents the moving average (MA) part, wherein q denotes the order of the MA process. Each coefficient θ_j corresponds to a parameter that is applied to the lagged error term ϵ_{k-j}, illustrating the impact of historical forecast errors on the current value. Within our traffic flow context, this encapsulates the momentum of vehicular movement, particularly in scenarios involving right-way cycling. Conversely, for instances of wrong-way cycling, which are irregular and unforeseen, the MA component is disregarded, thereby setting q to zero.

After building our ARMA models, we can estimate the parameters ϕ and θ using maximum likelihood estimation [39], [40]. We then derive sets of $\hat{N}^R$ and $\hat{N}^W$, which can be used to calculate the wrong-way cycling ratio:

$$\text{WWC Ratio} = \frac{E(N_W)}{E(N_R) + E(N_W)} = \frac{\sum_i \hat{N}_{i,W}}{\sum_i \hat{N}_{i,R} + \sum_i \hat{N}_{i,W}}. \tag{9}$$

VI. Experiments

A. Benchmark Datasets

Given that existing datasets related to non-motor vehicles are either not publicly available or exhibit varying quality, we recognized the need to create a new dataset from scratch. This effort aims to enhance the reliability of our research and support the training and evaluation of our method. To facilitate further research in this area, we propose three distinct datasets: the non-motor vehicle detection dataset, the orientation prediction dataset, and the WWC ratio estimation dataset. Each dataset serves a specific role in our model development and evaluation process.

1) Non-motor Vehicle Detection dataset: This dataset is designed for training the detection model. It contains 223 images with 474 non-motor vehicle bounding boxes captured under diverse conditions. The training-validation split is 8:2, which our experiments confirm as sufficient for effective model training.

2) Orientation Prediction Dataset: This dataset is designed for training the appearance-based orientation prediction model. It comprises three subsets: the pretraining set, the finetuning set, and the validation set. The pretraining set consists of synthetic images generated using instant-ngp. It includes 12 distinct 3D models, and for each model, we generate 72 labeled images captured from two different heights. The details has been introduced in Sec. IV-B. The finetuning set and the validation set contains 1,060 and 169 real-world images with manually-labeled orientation, respectively.

3) WWC Ratio Estimation Dataset: This dataset comprises four videos captured by road cameras situated in various locations: three short videos, each lasting 5 minutes (denoted as Case 1, Case 2, and Case 3), and one long video lasting 20 minutes (denoted as Case 4). As illustrated in Figure 3, which displays screenshots and detection results, this diverse collection of footage lays a solid foundation for evaluating the performance of different methods in terms of both speed and accuracy. By including both short and long videos, we can assess the methods' ability to predict the wrong-way cycling ratio under various conditions, thereby validating their effectiveness in capturing both short-term behaviors and

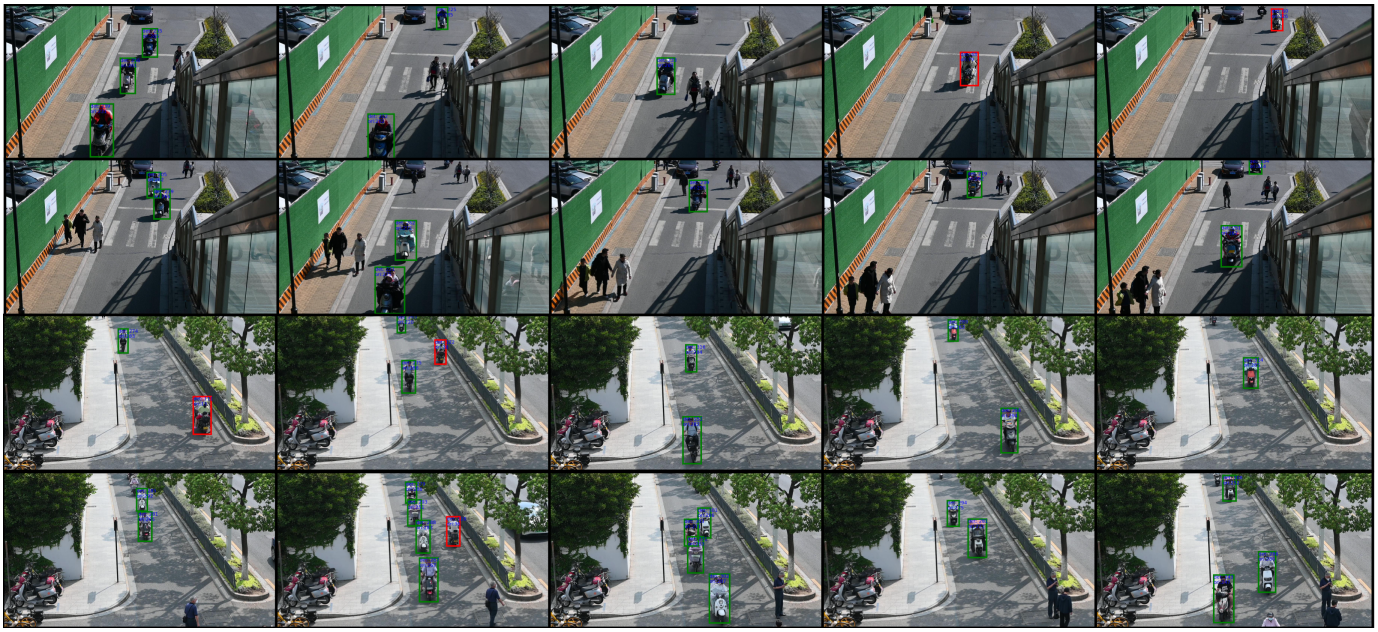

Fig. 3. Visualization from the validation videos for wrong-way cycling prediction in video, where green bounding box means right-way cycling, red means wrong-way cycling. 'Det' and 'Ori' refer to the output orientation from the detection-based branch and the appearance-based branch.

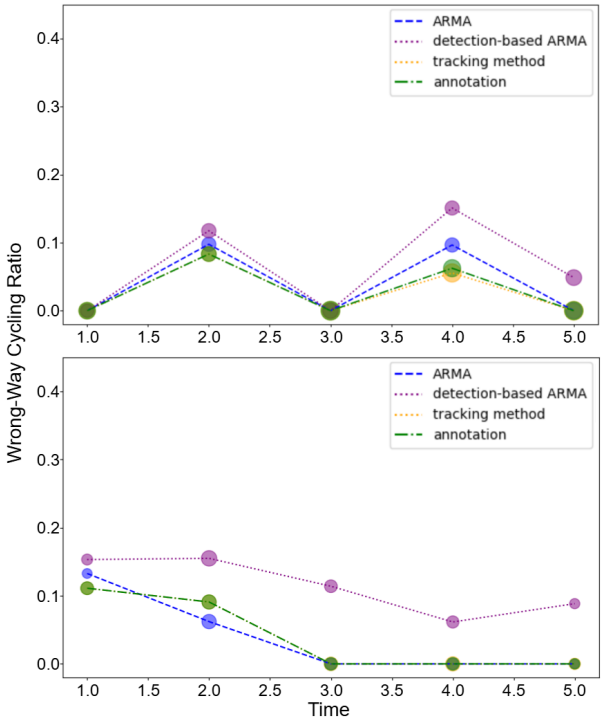

Fig. 4. Minute-level comparison between WWC-Predictor, detection-only WWC-predictor, and tracking-based method (SORT). The size of the circle showcases its scale of numbers.

longer-duration patterns. Furthermore, we annotate instances of wrong-way and right-way cycling every minute, enabling us to assess performance on a minute-by-minute basis. This evaluation strategy allows us to determine the effectiveness and reliability of various methods in accurately and efficiently predicting the wrong-way cycling ratio in road camera videos.

B. Baselines

To comprehensively assess the performance of our proposed WWC-Predictor, we establish a baseline comparison that includes both external tracking methods and a simplified version of our own method. This simplified method isolates the detection component, omitting the ensemble strategy from the full WWC-Predictor. In this detection-only baseline, we compute the orientation (O_{det}) between the center points of the two bounding boxes as the final output, providing a clear comparison to the more complex ensemble approach.

For the tracking component, since we lack annotated tracking data for training end-to-end trackers, we use tracking-by-detection methods. We employ SORT [41], DeepSORT [10], and ByteTrack [42], with ByteTrack representing the state-of-the-art in detection-based tracking.

C. Implementation Details

We employ the YOLOv5-m architecture for the detection module, which strikes a balance between accuracy and computational efficiency, making it suitable for real-time applications. To predict orientation, we utilize a refined Resnet-101 network combined with a Phase-Shifting Coder (PSC) to capture fine-grained spatial features and temporal shifts in the cycling trajectory. Also, we use the same Resnet-101 as a feature extractor in the DeepSORT baseline. We configure the system to sample video frames every 2 (or 4) seconds, ensuring a consistent temporal resolution for accurate tracking. GPU processing time was measured on a single RTX 3080Ti, with serial inference for the detection model and parallel inference for the appearance-based orientation estimation model (in cases processing a single image).

TABLE I
COMPARISON OF OUR METHOD AGAINST OTHERS ON OUR BENCHMARK. "ERROR" REFERS TO THE ABSOLUTE DEVIATION BETWEEN THE PREDICTED RATIO AND THE GROUND TRUTH, WHILE "OVERALL ERROR" REPRESENTS THE MEAN OF THESE FOUR DISCREPANCIES. T_{gap} DENOTES THE TEMPORAL INTERVAL BETWEEN SUCCESSIVE SAMPLES IN THIS METHODOLOGY. "TIME" REFERS TO GPU TIME CONSUMED FOR PROCESSING VIDEO OF ONE MINUTE. CASE 1-3 REFER TO THREE 5-MINUTE VIDEOS, WHILE CASE 4 REFERS TO A 20-MINUTE-LONG VIDEO.

	T_{gap}	Case 1		Case 2		Case 3		Case 4		Overall	Time
		Ratio	Error	Ratio	Error	Ratio	Error	Ratio	Error	Error	
SORT [41]	0.17s	4.2%	-0.3%	2.4%	-0.1%	4.1%	-0.2%	12.2%	+1.0%	0.375%	20.18s
DeepSORT [10]	0.17s	3.8%	-0.7%	2.4%	-0.1%	4.1%	-0.2%	11.5%	+0.3%	0.325%	29.37s
ByteTrack [42]	0.17s	4.3%	-0.2%	2.5%	+0.0%	4.1%	-0.2%	12.1%	+0.9%	0.325%	24.11s
ByteTrack [42]	0.5s	3.0%	-1.5%	7.7%	+5.2%	1.6%	-2.7%	14.3%	+3.1%	3.12%	15.85s
Detection only WWC Predictor (Ours)	2s	11.9%	+7.4%	6.2%	+3.7%	11.8%	+7.5%	17.3%	+6.1%	6.20%	3.83s
WWC Predictor (Ours)	2s	3.9%	-0.6%	3.6%	+1.1%	4.9%	+0.6%	14.8%	+3.6%	1.475%	4.50s
WWC Predictor (Ours)	4s	2.7%	-1.8%	2.1%	-0.4%	3.8%	-0.5%	16.6%	+5.4%	2.025%	2.95s

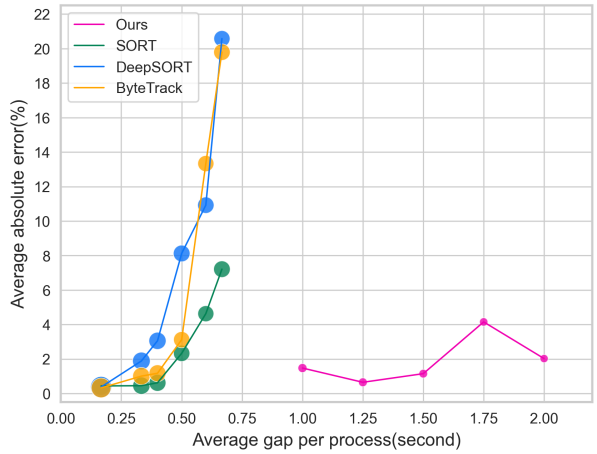

Fig. 5. Processed information scale, algorithm speed and performance comparison between WWC-Predictor, SORT, DeepSORT and ByteTrack. The size of circle showcases algorithm speed.

D. Results

As shown in Table I, the comprehensive WWC-Predictor method exhibits a competitive absolute error rate of 1.475% and a swift inference time of 4.50 seconds per video minute. The detection-only variant of WWC-Predictor demonstrates a higher error margin due to the inherent uncertainty of the detection model, which leads to a significant proportion of True-Negative errors. Conversely, traditional tracking methods showcase a lower error rate owing to their robust frame-to-frame relations and the simplicity of time-dimensional prediction. However, these methods require substantially higher computational resources.

Figure 5 presents a comparative analysis of the WWC Predictor against SORT and DeepSORT across varying time interval scales. It is evident that both SORT and DeepSORT require relatively short time intervals to maintain effective tracking, whereas our WWC Predictor demonstrates robust performance over a specific range of larger intervals.

We annotate instances of wrong-way and right-way cycling every minute, enabling us to assess performance on a minute-by-minute basis. This evaluation strategy allows us to determine the effectiveness and reliability of various methods in accurately and efficiently predicting the wrong-way cycling ratio in road camera videos. Figure 4 presents a minute-level comparison between those three method, which showcases that our method provides a more robust estimation than detection-only model and boosts the performance closer to traditional tracking methods.

E. Ablation Study

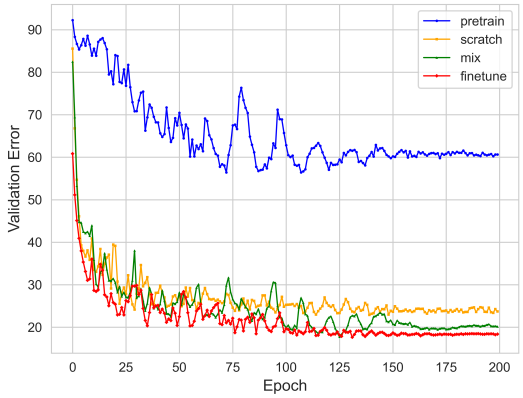

Fig. 6. Comparison of different training strategy for appearance-based orientation estimation model.

1) Ablation on appearance-based orientation estimation model: As described in Section IV-B, we have devised a pretrain-finetune architecture for training an appearance-based orientation estimation model. To evaluate the impact of the pretraining dataset and this specific training architecture, we conducted an ablation study. This study allowed us to assess the effectiveness and significance of these components in our model's performance.

Comparison of the validation error changes resulting from different training methods is presented in Figure 6. The training methods are categorized as follows:

- "Pretrain" denotes training exclusively with synthetic data.
- "Scratch" refers to training solely with real-world data.

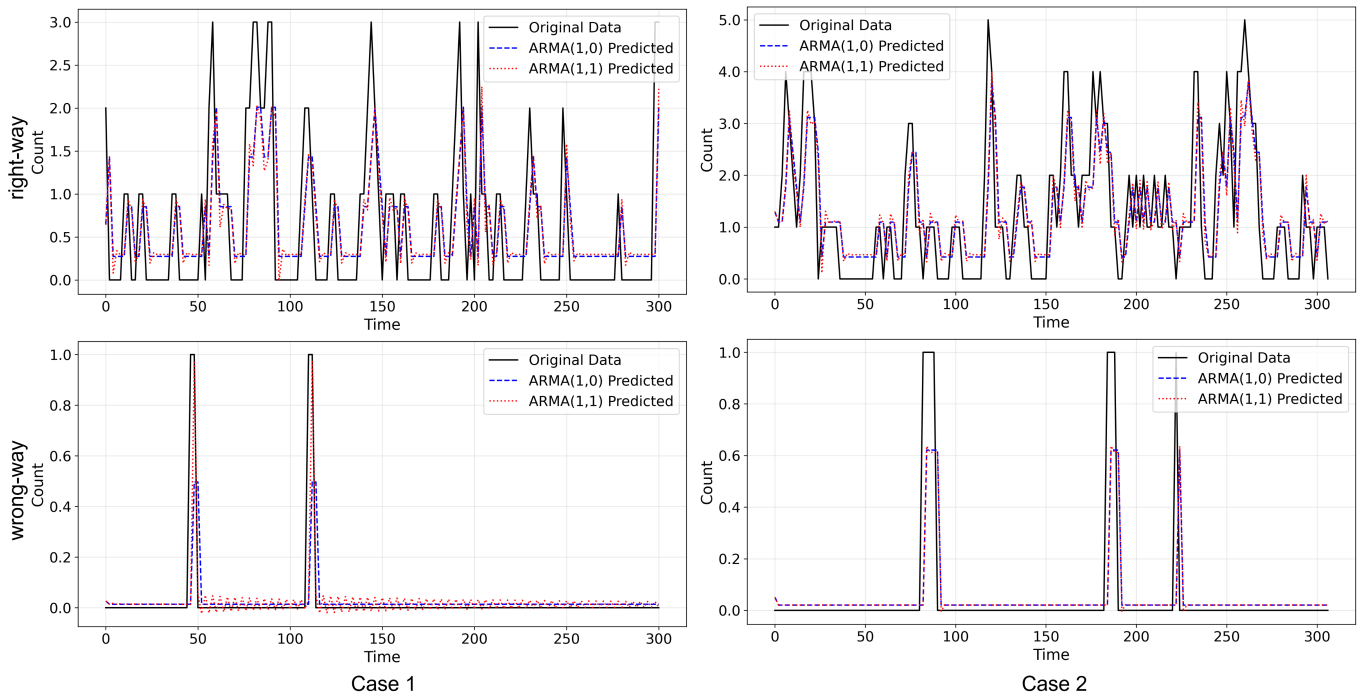

Fig. 7. Comparison between ARMA(1,1) and ARMA(1,0) for wrong-way and right-way data in case 1 and case 2.

TABLE II
STATISTIC RESULTS OF AUTO REGRESSION(AR) PART AND MOVING AVERAGE(MA) PART FOR RIGHT-WAY CYCLING AND WRONG-WAY CYCLING. CASE 1-3 REFER TO THREE 5-MINUTE VIDEOS, WHILE CASE 4 REFERS TO A 20-MINUTE-LONG VIDEO.

	ARMA model for Right-way cycling						ARMA model for Wrong-way cycling					
	AR			MA			AR			MA		
	std err	z	P>|z|	std err	z	P>|z|	std err	z	P>|z|	std err	z	P>|z|
Case 1	0.093	6.143	0.000	0.107	1.832	0.067	0.087	6.583	0.000	0.080	0.504	0.614
Case 2	0.130	3.428	0.001	1.320	1.482	0.138	0.316	-0.015	0.988	0.282	3.512	0.000
Case 3	0.095	5.857	0.000	0.115	1.603	0.109	0.491	-0.146	0.884	0.470	0.751	0.453
Case 4	0.048	10.783	0.000	0.055	3.800	0.000	0.071	6.094	0.000	0.083	-1.021	0.307
Ave	0.092	6.553	0.000	0.399	2.179	0.079	0.241	3.129	0.468	0.229	0.937	0.344

- "Mix" represents training by combining both real-world and synthetic data.
- "Finetune" indicates the process of fine-tuning on real-world data following a pretraining phase with synthetic data.

In our pretrain-finetune architecture, we successfully attain a minimal validation error of 17.63. Moreover, the performance post-convergence significantly surpasses that of training from scratch or employing a mixed dataset. While this error rate might initially appear substantial, it is deemed acceptable for its intended application as a verification model.

F. Experiments of ARMA part

In this section, we conduct a series of experiments to evaluate the performance of the ARMA model in the context of predicting wrong-way cycling occurrences. Figure 7 illustrates a comparative analysis between the ARMA(1,0) and ARMA(1,1) configurations. The notation ARMA(p, q) denotes an ARMA model where p and q are the orders of the autoregressive (AR) process and the moving average (MA) process, respectively.

The analysis, particularly evident in Case 2, suggests that the inclusion of the MA component, which originally intended to capture traffic flow dynamics, does not contribute positively to the predictive accuracy for wrong-way cycling events. This is attributed to the observation that wrong-way cycling incidents rarely exhibit the traffic flow characteristics modeled by the MA component.

Table II presents the respective standard errors, z-values, and p-values for auto regressive (AR) and moving average (MA) terms from ARMA models applied to two different cycling behaviors: Right-Way Cycling and Wrong-way Cycling. In the ARMA model analysis for cycling behaviors, the wrong-way cycling data within "Case 2" exhibits an unusual case where the AR term appears completely non-significant with a p-value of 0.988, indicating a misinterpretation of the auto regressive effect as if it were a moving average component. Generally, for Wrong-way Cycling, the AR terms consistently show non-significance across all categories, suggesting a lack

of moving average characteristic in this behavior. This pattern contrasts with the Right-way cycling data, where AR terms are significant, indicating a reliable auto regressive process.

To address this discrepancy, we incorporate domain knowledge into the model specification process. Consequently, we refine the ARMA model for wrong-way cycling prediction to an ARMA(1,0) configuration, effectively omitting the MA component. This adjustment is predicated on the rationale that the auto regressive component alone is more representative of the underlying process governing wrong-way cycling incidents, thereby enhancing the model's predictive relevance in this particular application.

VII. Discussion

Limitations. WWC-Predictor intentionally forgoes instance-level detection capabilities to optimize computational efficiency for its primary purpose of video-level analysis. While this trade-off enables effective system-wide monitoring of phenomena like wrong-way cycling ratios, it inherently sacrifices granularity for broader operational objectives. Consequently, the approach lacks resolution for fine-grained examination of individual objects or localized interactions, focusing instead on aggregate behavioral patterns across the surveillance network.

Future work. Future research will focus on enhancing cross-camera relationship modeling through graph-structured approaches to better capture traffic network topology. Additionally, significant efforts will address practical deployment challenges by optimizing the framework for edge devices. This includes developing lightweight architectures and efficient computation strategies suitable for real-world implementation in resource-constrained camera networks.

VIII. Conclusion

In this paper, we introduced the new problem of wrong-way cycling ratio prediction in CCTV videos, and proposed a novel method, WWC-Predictor, to tackle this problem by sparse sampling with efficient Two-Frame WWC-Detector and Temporal WWC-Predictor, which has been mathematically and experimentally proven to be effective compared with straightforward tracking methods. Additionally, to facilitate the training and validation of our method for this task, we have presented and open-sourced three datasets to build a convincing benchmark of this task. In our evaluation, our WWC Predictor demonstrates satisfying performance with few computational resource demand.

References

[1] N. Buch, S. A. Velastin, and J. Orwell, "A Review of Computer Vision Techniques for the Analysis of Urban Traffic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 920–939, 2011.

[2] K. Gaur, M. A. Siddique, K. Beernally, N. Madaan, and S. Tarwani, "Real-time wrong-way vehicle detection system with automatic number plate recognition for enhanced road safety," in *2024 International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2024, pp. 1–8.

[3] P. Suttiponpisarn, C. Charnsripinyo, S. Usanavasin, and H. Nakahara, "An Autonomous Framework For Real-time Wrong-way Driving Vehicle Detection from Closed-circuit Televisions," *Sustainability*, vol. 14, no. 16, p. 10232, 2022.

[4] F. H. Shubho, F. Iftekhar, E. Hossain, and S. Siddique, "Real-time traffic monitoring and traffic offense detection using YOLOv4 and OpenCV DNN," in *2021 IEEE Region 10 Conference (TENCON)*, 2021, pp. 46–51.

[5] M. H. Vardhan, K. V. S. Krishna, S. Munappa, and K. A. Manoj, "Wrong Route Vehicles Detection Using Deep Learning," in *2023 International Conference on Next Generation Electronics (NEleX)*, 2023, pp. 1–6.

[6] Z. Rahman, A. M. Ami, and M. A. Ullah, "A Real-time Wrong-way Vehicle Detection Based on YOLO and Centroid Tracking," in *2020 IEEE Region 10 Symposium (TENSYMP)*, 2020, pp. 916–920.

[7] P. Suttiponpisarn, C. Charnsripinyo, S. Usanavasin, and H. Nakahara, "Detection of Wrong Direction Vehicles on Two-way Traffic," in *2021 International Conference on Knowledge and Systems Engineering (KSE)*, 2021, pp. 1–6.

[8] A. Manasa and S. Renuka Devi, "An Enhanced Real-time System for Wrong-Way and Over Speed Violation Detection Using Deep Learning," in *2023 International Conference on Image Processing and Capsule Networks (ICIPCN)*. Springer Nature Singapore, 2023, pp. 309–322.

[9] Y. Yang, "FastMOT: High-performance Multiple Object Tracking Based on Deep SORT and KLT," Nov. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4294717

[10] N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric," in *2017 IEEE International Conference on Image Processing (ICIP)*, Sep 2017. [Online]. Available: http://dx.doi.org/10.1109/icip.2017.8296962

[11] K. Saho, "Kalman Filter for Moving Object Tracking: Performance Analysis and Filter Design," 2017. [Online]. Available: https://doi.org/10.5772/intechopen.71731

[12] M. Chaabane, P. Zhang, J. R. Beveridge, and S. O'Hara, "DEFT: Detection Embeddings for Tracking," *arXiv preprint arXiv:2102.02267*, 2021.

[13] P. Chu, J. Wang, Q. You, H. Ling, and Z. Liu, "TransMOT: Spatial-Temporal Graph Transformer for Multiple Object Tracking," in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2023, pp. 4859–4869.

[14] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the Fairness of Detection and Re-identification in Multiple Object Tracking," *International Journal of Computer Vision*, vol. 129, no. 11, p. 3069–3087, Nov. 2021. [Online]. Available: https://doi.org/10.1007/s11263-021-01513-4

[15] Q. Wang, Y.-Y. Chang, R. Cai, Z. Li, B. Hariharan, A. Holynski, and N. Snavely, "Tracking everything everywhere all at once," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 19 738–19 749.

[16] C. Doersch, Y. Yang, M. Vecerik, D. Gokay, A. Gupta, Y. Aytar, J. Carreira, and A. Zisserman, "TAPIR: Tracking Any Point with per-frame Initialization and temporal Refinement," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 10 027–10 038.

[17] P. Suttiponpisarn, C. Charnsripinyo, S. Usanavasin, and H. Nakahara, "An Enhanced System for Wrong-Way Driving Vehicle Detection with Road Boundary Detection Algorithm," *Procedia Computer Science*, vol. 204, pp. 164–171, 2022, 2022 International Conference on Industry Sciences and Computer Science Innovation (iSCSi). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187705092200758X

[18] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed And Accuracy of Object Detection," *arXiv preprint arXiv:2004.10934*, 2020.

[19] R. Choudhari, S. Goel, Y. Patel, and S. Ghane, "Traffic rule violation detection using detectron2 and yolov7," in *2023 World Conference on Communication & Computing (WCONF)*, 2023, pp. 1–7.

[20] W. Gu, Z. Zhou, Y. Zhou, H. Zou, Y. Liu, C. J. Spanos, and L. Zhang, "BikeMate: Bike Riding Behavior Monitoring with Smartphones," ser. MobiQuitous 2017. Association for Computing Machinery, 2017, p. 313–322. [Online]. Available: https://doi.org/10.1145/3144457.3144462

[21] H. Hayashi, A. Xu, Z. Zhou, and K. Yatani, "Vision-based Scene Analysis toward Dangerous Cycling Behavior Detection Using Smartphones," in *2021 Adjunct Proceedings of the 2021 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2021 ACM International Symposium on Wearable Computers (UbiComp/ISWC)*, ser. UbiComp/ISWC '21 Adjunct. Association for Computing Machinery, 2021, p. 28–29. [Online]. Available: https://doi.org/10.1145/3460418.3479300

[22] N. Dhakal, C. R. Cherry, Z. Ling, and M. Azad, "Using CyclePhilly data to assess wrong-way riding of cyclists in Philadelphia," *Journal of Safety Research*, vol. 67, pp. 145–153, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022437517307338

[23] X. Xie, G. Cheng, J. Wang, X. Yao, and J. Han, "Oriented R-CNN for Object Detection," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2021. [Online]. Available: http://dx.doi.org/10.1109/iccv48922.2021.00350

[24] J. Han, J. Ding, N. Xue, and G.-S. Xia, "ReDet: A Rotation-equivariant Detector for Aerial Object Detection," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2021. [Online]. Available: http://dx.doi.org/10.1109/cvpr46437.2021.00281

[25] L. Wen, Y. Cheng, Y. Fang, and X. Li, "A comprehensive survey of oriented object detection in remote sensing images," *Expert Systems with Applications*, vol. 224, p. 119960, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423004621

[26] Y. Yu and F. Da, "Phase-shifting coder: Predicting accurate orientation in oriented object detection," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 13 354–13 363.

[27] M. Tian, C. Sun, and S. Wu, "An EMD and ARMA-based Network Traffic Prediction Approach in SDN-based Internet of Vehicles," *Wireless Networks*, pp. 1–13, 2021.

[28] J. Peng, Y. Xu, and M. Wu, "Short-term Traffic Flow Forecast Based on ARIMA-SVM Combined Model," in *2021 International Conference on Green Intelligent Transportation System and Safety*. Springer, 2021, pp. 287–300.

[29] J. H. Giraldo, S. Javed, M. Sultana, S. K. Jung, and T. Bouwmans, "The emerging field of graph signal processing for moving object segmentation," in *Frontiers of Computer Vision*, H. Jeong and K. Sumi, Eds. Springer International Publishing, 2021, pp. 31–45.

[30] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[31] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Autoregressive Moving Average Graph Filtering," *IEEE Transactions on Signal Processing*, vol. 65, no. 2, p. 274–288, Jan. 2017. [Online]. Available: http://dx.doi.org/10.1109/TSP.2016.2614793

[32] G. Leus, A. G. Marques, J. M. Moura, A. Ortega, and D. I. Shuman, "Graph signal processing: History, development, impact, and outlook," *IEEE Signal Processing Magazine*, vol. 40, no. 4, pp. 49–60, 2023.

[33] G. Jocher, "YOLOv5 by Ultralytics," May 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2016.90

[35] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[36] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Springer International Publishing, 2016, pp. 501–518.

[37] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4104–4113.

[38] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding," *ACM Transactions on Graphics*, p. 1–15, Jul 2022. [Online]. Available: http://dx.doi.org/10.1145/3528223.3530127

[39] J. Perktold, S. Seabold, K. Sheppard, ChadFulton, K. Shedden, jbrockmendel, j grana6, P. Quackenbush, V. Arel-Bundock, W. McKinney, I. Langmore, B. Baker, R. Gommers, yogabonito, s scherrer, Y. Zhurko, M. Brett, E. Giampieri, yl565, J. Millman, P. Hobson, Vincent, P. Roy, T. Augspurger, tvanzyl, alexbrc, T. Hartley, F. Perez, Y. Tamiya, and Y. Halchenko, "statsmodels/statsmodels: Release 0.14.1," Dec. 2023. [Online]. Available: https://doi.org/10.5281/zenodo.10378921

[40] A. I. McLeod and Y. Zhang, "Faster ARMA maximum likelihood estimation," *Computational Statistics and Data Analysis*, vol. 52, no. 4, p. 2166–2176, Jan. 2008. [Online]. Available: https://doi.org/10.1016/j.csda.2007.07.020

[41] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple Online and Realtime Tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sep 2016. [Online]. Available: http://dx.doi.org/10.1109/icip.2016.7533003

[42] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "ByteTrack: Multi-object Tracking by Associating Every Detection Box," in *2022 European Conferenceon Computer Vision (ECCV)*. Springer-Verlag, 2022, p. 1–21. [Online]. Available: https://doi.org/10.1007/978-3-031-20047-2_1