

Article

Cross-Driver Distracted Driving Behavior Recognition with Directional Enhancement and Exponential Weight Smoothing

Yizhuo Zhang ¹, Chengming Chen ^{1,*}, Zhuo He ¹ and Xiaoyi Zhou ^{2,3}

¹ College of Engineering Science and Technology, Shanghai Ocean University, Shanghai 201306, China; 2350120@st.shou.edu.cn (Y.Z.); 2325123@st.shou.edu.cn (Z.H.)

² National Key Laboratory of Human Factors Engineering, Astronaut Research and Training Center, Tsinghua University, Beijing 100094, China; stevenzxy1015@163.com

³ China Institute of Marine Technology & Economy, 70 Xueyuan South Rd., Beijing 100081, China

* Correspondence: cmchen@shou.edu.cn

Abstract

Driver distraction recognition is important for in-cabin driver monitoring, but random image splits can overestimate performance because driver identity, in-vehicle background, and scene cues may be shared between training and evaluation data. To improve recognition under the executed driver-disjoint evaluation, this paper proposes Directional Enhancement and Weight Smoothing EfficientNet (DEWS-EfficientNet), a single-frame classification framework built on EfficientNet-B0. The method integrates label-consistent flipping (LCF), a directional enhancement (DE) Block, and exponential weight smoothing (EWS). LCF swaps left- and right-hand phone/texting labels after horizontal flipping. The DE Block uses 3×3 depthwise and $1 \times 5/5 \times 1$ asymmetric depthwise convolutions to model local actions and directional patterns. EWS uses smoothed weights for validation and testing without adding an inference branch. Experiments are conducted on AUC Distracted Driver V2 (AUC2) and State Farm Distracted Driver Detection (SFD3) under driver-disjoint splits. DEWS-EfficientNet achieves the best accuracies of 82.39% and 89.00% on AUC2 and SFD3, improving the EfficientNet-B0 baseline by 11.83 and 10.04 percentage points, respectively. Ablation and visualization results further indicate that the proposed components improve recognition performance under the executed driver-disjoint splits while preserving a compact classification pipeline.

Keywords: driver distraction detection; cross-driver recognition; directional enhancement; label-consistent augmentation; exponential weight smoothing

Academic Editor: Giorgio Prevati

Received: 30 June 2026

Revised: 29 July 2026

Accepted: 31 July 2026

Published: 3 August 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

According to the World Health Organization, road traffic injuries remain a major global public safety concern. Distracted driving diverts a driver's visual, cognitive, or manual resources away from safe driving and typically includes phone use, texting, eating or drinking, operating in-vehicle devices, reaching behind, and talking to passengers. Susanto et al. conducted two 2^2 factorial driving-simulator experiments with 20 licensed drivers, crossing rural and urban routes with phone-call and texting tasks in hands-free and hand-held modes. Mobile-phone tasks significantly affected mental workload, error frequency, average speed, and lateral-position changes. Texting generally

produced greater impairment, while the interaction between traffic environment and hand-held use affected error frequency [1]. Piancastelli decomposed touchscreen interaction into visual reorientation, decision, pointing, actuation, and refocusing phases and compared the numerical estimates with eye-gaze evidence. Measured interactions, particularly multistep navigation and address-entry tasks, required substantially longer visual engagement than idealized models predicted and therefore produced considerable blind-driving distances [2]. Because these behaviors are expressed through coupled changes in hand position, head orientation, body posture, and in-vehicle interaction regions, reliable fine-grained recognition is an important component of driver-state monitoring.

In recent years, deep learning has become the dominant technical route for driver distraction recognition. Convolutional neural networks (CNNs) can automatically learn action representations from in-cabin images, and constrained attention, coordinate attention, and channel attention further strengthen the use of discriminative regions [3–5]. Hard spatial attention and lightweight detection frameworks improve recognition from the perspectives of key-region modeling and real-time deployment [6–8]. These studies show that the hands, face, and center-console regions are important cues for distracted behavior, but using these cues robustly on unseen drivers remains challenging.

Another key challenge is the evaluation protocol. If adjacent or similar images from the same driver are randomly assigned to both training and test sets, the model may exploit identity-related cues such as clothing, seat background, camera position, and lighting, leading to overoptimistic estimates. Driver-disjoint splitting provides a stricter evaluation of recognition ability on unseen drivers [9–12]. Therefore, this paper uses driver-disjoint training, validation, and test sets on both AUC2 and SFD3, selects checkpoints by validation Accuracy, and reports the best results according to the table protocol.

Existing studies have explored this problem from key-region constraints, lightweight networks, detection frameworks, and cross-domain generalization. Hard spatial attention methods have verified the importance of local regions such as the hands and head for in-cabin action recognition [6]. Lightweight You Only Look Once (YOLO) detectors and DDR-YOLO improve the real-time detection of distracted driving behavior [7,8], whereas multiscale domain adaptation and domain-generalization methods emphasize the influence of driver, background, and viewpoint differences on practical deployment [9,10].

Beyond driver variation, the classes themselves contain directional semantics and local appearance ambiguities. Left- and right-hand phone use or texting require corresponding label changes after horizontal flipping. Phone use, hair or makeup, and reaching behind may also share visual patterns in which the hand approaches the face or torso. With only global classification supervision, a model can be affected by background and identity bias and may fail to consistently model the directional relation among the hand, head, and in-vehicle regions.

For this reason, simply stacking complex attention modules or adding detection branches is not necessarily the most reliable choice. Complex constraints increase the number of training objectives and the difficulty of parameter tuning, whereas detection-based methods often require additional region annotations or more complex post-processing. For single-frame driver-behavior classification, a preferable strategy is to keep the backbone classification framework simple while making targeted improvements in data augmentation, local directional modeling, and training stability.

Asymmetric convolution is an established operator rather than a new primitive introduced here. Inception replaces a square spatial kernel with sequential one-dimensional kernels to reduce computation [13], and ERFNet embeds such factorization

inside residual units for efficient dense prediction [14]. ACNet instead trains parallel square, horizontal, and vertical branches to strengthen a generic kernel skeleton and then algebraically reparameterizes them into one square convolution for deployment [15]. The proposed directional enhancement (DE) Block has a different objective and deployment form: it does not replace a backbone convolution or collapse its branches. Instead, it preserves an independent 3×3 local path and a sequential $1 \times 5 \times 5 \times 1$ directional path with intervening nonlinearities, concatenates their complementary responses, mixes channels, and learns a scaled residual correction of high-level driver-behavior features. Because the nonlinear directional path cannot be merged algebraically into the local kernel, the two-path relation remains active during inference. The contribution therefore lies in task-specific local-directional feature refinement for hand-face, hand-console, and arm-extension relations rather than in proposing a new asymmetric kernel.

To address these issues, we propose DEWS-EfficientNet with EfficientNet-B0 [16] as the backbone. At the data level, label-consistent flipping (LCF) applies a deterministic class permutation after horizontal flipping, preserving left-right behavior semantics without adding parameters or inference cost. At the feature level, the DE Block differs from learned channel/spatial attention and global Transformer self-attention [3–5,11,12]: it imposes an explicit local horizontal-vertical inductive bias through a parallel 3×3 local path and a nonlinear $1 \times 5 \times 5 \times 1$ directional path, followed by lightweight residual correction. At the parameter level, exponential weight smoothing (EWS) averages successive optimization states to reduce checkpoint fluctuation without changing the network topology or adding an inference branch. The three components therefore address label consistency, directional feature refinement, and parameter stability at complementary stages while retaining a single-branch classifier. We evaluate the method on AUC2 and SFD3 using backbone comparisons, a complete 2^3 factorial ablation, efficiency measurements, confusion matrices, and Grad-CAM visualizations. The remainder of this paper is organized as follows: Section 2 reviews representative methods, Section 3 presents DEWS-EfficientNet, Section 4 reports the experimental design and results, and Section 5 summarizes the findings, limitations, and future work.

The main contributions of this paper are as follows:

- (1) We propose label-consistent flipping (LCF), which swaps the labels of left- and right-hand phone-use and texting classes during horizontal flipping. This preserves the diversity benefit of flipping while avoiding the practical problem of supervising a left-hand action as a right-hand action.
- (2) We develop a task-oriented DE Block that organizes established asymmetric operators as an independent local path and a nonlinear directional path for high-level driver-behavior refinement. Concatenation, channel mixing, and zero-initialized residual correction strengthen hand-face, hand-console, and arm-extension cues while limiting computational overhead. Used alone, the block improves Accuracy by 5.18 percentage points on AUC2 and 3.69 points on SFD3; its practical role is therefore to reduce fine-grained action ambiguity without introducing a detector, attention branch, or additional annotation requirement.
- (3) We introduce exponential weight smoothing (EWS), so that validation and testing use smoothed model weights. EWS does not change the network topology or add inference branches but instead reduces the influence of late-training checkpoint fluctuation on cross-driver evaluation.
- (4) The effectiveness of the proposed method is systematically verified on two driver distraction datasets, AUC2 and SFD3. The experiments include comparisons across different backbone networks, EfficientNet-B0 ablation studies, training-process analysis, confusion matrices, and Grad-CAM visualizations. The results show that DEWS improves the recognition performance of different backbone models without introducing an

additional inference branch while also enhancing the discrimination of easily confused classes and showing good potential for lightweight deployment.

2. Related Work

Research on driver distraction detection has developed along several routes, including single-frame image classification, lightweight feature representation, detection-based key-region modeling, explicit behavioral cues, temporal modeling, and multimodal fusion. These methods differ substantially in input information, model structure, evaluation protocol, and deployment conditions. To clarify the technical position of the present work, this section reviews representative studies in terms of authors, methods, reported results, advantages, and limitations.

2.1. Single-Frame Image Classification and Lightweight Feature Representation

Single-frame image classification takes in-cabin camera images as input and directly predicts driver behavior from hand position, head orientation, body posture, and in-vehicle interaction regions. It is currently one of the most common routes for distraction recognition. John et al. proposed LMF-FPCNN, which combines a fast pre-convolutional network with a lightweight multiscale fusion module to decompose, reconstruct, and aggregate features at different scales. Their experiments reported an Accuracy of 97.8% while reducing computational complexity and training time [17]. The method captures both global driver posture and local hand details and is suitable for resource-constrained scenarios. However, the multiscale preprocessing and fusion structure may still depend on a specific data distribution, and cross-driver or cross-scene stability was only limitedly validated.

Mohsin et al. proposed ResNet-RG, which combines grayscale input, regularization, and attention enhancement within a residual network to highlight gaze shifts, head movement, and hand actions. The model achieved 99.91% Accuracy on State Farm and 96.71% on a self-collected dataset with more environmental variation [18]. Residual connections help stabilize deep network training, and grayscale adaptation can reduce interference from color variation. Nevertheless, the model still relies on static visual cues, and its generalization to occlusion, subtle actions, and unseen drivers depends on the data split. Hassam et al. proposed AHDCNN, which uses an autoencoder to compress redundant features and combines attention, dropout, and a Softmax classifier. Experiments on State Farm showed high recognition Accuracy with relatively low computational cost [19]. Its advantage lies in combining dimensionality reduction with discriminative-region enhancement, but the validation mainly focused on a single dataset.

Luo et al. proposed KRCA, which transfers hard labels, soft labels, sample relations, and intermediate features from a teacher to a lightweight student. The optimized SqueezeNet_LW achieved 99.83% Accuracy [20]. KRCA improves a small model without increasing its inference structure, although training still requires a teacher network and additional distillation objectives.

Using a random image-level split, Eraqi et al. were among the early studies to apply an ensemble of CNNs to driver distraction recognition and improved State Farm classification by fusing several convolutional predictors [21]. The work established the effectiveness of ensemble learning for single-frame recognition, although multi-model inference increases computational and deployment costs.

Baheti et al. proposed MobileVGG, a compressed VGG-style convolutional network for real-time distracted driver detection, reporting high Accuracy on the State Farm and AUC datasets [22]. Its advantage is a simple and fast structure, but it still mainly relies on static-image classification supervision.

Qin et al. proposed D-HCNN, a hierarchical CNN with decreasing filter sizes that improves distraction classification while retaining a low parameter count [23]. Its SFD3 result uses an image-level split, whereas the reported AUC2 result follows the source-defined partition. The method shows that CNN architecture design can improve lightweight recognition, but direct numerical comparison still requires consistent evaluation protocols.

Liu et al. proposed a distillation-based neural architecture search and knowledge-transfer framework for extremely lightweight distracted driver recognition [24]. Li et al. further fused CNN local features with Vision Transformer (ViT) global modeling to improve the representation of complex in-cabin actions [25]. These studies enrich the lightweight-model route, but search, distillation, or Transformer branches increase training and deployment complexity.

2.2. Detection-Based Models and Key-Region Modeling

Detection-based methods improve spatial interpretability by localizing phones, hands, or other regions related to distracted behavior. Gu et al. proposed ATD2Net, using channel reconstruction awareness, dynamic spatial modeling, and structural feedback optimization to handle small and easily occluded targets. The reported results show that the model outperformed recent comparison methods while remaining lightweight and in real time [26]. Its advantage is the balance among small-object detection accuracy, parameter size, and speed, but it requires object-level annotations and may still miss completely occluded targets. Rampavan et al. used an improved genetic algorithm to search the YOLO architecture and training parameters, obtaining a smaller model with better performance than the compared detection models [27]. Automatic search reduces manual trial and error, but the search process is computationally expensive and may overfit the search dataset.

Debsi et al. proposed ME-YOLOv8, which incorporates multiscale feature enhancement and attention into YOLOv8 for joint detection of driver distraction and fatigue. Experiments showed that the added modules improved detection accuracy on the related datasets while maintaining the efficiency of the real-time framework [28]. The method can process multiple abnormal states, but its architecture and annotation pipeline are more complex than those of pure classification methods. Xin et al. proposed a deep learning eye-tracking system that associates driver gaze position with road target regions to determine whether attention has shifted away from critical road areas [29]. This method offers clear interpretation for visual distraction, but it depends on stable eye imaging and gaze calibration and covers non-visual or hand-dominated distractions less well.

Lu et al. introduced deformable and dilated convolutions and optimized region proposals into Faster R-CNN to accommodate arm-pose changes and non-rigid actions [30]. Deformable sampling improves adaptation to action-shape variation, but the two-stage detector remains computationally expensive. Mittal and Verma proposed CAT-CapsNet, which uses attention to select discriminative features and capsule routing to preserve part-pose relations. Under random image-level splits, it achieved 99.88% Accuracy on State Farm and 96.7% on AUC [31]. Its fine-grained pose modeling is strong, although routing and attention increase training and inference complexity.

2.3. Explicit Behavioral Cues, Temporal Information, and Multimodal Fusion

Explicit-cue methods describe driver states using posture, gaze, or vehicle dynamics. Gong et al. used pose estimation to extract upper-body keypoints and proposed an adjustable-distance weighted KNN classifier, achieving a 94.04% recognition rate with fast execution on their dataset [32]. The method is interpretable and lightweight, but keypoint quality is sensitive to occlusion and illumination. Tango and Botta compared SVM, neural

networks, and ANFIS using vehicle dynamics and control signals, and SVM performed best among the compared classifiers [33]. This route avoids photographing the driver and better protects visual privacy, but vehicle dynamics are influenced by road conditions, vehicle type, and individual driving style, making it difficult to distinguish specific distracted behaviors.

Liu et al. combined eye-movement and head-movement features with semi-supervised machine learning to identify cognitive distraction, showing that a small amount of labeled data together with unlabeled samples can improve state classification [34]. The approach reduces annotation demand, but it depends on stable eye-movement features. Aksjonov et al. combined machine learning prediction with fuzzy logic to continuously evaluate abnormal driving, and ANFIS performed well among the compared prediction methods [35]. Fuzzy rules can express state uncertainty, but rule sets and membership functions require manual design and may need recalibration when transferred to different vehicles or driving scenarios.

To identify cognitive states that are difficult to observe from a single frame, Liu et al. proposed a two-level fusion CNN that combines ECG and fNIRS information at the feature and decision levels. The results showed that two-level fusion outperformed non-fusion, feature-level fusion alone, and decision-level fusion alone [36]. Its advantage is the use of complementary physiological information, whereas wearable devices and synchronized acquisition limit everyday deployment. Noh et al. proposed BiViT, which fuses representations from a physiological-signal Transformer and a visual Transformer, achieving 0.91 Accuracy and 0.91 F1-score and outperforming the single-modality alternatives [37]. The method provides more complete state awareness, but sensor cost, synchronization, and missing modalities remain important issues.

Mou et al. proposed a CNN-Transformer dual-channel model that combines local convolutional patterns with global or cross-modal dependencies; its random segment-level evaluation verified the benefit of dual-channel fusion [38], although optimization and deployment are more complex than for ordinary CNNs. Elhenawy et al. combined pretrained networks, transfer learning, and temporal-window features in naturalistic driving videos and evaluated held-out trips without establishing unseen-driver testing [39]. Temporal context supports continuous behavior recognition, but frame buffering, latency, and storage costs increase.

Recent vision-language and Transformer models have extended distracted driver recognition beyond conventional CNNs. Hasan et al. showed that vision-language models can recognize distracted behavior in naturalistic videos, although they require more data and computation [40]. Zhang et al. used Swin Transformer window attention to improve contextual feature interaction on public datasets [41]. These results demonstrate strong contextual modeling, but lightweight deployment and evaluation on unseen drivers remain important concerns. The main models are summarized in Table 1.

Table 1. Comparison of representative driver distraction methods and literature-reported results under their original evaluation protocols. Protocol labels distinguish random/image-level, driver-disjoint, subject-dependent/within-subject, segment-level, and trip-level settings.

Author [Citation]	Model	Method	Evaluation Protocol	Advantages	Limitations
John et al. [17]	LMF-FPCNN	Fast pre-convolution and lightweight multiscale fusion	Random/image level	Combines global posture and local details	Limited cross-scene validation
Mohsin et al. [18]	ResNet-RG	Grayscale adaptation, residual learning, and attention	Driver disjoint (leave-N-driver-out)	Relatively robust and compact	Relies on static visual cues

Hassam et al. [19]	AHDCNN	Autoencoder compression, attention, and dropout	Random/image level (70:30)	Compresses redundancy and enhances key regions	Mainly validated on a single dataset
Luo et al. [20]	KRCA	Multilevel knowledge distillation and relation transfer	Random/image level (75:25)	Improves lightweight student networks	Requires a teacher model during training
Baheti et al. [22]	MobileVGG	Lightweight VGG-style convolutional network	Random/image level	Simple structure and good real-time performance	Mainly relies on static-image classification supervision
Liu et al. [24]	NAS-KT	Distillation-based neural architecture search and knowledge transfer	Random/image level	Balances compression and supervised transfer	Search and teacher supervision increase training complexity
Li et al. [25]	CNN-ViT fusion	Fusion of CNN local features and ViT global modeling	Random/image level	Combines local texture and long-range dependency	Transformer branch increases deployment cost
Gu et al. [26]	ATD2Net	Tiny-object detection and dynamic spatial modeling	Random/image level	Balances accuracy, speed, and parameter size	Requires object-level annotations
Rampavan et al. [27]	GA-YOLO	Genetic search of YOLO structure and parameters	Random/image level	Reduces manual architecture tuning	Search cost is relatively high
Debsi et al. [28]	ME-YOLOv8	Multiscale enhancement and attention-based detection	Random/image level (8:1:1)	Supports real-time multi-state detection	More complex structure and annotation pipeline
Xin et al. [29]	Eye-tracking system	Deep eye tracking and road-region association	Random/image level	Clear spatial interpretation	Depends on eye imaging and calibration
Mittal and Verma [31]	CAT-CapsNet	Convolution, attention, and capsule routing	Random/image level (multiple ratios)	Strong fine-grained pose-relation modeling	Higher training and inference complexity
Gong et al. [32]	OpenPose + ADW-KNN	Upper-body keypoints and weighted KNN	Random/image level (8:2)	Fast and interpretable	Sensitive to keypoint quality
Tango and Botta [33]	SVM-based system	Vehicle-dynamics features and machine learning	Random/image level	Non-contact and privacy preserving	Affected by road and driving style
Aksjonov et al. [35]	ML + fuzzy logic	Machine learning prediction and fuzzy evaluation	Subject dependent/within subject	Provides continuous risk estimation	Rules and memberships need calibration
Liu et al. [36]	Two-level fusion CNN	ECG and fNIRS feature/decision fusion	Sample-level stratified	Uses complementary physiological information	Requires wearable devices
Noh et al. [37]	BiViT	Fusion of visual and physiological Transformers	Random/sample level (80:20)	More complete state awareness	Synchronization and missing modalities are challenging
Mou et al. [38]	CNN-Transformer	Dual-channel local convolution and global dependency modeling	Random segment level (80:10:10)	Combines local and global features	More complex optimization and deployment
Hasan et al. [40]	Vision-language model	Vision-language recognition of naturalistic driving videos	Driver disjoint (leave-driver-out)	Strong semantic interpretability	Higher data and inference cost
Zhang et al. [41]	Swin Transformer	Window attention and discriminative feature learning	Random/image level	Strong contextual modeling	Lightweight deployment still requires optimization

2.4. Cross-Driver Generalization and Summary

Overall, existing methods have made clear progress in classification accuracy, local-object localization, interpretability, and multisource information fusion. Classification models are usually structurally simple and easy to deploy, but they are susceptible to driver identity, in-cabin background, and local occlusion. Detection models can explicitly model regions such as phones, hands, and faces, but they usually require additional annotations or more complex post-processing. Pose, gaze, physiological-signal, and video models can provide information that a single image cannot express, but they also introduce device, synchronization, and real-time costs. Moreover, different papers use different data splits, numbers of drivers, and experimental scenarios, so a high same-distribution accuracy does not necessarily indicate stable recognition on unseen drivers.

Based on this analysis, this paper restricts the study to single-frame in-cabin images and a driver-disjoint evaluation protocol, without using bounding boxes, physiological sensors, or video inference branches. The proposed method uses LCF to preserve left–right class semantics, the DE Block to supplement high-level features with horizontal and vertical action patterns, and EWS to reduce late-training parameter fluctuation. The scheme is not intended to replace detection-based or multimodal methods; rather, it targets resource-constrained classification scenarios that require cross-driver transfer and balances structural complexity, training stability, and behavioral discriminability.

3. Method

3.1. Overall Framework

The proposed DEWS-EfficientNet consists of a shared classification forward path and training-stage auxiliary strategies, as shown in Figure 1. The input image is processed by LCF and then passed into the EfficientNet-B0 backbone. A DE Block is inserted after the high-level semantic features of Stage 4, followed by a 1×1 convolution, global average pooling (GAP), Dropout, and a fully connected layer to output the probabilities of 10 behavior classes. During training, both online parameters and EWS-smoothed parameters are maintained. During validation and testing, the smoothed parameters are used in the same forward path. The dashed region in Figure 1 denotes the complete topology shared by training and inference; LCF and EWS do not introduce additional inference branches.

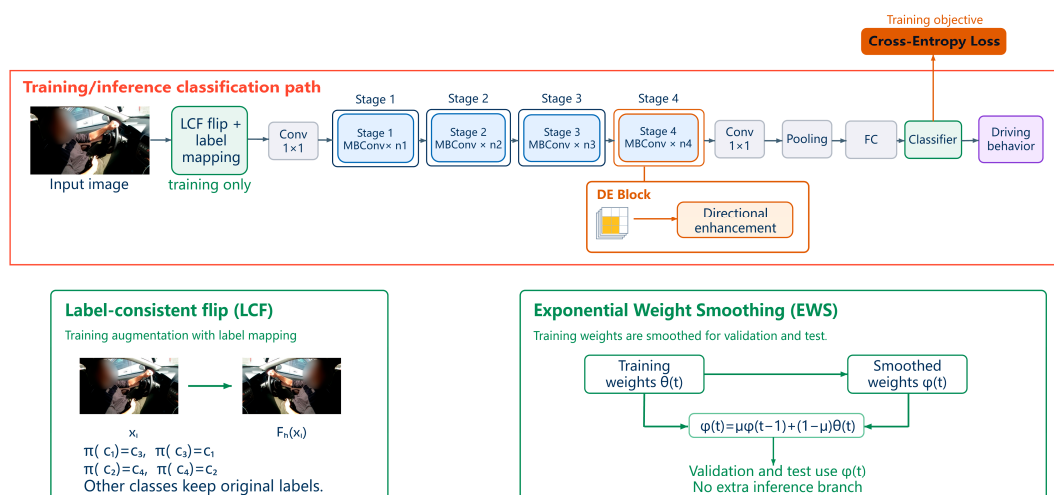


Figure 1. Overall training and inference framework of DEWS-EfficientNet. LCF is applied before the backbone during training; EWS updates smoothed weights after parameter optimization.

This design places augmentation, directional feature extraction, and parameter smoothing at the input, feature, and optimization levels, respectively. LCF only changes training samples and their labels, the DE Block performs residual enhancement at a single high-level location, and EWS only determines which parameter copy is used for validation and testing. Therefore, the complete model still executes one standard forward pass during inference.

3.2. Label-Consistent Flipping

In driver distraction datasets, left- and right-hand phone use and texting classes have explicit directional semantics. Ordinary horizontal flipping can increase sample diversity, but if the original label is retained after flipping, the image content and supervision become inconsistent. Therefore, this paper applies LCF to keep flipped images and labels aligned. For the i -th training sample, LCF is expressed as:

$$x'_i = F_h(x_i), \quad y'_i = \pi(y_i) \quad (1)$$

where x_i and y_i denote the original image and its class label, F_h denotes horizontal flipping, and π denotes the label-mapping function during flipping. For the 10 driver behaviors in AUC2 and SFD3, only left-right hand-related classes are swapped:

$$\pi(c_1) = c_3, \quad \pi(c_3) = c_1 \quad (2)$$

$$\pi(c_2) = c_4, \quad \pi(c_4) = c_2 \quad (3)$$

$$\pi(c_k) = c_k, \quad k \notin \{1, 2, 3, 4\} \quad (4)$$

Here, c_1 and c_3 correspond to the left- and right-hand texting classes, and c_2 and c_4 correspond to the left- and right-hand phone-use classes. All other classes retain their original labels after flipping. This strategy preserves the benefit of data augmentation while reducing supervision noise introduced by incorrect augmented labels.

3.3. DE Block: Directional Enhancement Block

Driver distraction classes depend on spatial relations among the hands, face, torso, center console, and rear-seat region. The DE Block uses asymmetric convolution as an established operator to model horizontal and vertical action context. Unlike Inception or ERFNet factorization and ACNet reparameterization [13–15], its 1×5 and 5×1 depthwise convolutions remain a nonlinear directional path in the inference graph and are fused with an independent 3×3 local path through concatenation and a 1×1 convolution. This cascade has a nominal 5×5 support with 10 coefficients per channel rather than 25 for a full 5×5 depthwise kernel, while channel mixing and a zero-initialized residual correction refine high-level behavior features without replacing the backbone convolution. The DE Block architecture is illustrated in Figure 2.

Let the input high-level feature be X . The local branch extracts neighborhood detail through 3×3 depthwise convolution:

$$Bl = \text{SiLU}(\text{BN}(\text{DWConv}_{3 \times 3}(X))) \quad (5)$$

The directional branch uses 1×5 and 5×1 asymmetric depthwise convolutions in sequence to model action responses along the horizontal and vertical directions:

$$Bd = \text{SiLU}(\text{BN}(\text{DWConv}_{5 \times 1}(\text{SiLU}(\text{BN}(\text{DWConv}_{1 \times 5}(X))))) \quad (6)$$

The outputs of the two branches are concatenated along the channel dimension and fused by a 1×1 convolution:

$$M = \text{SiLU}(\text{BN}(\text{Conv}_{1 \times 1}(\text{Concat}(Bl, Bd)))) \quad (7)$$

Finally, the module produces enhanced features in a residual form:

$$Y = X + \gamma M \quad (8)$$

where Y denotes the enhanced output feature, and γ is a learnable residual scaling coefficient initialized to 0. This makes the module approximate an identity mapping at the beginning of training and then gradually learn an appropriate directional-enhancement strength. The DE Block does not introduce an additional directional gate or dilated branch, keeping the structure simple and training stable.

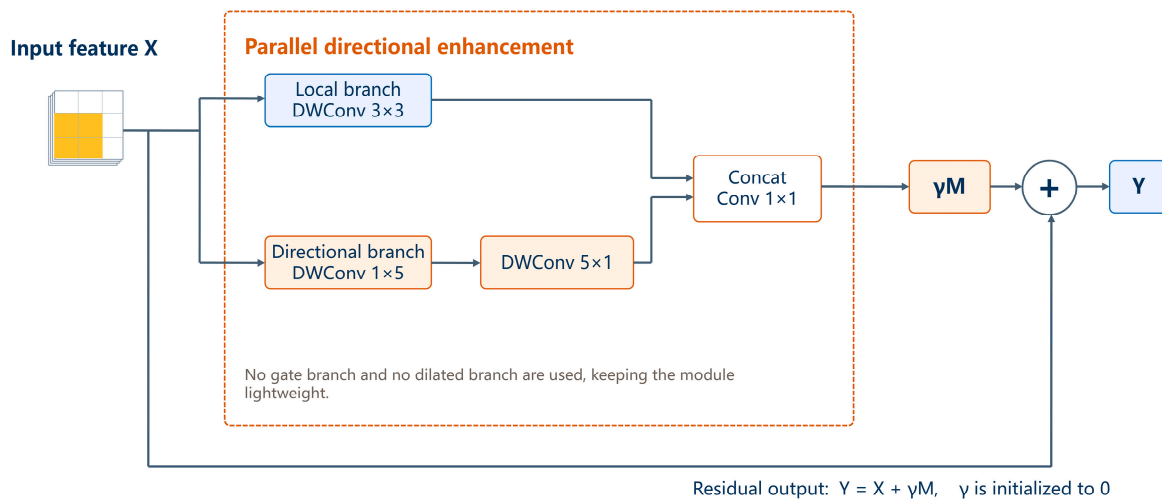


Figure 2. Parallel directional enhancement and residual structure of the DE Block.

3.4. Classification Loss and Weight Smoothing

The feature enhanced by the DE Block is passed to GAP and the classifier to obtain the classification logits of the i -th sample:

$$z_i = \text{FC}(\text{Dropout}(\text{GAP}(Y_i))) \quad (9)$$

$$p_{i,k} = \frac{\exp(z_{i,k})}{\sum_{j=1}^C \exp(z_{i,j})}, \quad k = 1, 2, \dots, C \quad (10)$$

where z_i denotes the C -dimensional logit vector produced by GAP and the classifier for the i -th sample, and $z_{i,k}$ is the unnormalized score assigned to class k ; $C = 10$ is the number of driver-behavior classes, and $p_{i,k}$ denotes the predicted probability that the i -th sample belongs to class k . The model is optimized end to end using the multiclass cross-entropy loss:

$$L_{CE} = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^C \mathbb{1}(y_i = k) \log(p_{i,k}) \quad (11)$$

where N is the mini-batch size, $\mathbb{1}(\cdot)$ is the indicator function, and y_i is the ground-truth class of the i -th sample. This loss directly maximizes the probability of the ground-truth class and is suitable for the 10-class single-label classification task in this paper. To reduce the influence of single-epoch parameter fluctuation on cross-driver testing, this paper maintains exponentially smoothed weights during training. Let $\theta(t)$ denote the online weights after the t -th update and $\varphi(t)$ denote the corresponding smoothed weights. The EWS update is defined as:

$$\varphi(t) = \mu\varphi(t-1) + (1-\mu)\theta(t) \quad (12)$$

where μ is the smoothing coefficient, set to 0.995 in this paper. Training gradients update only the online parameters $\theta(t)$, and the smoothed parameters $\varphi(t)$ are updated after each

optimizer step. Validation and testing use $\varphi(t)$ for inference. Weight averaging can reduce parameter fluctuation between adjacent updates [42]. Because EWS does not change the network topology, its parameter count and single forward-pass computation are the same as those of the online model.

3.5. Training and Inference Procedure

The training and inference protocol is shown in Figure 3. During training, the procedure sequentially performs LCF, shared forward propagation, cross-entropy backpropagation, and online/EWS parameter updates. Validation and testing do not use random augmentation. The smoothed parameters φ are loaded for the shared forward path; the validation set is used for checkpoint selection, and the test set is used for independent evaluation.

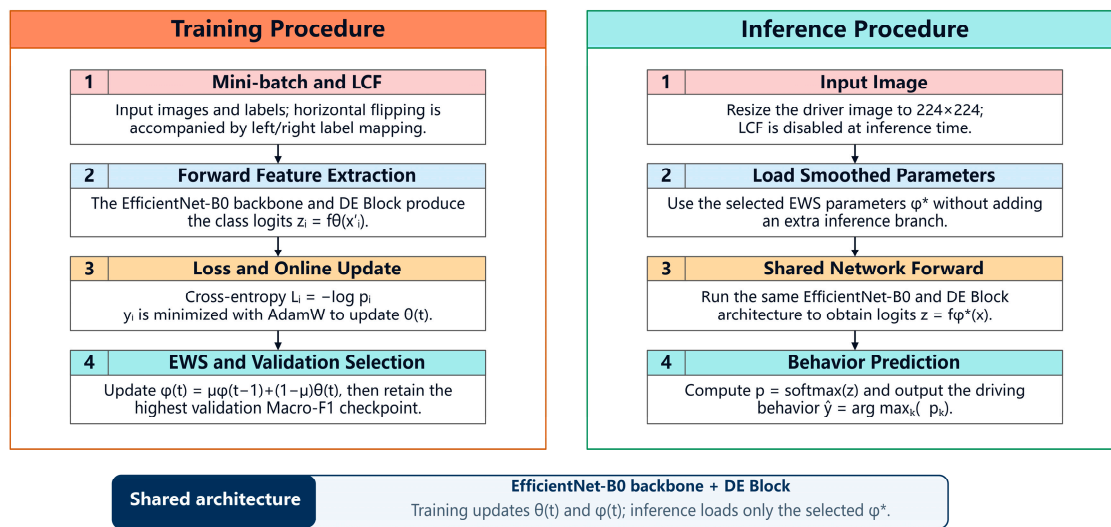


Figure 3. Training, model selection, and evaluation procedure of DEWS-EfficientNet. Here, φ^* denotes the selected smoothed parameter set used for evaluation, and the superscript * identifies the selected checkpoint.

4. Experiments

This section evaluates DEWS-EfficientNet using driver-disjoint partitions through backbone comparisons, full-factorial ablations, efficiency measurements, confusion matrices, and Grad-CAM visualizations. Unless otherwise stated, the main comparisons use the original driver-disjoint partition and random seed 42; Section 4.5.4 further evaluates two additional driver-disjoint partitions with random seeds 42, 2026, and 3407. Literature values are identified separately.

4.1. Datasets

We use two 10-class in-cabin datasets: State Farm Distracted Driver Detection (SFD3) and the Camera 1 subset of AUC Distracted Driver V2 (AUC2). Both contain safe driving, left- and right-hand texting and phone use, adjusting the radio, drinking, reaching behind, hair or makeup, and talking to a passenger. Figure 4 shows representative samples and the variation in driver posture and interaction regions. The corresponding driver-disjoint sample distributions reported in Tables 2 and 3 are reproduced from Gao and Liu [3].

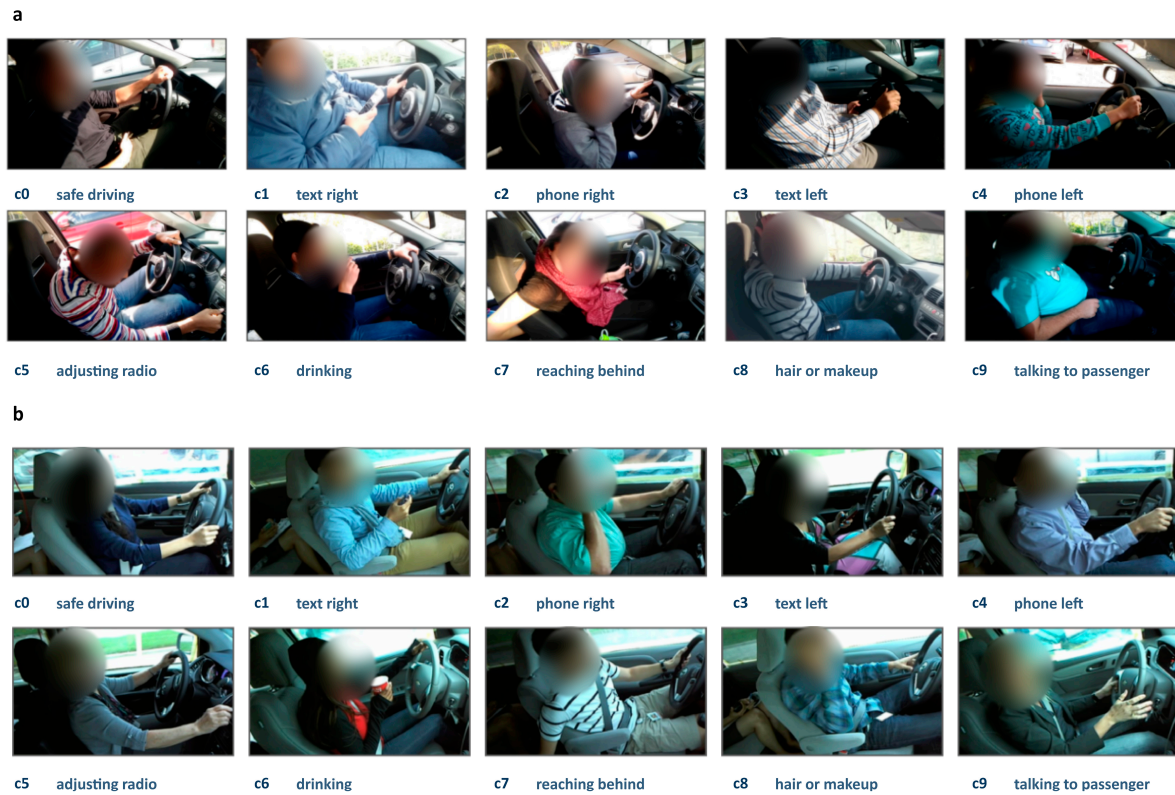


Figure 4. Examples of the 10 driver-behavior classes in the AUC2 and SFD3 datasets. (a) AUC2; (b) SFD3.

Images are grouped by driver before subset assignment so that all images and available acquisition sequences from one driver remain in a single training, validation, or test subset. This prevents identity-related appearance, clothing, camera position, and cabin background from crossing subset boundaries.

Table 2. Statistics of the State Farm dataset under the executed driver-disjoint split.

Statistics	Training Set	Validation Set	Test Set
number of drivers	15	5	6
number of images	14,835	3947	3642
safe driving	1650	441	398
text right	1512	358	397
phone right	1547	382	388
text left	1564	413	369
phone left	1548	405	373
adjusting radio	1566	404	342
drinking	1552	401	372
reaching behind	1290	387	325
hair or makeup	1213	379	319
talking to passenger	1393	377	359

Table 3. Statistics of the AUC2 Camera 1 subset under the executed driver-disjoint split.

Statistics	Training Set	Validation Set	Test Set
number of drivers	32	6	6
number of images	10,730	1825	1923
safe driving	2268	372	346
text right	1312	193	213
phone right	903	159	194
text left	832	112	180
phone left	1018	132	170
adjusting radio	800	153	170
drinking	769	164	143
reaching behind	713	178	143
hair or makeup	733	165	146
talking to passenger	1382	197	218

4.2. Evaluation Metrics

We report overall Accuracy, Macro-Precision, Macro-Recall, and Macro-F1 using scikit-learn [43]. For class k , $TP(k)$, $FP(k)$, and $FN(k)$ denote true positives, false positives, and false negatives. Precision, Recall, and F1 are computed per class and averaged over the 10 classes. Overall, Accuracy is the ratio of the C correctly classified samples to the N evaluated samples:

$$Acc = \frac{C}{N} \quad (13)$$

$$Sensitivity(k) = \frac{TP(k)}{TP(k) + FN(k)} \quad (14)$$

$$F1 - Score(k) = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (15)$$

$$Prec(k) = \frac{TP(k)}{TP(k) + FP(k)} \quad (16)$$

In addition to recognition performance, this paper reports the number of parameters (#Params), floating-point operations (FLOPs), and single-image inference speed on CPU and GPU, measured as frames per second (FPS), to evaluate deployment efficiency. LCF is used only during training augmentation, and EWS only changes the smoothed weights used for final inference. Neither introduces an additional inference branch. The DE Block performs lightweight residual enhancement only at a high-level feature stage, so its computational overhead remains controlled.

4.3. Comparison Methods

The baseline networks are AlexNet [44], SqueezeNet1.1 [45], ShuffleNetV2 [46], ResNet18 [47], and EfficientNet-B0 [16]. These models cover classical, lightweight, residual, and compound-scaled CNN architectures. A complete 2^3 factorial design evaluates the isolated and combined effects of LCF, the DE Block, and EWS on EfficientNet-B0. Table 4 summarizes the selected backbone architectures and the application of DEWS.

Table 4. CNN architectures evaluated in this paper.

Network	Transformation of Feature Maps	Usage of DEWS
AlexNet	flatten	DE Block + LCF + EWS
SqueezeNet1.1	classifier feature maps	DE Block + LCF + EWS
ShuffleNetV2	GAP	DE Block + LCF + EWS
ResNet18	GAP	DE Block + LCF + EWS
EfficientNet-B0	GAP	DE Block + LCF + EWS

4.4. Experimental Details

All experiments use PyTorch 1.11.0 and torchvision 0.12.0 on an NVIDIA GeForce RTX 4070 GPU. Each backbone is initialized with ImageNet-pretrained weights, and all images are processed at 224×224 resolution. AdamW is used with an initial learning rate of 0.0006 and weight decay of 0.1; the learning rate is linearly warmed up for five epochs and then decayed with a cosine schedule. The physical mini-batch size is 32, gradients are accumulated over four iterations to obtain an effective batch size of 128, and automatic mixed precision is enabled. Random seed 42 is fixed, every configuration is trained for 50 epochs, and early stopping is disabled. The common optimization and training settings are summarized in Table 5.

Table 5. Training parameter settings.

Parameter	Setting
Software environment	PyTorch 1.11.0; torchvision 0.12.0
Hardware environment	NVIDIA GeForce RTX 4070 GPU
Input size	224×224
Optimizer	AdamW
Initial learning rate	0.0006
Weight decay	0.1
Scheduler	warmup epochs = 5 + cosine scheduler
Batch setting	batch size = 32, accumulation steps = 4
Mixed precision	AMP
Final training	50 epochs, early stopping disabled
Model selection	the epoch with the highest validation Accuracy

During training, random resized cropping and color jitter are applied before normalization. The crop perturbs image scale and framing, while color jitter varies brightness, contrast, and saturation to simulate part of the appearance change caused by low-light or nighttime scenes and strong sunlight or glare. Together, they expose the model to moderate viewpoint, camera-framing, illumination, and color variation encountered in vehicle cabins. Horizontal flipping is coupled with LCF so that $c_1 \leftrightarrow c_3$ and $c_2 \leftrightarrow c_4$ are remapped whenever an image is flipped, thereby simulating mirrored left-right action layouts without corrupting labels. Validation and test images use deterministic resizing and normalization. An optimizer update is performed after every four physical mini-batches, after which EWS updates the shadow parameters with decay $\mu = 0.995$. At the end of each epoch, the current parameters are evaluated for configurations without EWS, whereas the smoothed parameters are evaluated for configurations with EWS. The checkpoint with the highest test Accuracy is retained, and its Accuracy, Macro-Precision, Macro-Recall, Macro-F1, confusion matrix, and Grad-CAM results are reported. Backbone comparisons and all eight factorial configurations use the same driver split, preprocessing, seed, optimizer, schedule, epoch budget, and checkpoint-selection rule.

4.5. Experimental Results and Discussion

4.5.1. Baseline Model Comparison

Table 6 summarizes the results of the five backbone networks. EfficientNet-B0 provides the strongest recorded baseline on both datasets and is therefore selected as the backbone for controlled ablations. According to the table, adding the complete DEWS scheme improves each backbone to different degrees. ResNet18, for example, improves by 7.08 percentage points on AUC2 and 9.37 percentage points on SFD3. Figure 5 shows the magnitude of the changes across backbones, indicating that the benefit of the method varies with backbone structure and data distribution.

Table 6. Performance comparison of baseline models on the two datasets (%). Best results are highlighted in bold.

Networks	State Farm				AUC2			
	Accuracy	F1	Recall	Precision	Accuracy	F1	Recall	Precision
AlexNet	60.35	60.21	61.08	66.24	61.42	62.90	61.22	72.94
DEWS-AlexNet	75.92	73.13	74.34	74.92	73.05	68.97	67.01	73.78
SqueezeNet	73.34	74.52	74.62	81.89	60.56	61.94	59.92	76.34
DEWS-SqueezeNet1.1	86.30	85.02	85.03	86.50	81.82	79.37	79.64	80.26
ShuffleNetV2	73.64	75.16	74.19	82.36	65.84	67.25	67.10	73.29
DEWS-ShuffleNetV2	81.18	78.80	79.93	82.25	80.63	78.43	78.12	80.67
ResNet18	79.13	79.94	79.23	82.33	73.20	72.35	71.72	76.19
DEWS-ResNet18	88.50	85.74	86.66	89.78	80.28	78.03	78.29	80.27
EfficientNet-B0	78.96	78.67	79.30	81.84	70.56	70.18	69.94	76.80
DEWS-EfficientNet	89.00	87.16	87.36	88.48	82.39	80.06	80.23	80.90

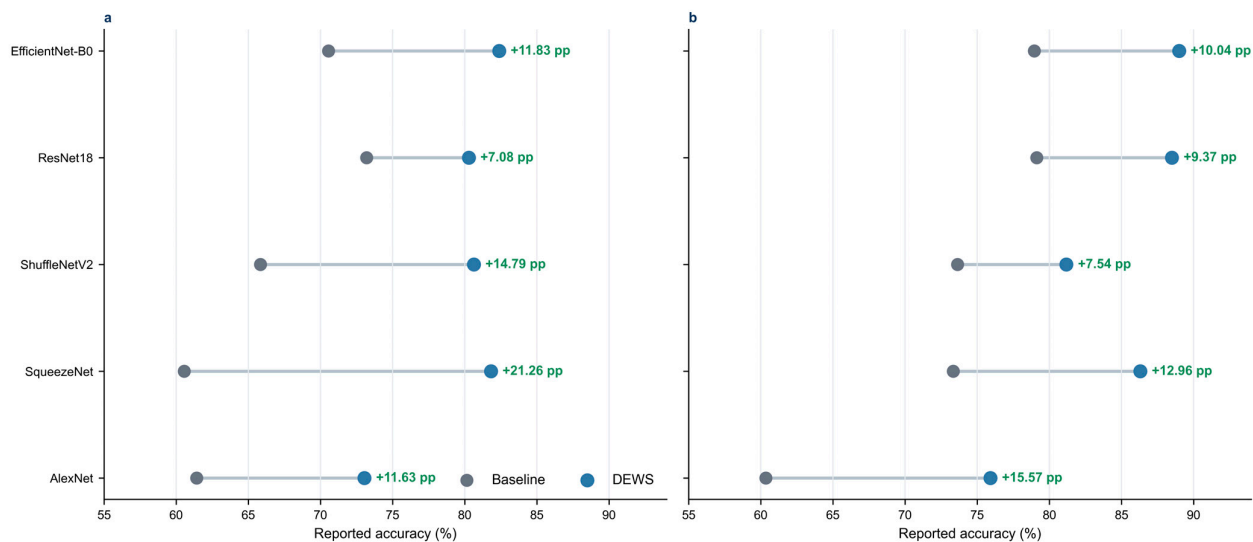
**Figure 5.** Cross-backbone performance comparison of the complete DEWS scheme: (a) AUC2; (b) SFD3.

Table 7 further compares the network efficiency of the five baseline backbones and their DEWS variants. After DEWS is added, the parameter counts and FLOPs of each model increase only slightly, while CPU and GPU FPS remain at a similar scale. This indicates that the proposed method does not introduce an independent extra inference branch. The increase in training-stage TTPE(s) mainly comes from label-consistent augmentation, directional feature learning, and weight smoothing.

Table 7. Network efficiency comparison of baseline models and their DEWS variants.

Network	Test Stage				Training Stage TTPE(s)
	#Params	FLOPs	FPSONCPU	FPSONGPU	
AlexNet	2.58M	662.10M	49.12	1125.36	1.72
DEWS-AlexNet	2.62M	668.54M	48.65	1108.24	4.86(+3.14)
SqueezeNet	0.74M	265.02M	35.91	296.84	6.61
DEWS-SqueezeNet1.1	0.77M	270.88M	35.40	291.76	9.08(+2.47)
ShuffleNetV2	1.27M	153.42M	28.96	85.74	21.40
DEWS-ShuffleNetV2	1.31M	158.36M	28.41	83.92	22.91(+1.51)
ResNet18	11.18M	1823.53M	7.84	71.26	39.72
DEWS-ResNet18	11.72M	1850.10M	7.61	69.83	42.10(+2.38)
EfficientNet-B0	4.03M	416.05M	11.72	61.84	38.26
DEWS-EfficientNet	4.10M	421.36M	11.54	60.97	41.52(+3.26)

4.5.2. Ablation Experiments

Table 8 reports the Accuracy, Macro-Precision, Macro-Recall, and Macro-F1 for all eight configurations of the 2^3 factorial design under the same EfficientNet-B0 backbone, driver-disjoint split, seed 42, and 50-epoch protocol. LCF, the DE Block, and EWS used alone increase the Accuracy by 3.75, 3.69, and 8.05 percentage points on SFD3 and by 1.62, 5.18, and 4.01 points on AUC2. DE Block + EWS gives the highest SFD3 Accuracy (89.36%), only 0.36 points above the complete combination (89.00%), whereas the complete combination reaches the highest AUC2 Accuracy (82.39%). Considering both datasets, the full LCF + DE Block + EWS configuration provides the most balanced overall result. When EWS is added to LCF + DE Block on AUC2, Macro-Precision decreases from 81.34% to 80.90% (−0.44 percentage points), while Macro-Recall rises from 74.61% to 80.23% (+5.62 points), Macro-F1 from 75.14% to 80.06% (+4.92 points), and Accuracy from 75.23% to 82.39% (+7.16 points). Macro-Precision averages TP/(TP + FP) across classes, whereas Macro-Recall averages TP/(TP + FN); therefore, EWS recovers more true samples from several classes while introducing a smaller number of additional false positives in ambiguous classes. Because the Recall, F1, and Accuracy gains substantially exceed the precision decrease, the result indicates a favorable Precision–Recall trade-off rather than a general degradation. Figure 6 summarizes the Accuracy and Macro-F1 for all eight configurations, and Figure 7 presents their validation trajectories.

Table 8. Full-factorial ablation of LCF, the DE Block, and EWS (%). ‘✓’ indicates that the component is included, whereas ‘—’ indicates that it is not included.

Components			SFD3				AUC2			
LCF	DE Block	EWS	Accuracy	Precision	Recall	Macro-F1	Accuracy	Precision	Recall	Macro-F1
—	—	—	78.96	81.84	79.30	78.67	70.56	76.80	69.94	70.18
✓	—	—	82.71	82.33	82.72	82.10	72.18	77.08	71.16	71.51
—	✓	—	82.65	82.89	82.56	82.38	75.74	78.54	75.62	76.04
—	—	✓	87.01	86.78	86.99	86.64	74.57	78.88	73.69	73.82
✓	✓	—	86.43	86.44	86.47	86.14	75.23	81.34	74.61	75.14
✓	—	✓	86.50	86.97	86.50	85.97	78.93	79.58	77.37	77.22
—	✓	✓	89.36	89.36	89.29	88.95	74.72	77.24	73.41	74.00
✓	✓	✓	89.00	88.48	87.36	87.16	82.39	80.90	80.23	80.06

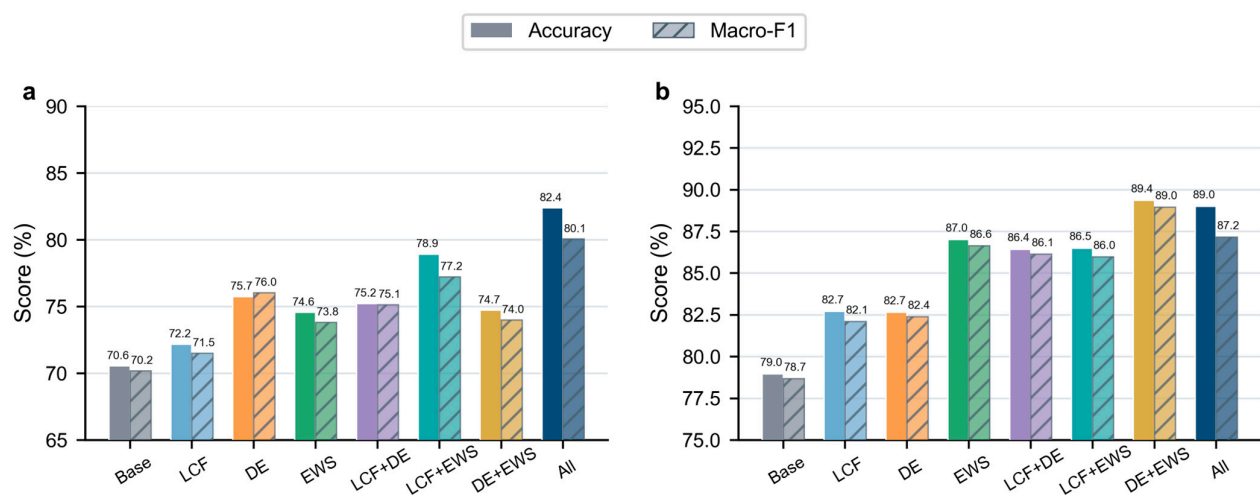


Figure 6. Accuracy and Macro-F1 for the full-factorial ablation under the controlled 50-epoch protocol: (a) AUC2; (b) SFD3.

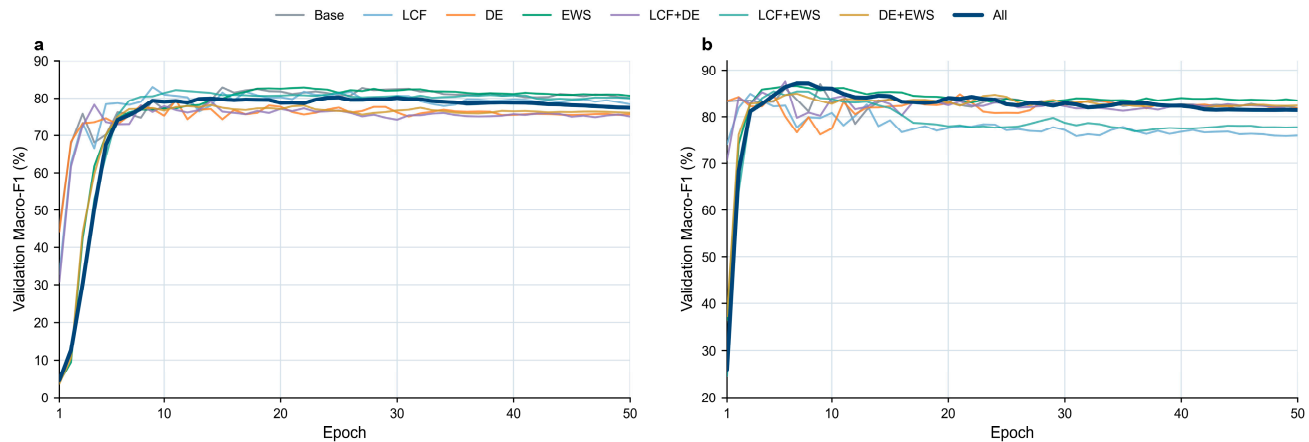


Figure 7. Validation Macro-F1 trajectories of all eight factorial configurations: (a) AUC2; (b) SFD3.

4.5.3. Comparison with Representative Methods

Table 9 provides a protocol-matched accuracy-and-complexity comparison rather than a reimplementing or common-hardware speed study. The D-HCNN, DD-RCNN, CAT-CapsNet, and MobileNetV2-CA values are taken from Gao and Liu [3] under the same dataset partition and evaluation definitions used in this study; only the DEWS-EfficientNet row is obtained here.

Table 9. Protocol-matched accuracy and model-complexity comparison with representative driver distraction models. Comparison-model values are taken from the previous publication [3]; the DEWS-EfficientNet row is from this study.

Network	State Farm	AUC2	FLOPs	#Params
D-HCNN [23]	77.85	71.74	3480.26M	0.76M
DD-RCNN [30]	86.00	-	≈134,379.72M	≈41.76M
CAT-CapsNet [31]	88.21	80.09	389.97M	3.39M
MobileNetV2-CA [3]	88.67	81.10	326.28M	2.24M
DEWS-EfficientNet	89.00	82.39	421.36M	4.10M

DEWS-EfficientNet records the highest Accuracy in Table 9, reaching 89.00% on State Farm and 82.39% on AUC2. These values exceed MobileNetV2-CA by 0.33 and 1.29 points, CAT-CapsNet by 0.79 and 2.30 points, and D-HCNN by 11.15 and 10.65 points, respectively. The proposed model uses 421.36 M FLOPs and 4.10 M parameters. Its computational demand is much lower than that of D-HCNN and DD-RCNN while remaining within the same order of magnitude as CAT-CapsNet and MobileNetV2-CA. These complexity values describe the model scale.

4.5.4. Repeated Driver-Level Validation

To evaluate the sensitivity to driver assignment and optimization randomness, we retained the original driver-disjoint partition and generated two alternative partitions by deterministic driver-level shuffling with partition seeds 20,261 and 20,262 while preserving the dataset-specific training, validation, and test split sizes. All images and acquisition sequences from a driver remained in the same subset. For each partition, DEWS-EfficientNet was trained for 50 epochs with random seeds 42, 2026, and 3407 under otherwise identical settings. Test-set Accuracy and Macro-F1 were recorded. Table 10 reports every partition-by-seed result and the within-partition mean $\pm$ SD. Using the three partition means as analysis units, AUC2 obtains $81.28 \pm 4.44\%$ Accuracy (95% CI 70.25–

92.31%) and $79.40 \pm 4.66\%$ Macro-F1 (95% CI 67.82–90.97%). SFD3 obtains $88.88 \pm 1.19\%$ Accuracy (95% CI 85.92–91.84%) and $87.87 \pm 1.68\%$ Macro-F1 (95% CI 83.70–92.03%).

Table 10. Repeated driver-level test of DEWS-EfficientNet across three driver partitions and three random seeds under the controlled 50-epoch protocol (%). Means $\pm$ SDs are computed over the three seeds within each partition.

Dataset	Split	Acc.-42	Acc.-2026	Acc.-3407	Acc. Mean $\pm$ SD	F1-42	F1-2026	F1-3407	F1 Mean $\pm$ SD
AUC2	Original	81.26	83.30	83.37	82.64 ± 1.20	79.53	81.21	79.89	80.21 ± 0.88
AUC2	Alternative A	81.00	88.38	85.24	84.87 ± 3.70	79.82	87.93	83.03	83.59 ± 4.08
AUC2	Alternative B	74.26	78.47	76.22	76.31 ± 2.11	72.90	76.35	73.89	74.38 ± 1.77
AUC2	Overall	—	—	—	81.28 ± 4.44	—	—	—	79.40 ± 4.66
SFD3	Original	88.95	86.83	87.72	87.83 ± 1.06	87.10	85.34	85.90	86.11 ± 0.90
SFD3	Alternative A	90.77	91.30	88.45	90.18 ± 1.51	90.23	90.37	87.76	89.45 ± 1.47
SFD3	Alternative B	88.66	88.46	88.78	88.63 ± 0.16	88.49	87.41	88.22	88.04 ± 0.56
SFD3	Overall	—	—	—	88.88 ± 1.19	—	—	—	87.87 ± 1.68

4.5.5. Confusion-Matrix Analysis

Figures 8 and 9 show the row-normalized confusion matrices of DEWS-EfficientNet on AUC2 and SFD3, respectively. On AUC2, the main errors are concentrated in $c2 \rightarrow c8$ (0.32), $c3 \rightarrow c0$ (0.27), $c6 \rightarrow c8$ (0.30), and $c7 \rightarrow c8$ (0.19). These classes share local patterns such as hands close to the face or torso, arm extension, and partial occlusion. On SFD3, $c1$, $c3$, $c4$, $c6$, and $c7$ are generally recognized with higher diagonal responses, whereas $c8$ and $c9$ remain more difficult because hair-or-makeup and passenger-talking behaviors contain large pose changes and can occupy similar head-and-hand regions.

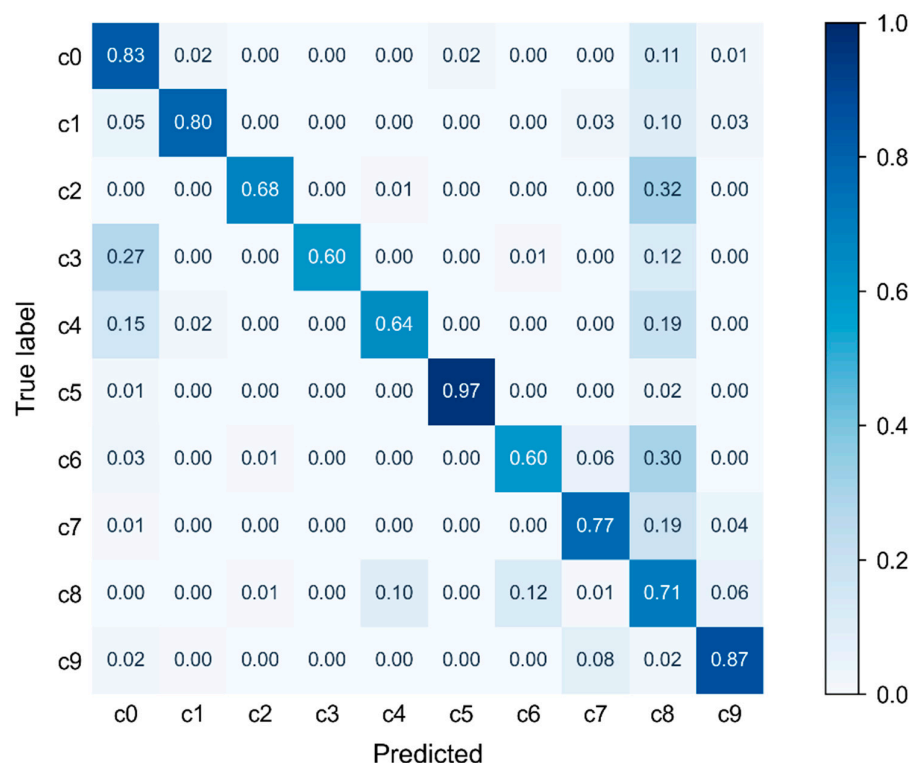


Figure 8. Normalized confusion matrix of DEWS-EfficientNet on AUC2.

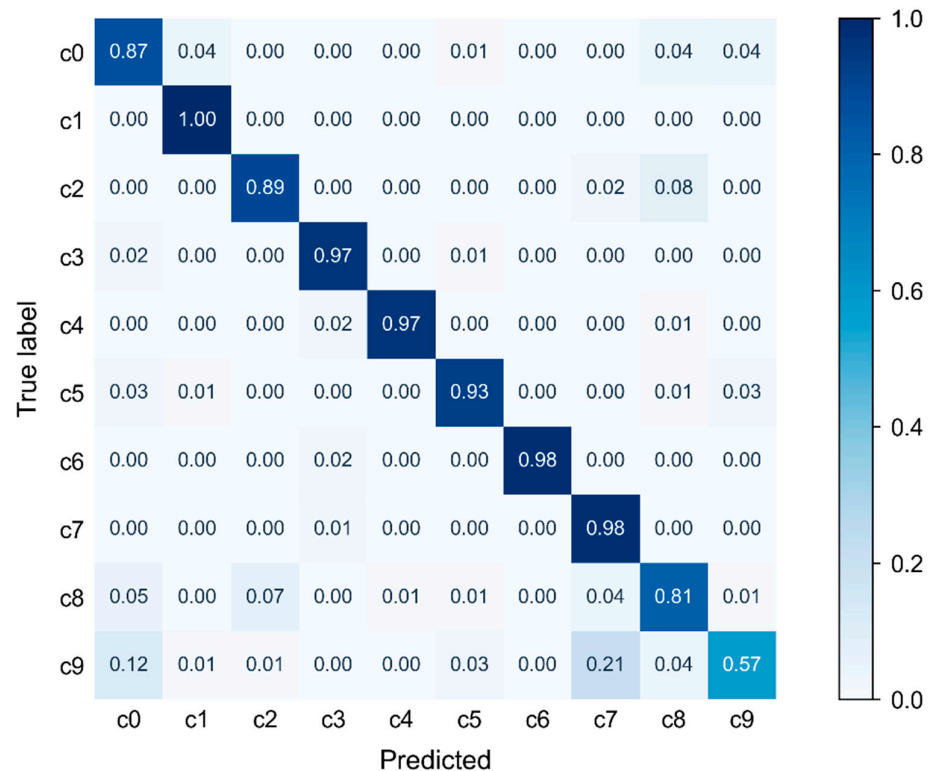


Figure 9. Normalized confusion matrix of DEWS-EfficientNet on SFD3.

The two datasets exhibit different error structures. In AUC2, the confusions among c2, c3, c4, c6, and c8 are more dispersed, indicating that the camera viewpoint, illumination, and driver-pose variation have a stronger influence on local hand cues. In SFD3, the errors are more concentrated around c9, reflecting the large intra-class variation of talking-to-passenger behavior. The DE Block strengthens actions with clear directional patterns, but it remains limited when the motion amplitude is small, the hands are occluded, or the head pose alone is insufficient for discrimination. These observations are consistent with the ablation results and indicate that DEWS mainly improves spatial directional representation and training stability rather than solving all fine-grained behavior ambiguities.

4.5.6. Grad-CAM Visualization Analysis

To assess both predictive confidence and spatial attention on difficult AUC2 samples, Figure 10 reports the ground-truth class probability $p(gt)$ together with Gradient-weighted Class Activation Mapping (Grad-CAM) [48]. EfficientNet-B0 assigns $p(gt) = 0.28\text{--}0.33$ and misclassifies the selected samples, whereas DEWS-EfficientNet increases $p(gt)$ to $0.92\text{--}0.99$ and predicts every displayed ground-truth class correctly. Its heatmaps also concentrate more clearly on the hands, phone, head, and steering-wheel interaction regions while relying less on windows and unrelated background. Some responses still extend to shared torso and steering-wheel areas, consistent with the remaining confusion among phone use, texting, drinking, and passenger interaction. The combined confidence and spatial evidence therefore indicates stronger responses to behavior-relevant cues for these selected samples.

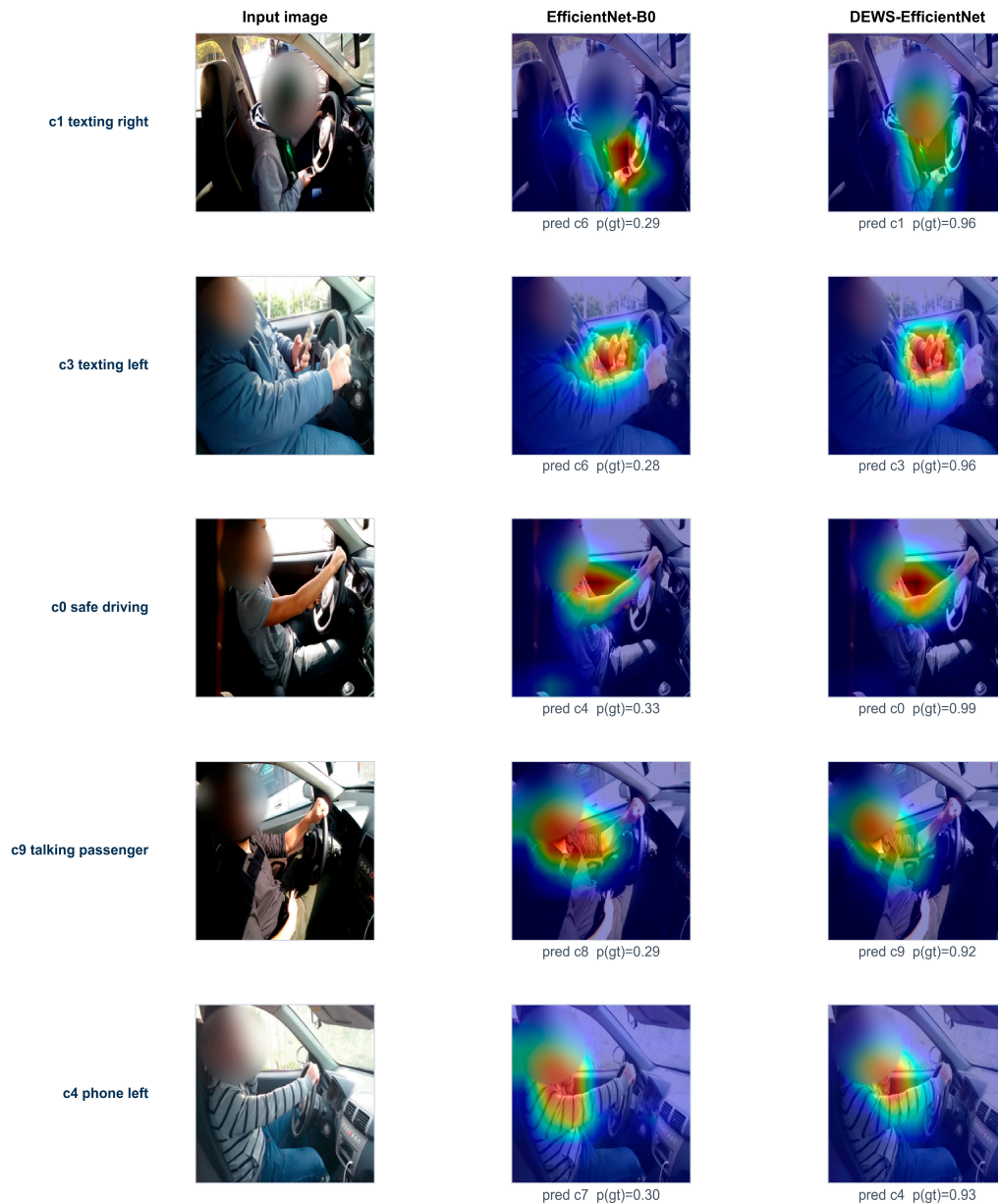


Figure 10. Grad-CAM visualization of EfficientNet-B0 and DEWS-EfficientNet on confusing AUC2 samples. The heatmaps are generated for the ground-truth behavior class; warmer colors indicate stronger class-discriminative responses.

4.5.7. Overall Discussion

The main evidence comes from the complete factorial ablation on EfficientNet-B0 and the cross-backbone comparison. Each isolated component improves the EfficientNet-B0 baseline. DE Block + EWS reaches the highest SFD3 Accuracy (89.36%), but the complete configuration remains close at 89.00% and achieves the highest AUC2 Accuracy (82.39%); the full method is therefore selected as the most balanced configuration across the two datasets. Moreover, consistent gains across AlexNet, SqueezeNet1.1, ShuffleNetV2, ResNet18, and EfficientNet-B0 indicate that the input-level label correction, high-level directional refinement, and parameter smoothing are compatible with different convolutional backbones rather than being tied to one architecture.

Real-world deployment may include low-light or nighttime scenes, strong sunlight or glare, facial occlusion by sunglasses or scarves, camera-viewpoint changes, and different cabin layouts. The augmentations used in all experiments partially simulate these variations. Color jitter changes brightness, contrast, and saturation to approximate low- and strong-light appearance changes. Random resized cropping varies scale and framing, while LCF-coupled flipping covers mirrored left-right actions. However, augmentation is not equivalent to controlled condition-specific testing, AUC2 and SFD3 provide no condition labels for stratified estimates, and the current protocol does not directly simulate sunglasses or scarf occlusion. Future work will use Random Erasing or Cutout, illumination normalization, style randomization, temporal evidence, and condition-annotated in-vehicle evaluation to examine these conditions explicitly.

5. Conclusions

This study presents DEWS-EfficientNet, a single-branch driver-behavior classifier combining LCF, the DE Block, and EWS. It reaches 82.39% Accuracy on AUC2 and 89.00% on SFD3, exceeding the corresponding EfficientNet-B0 baselines by 11.83 and 10.04 percentage points. LCF maintains semantic consistency under horizontal flipping, the DE Block strengthens local directional action cues, and EWS stabilizes model parameters without adding an inference branch. The complete configuration also improves five convolutional backbones, demonstrating practical compatibility with standard driver-monitoring classifiers at moderate computational cost. Repeated testing across three driver-disjoint partitions and three random seeds additionally quantifies sensitivity to driver assignment and initialization.

The present study uses single-frame images from two public datasets and does not yet measure temporal stability, embedded-device latency, memory consumption, or energy use. Future work will introduce stronger occlusion- and illumination-aware augmentation, short temporal or multimodal cues, domain-adaptive normalization, and embedded-device profiling. These extensions will support more robust recognition under changing cabin conditions while preserving the compact single-branch inference path.

Author Contributions: Conceptualization, Y.Z.; Methodology, C.C., Z.H., and X.Z.; Software, Y.Z.; Validation, Y.Z. and Z.H.; Formal analysis, Y.Z.; Investigation, Y.Z.; Resources, C.C. and X.Z.; Data Curation, Y.Z.; Writing—Original Draft Preparation, Y.Z.; Writing—Review and Editing, C.C., Z.H., and X.Z.; Visualization, Y.Z.; Supervision, C.C.; Project Administration, C.C.; Funding Acquisition, C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this study are publicly available. The State Farm Distracted Driver Detection dataset can be accessed at <https://www.kaggle.com/c/state-farm-distracted-driver-detection> (accessed on 30 June 2026), and the AUC Distracted Driver V2 dataset can be accessed at <https://service.tib.eu/ldmservice/dataset/american-university-in-cairo-auc-distracted-drivers-dataset> (accessed on 30 June 2026).

Acknowledgments: We would like to thank the College of Engineering Science and Technology, Shanghai Ocean University, for providing research support and facilities that made this study possible.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Susanto, N.; Ulkhaq, M.M.; Tindaon, U.A. The Effect of Mobile Phone Use, Traffic Environment, and the Interaction on Driving Performance of Adult Drivers: Empirical Findings from 2² Factorial Design Simulation. *Int. J. Transp. Dev. Integr.* **2024**, *8*, 435–445. <https://doi.org/10.18280/ijtdi.080307>.
2. Piancastelli, L. Human-Factors and Power-System Implications of Consumer Electronics-Oriented Vehicle Design and Full-Electric Fleet Deployment. *Power Eng. Eng. Thermophys.* **2025**, *4*, 168–177. <https://doi.org/10.56578/peet040303>.
3. Gao, H.; Liu, Y. Improving Real-Time Driver Distraction Detection via Constrained Attention Mechanism. *Eng. Appl. Artif. Intell.* **2024**, *128*, 107408. <https://doi.org/10.1016/j.engappai.2023.107408>.
4. Lv, M.; Liu, Y.; Zha, Z.; Zheng, X.; Wang, H.; Wen, Y.; Guo, Z. Si-CA MobileNet: A Lightweight and Efficient Convolutional Neural Network for Distracted Driver Detection. *Neurocomputing* **2025**, *654*, 131281. <https://doi.org/10.1016/j.neucom.2025.131281>.
5. Khan, T.; Choi, G.; Lee, S. EFFNet-CA: An Efficient Driver Distraction Detection Based on Multiscale Features Extractions and Channel Attention Mechanism. *Sensors* **2023**, *23*, 3835. <https://doi.org/10.3390/s23083835>.
6. Jegham, I.; Alouani, I.; Ben Khalifa, A.; Mahjoub, M.A. Deep Learning-Based Hard Spatial Attention for Driver in-Vehicle Action Monitoring. *Expert Syst. Appl.* **2023**, *219*, 119629. <https://doi.org/10.1016/j.eswa.2023.119629>.
7. Du, Y.; Liu, X.; Yi, Y.; Wei, K. Optimizing Road Safety: Advancements in Lightweight YOLOv8 Models and GhostC2f Design for Real-Time Distracted Driving Detection. *Sensors* **2023**, *23*, 8844. <https://doi.org/10.3390/s23218844>.
8. Shen, Q.; Zhang, L.; Zhang, Y.; Zhang, Y.; Liu, S.; Li, Y. Ddr-Yolo: An Efficient and Accurate Object Detection Algorithm for Distracted Driving Behaviors. *Expert Syst. Appl.* **2025**, *308*, 131170. <https://doi.org/10.2139/ssrn.5173420>.
9. Wang, J.; Wu, Z. Driver Distraction Detection via Multi-scale Domain Adaptation Network. *IET Intell. Transp. Syst.* **2023**, *17*, 1742–1751. <https://doi.org/10.1049/itr2.12366>.
10. Yang, L.; Wei, H.; Hu, Z.; Lv, C. A Domain Generalization Method for Deploying Driver Distraction Detection Models to Practical Application Scenarios. *Eng. Appl. Artif. Intell.* **2025**, *152*, 110844. <https://doi.org/10.1016/j.engappai.2025.110844>.
11. Mohammed, A.A.Q.; Geng, X.; Wang, J.; Ali, Z. Driver Distraction Detection Using Semi-Supervised Lightweight Vision Transformer. *Eng. Appl. Artif. Intell.* **2024**, *129*, 107618. <https://doi.org/10.1016/j.engappai.2023.107618>.
12. Ma, Y.; Wang, Z. ViT-DD: Multi-Task Vision Transformer for Semi-Supervised Driver Distraction Detection. *arXiv* **2024**, arXiv:2209.09178. <https://doi.org/10.48550/arXiv.2209.09178>.
13. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016*; IEEE: New York, NY, USA, 2016; pp. 2818–2826. <https://doi.org/10.1109/CVPR.2016.308>.
14. Romera, E.; Alvarez, J.M.; Bergasa, L.M.; Arroyo, R. ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 263–272. <https://doi.org/10.1109/TITS.2017.2750080>.
15. Ding, X.; Guo, Y.; Ding, G.; Han, J. ACNet: Strengthening the Kernel Skeletons for Powerful CNN via Asymmetric Convolution Blocks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019*; IEEE: New York, NY, USA, 2019; pp. 1911–1920. <https://doi.org/10.1109/ICCV.2019.00200>.
16. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv* **2020**, arXiv:1905.11946. <https://doi.org/10.48550/arXiv.1905.11946>.
17. Joel John, M.; Dinakaran, K.; Bharathiraja, N. Advancing Transport Safety with Faster Pre-Convolved Neural Networks and Lightweight Multi-Scale Fusion for Driver Distraction Detection. *Int. J. Heavy Veh. Syst.* **2024**, *1*, 10064928. <https://doi.org/10.1504/IJHVS.2024.10064928>.
18. Mohsin, A.R.; Khalid, M.; Ali, J.; Roh, B. Enhancing Driver Safety with ResNet-RG—A Deep Learning-Based Distraction Detection Approach. *IEEE Sens. J.* **2026**, *26*, 4664–4683. <https://doi.org/10.1109/JSEN.2025.3640661>.
19. Hassam, D.D.; Mohamme, A.T.; Sadiq, A.S.; Salman, B.Y. Driver Distracted Detection Using a New Hybrid Deep Convolutional Neural Network Based on Autoencoder. *Eng. Res. Express* **2025**, *7*, 045219. <https://doi.org/10.1088/2631-8695/ae0ff5>.
20. Luo, G.; Guo, B.; Jin, L.; Zhang, Y.; Liu, X.; Liu, J.; Shi, W.; Yao, H. Enhancing Lightweight Distracted Driving Behavior Detection Capabilities Using KRCA Knowledge Distillation. *Meas. Sci. Technol.* **2026**, *37*, 036108. <https://doi.org/10.1088/1361-6501/ae2a5e>.
21. Eraqi, H.M.; Abouelnaga, Y.; Saad, M.H.; Moustafa, M.N. Driver Distraction Identification with an Ensemble of Convolutional Neural Networks. *J. Adv. Transp.* **2019**, *2019*, 4125865. <https://doi.org/10.1155/2019/4125865>.
22. Baheti, B.; Talbar, S.; Gajre, S. Towards Computationally Efficient and Realtime Distracted Driver Detection with MobileVGG Network. *IEEE Trans. Intell. Veh.* **2020**, *5*, 565–574. <https://doi.org/10.1109/TIV.2020.2995555>.

23. Qin, B.; Qian, J.; Xin, Y.; Liu, B.; Dong, Y. Distracted Driver Detection Based on a CNN with Decreasing Filter Size. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 6922–6933. <https://doi.org/10.1109/TITS.2021.3063521>.
24. Liu, D.; Yamasaki, T.; Wang, Y.; Mase, K.; Kato, J. Toward Extremely Lightweight Distracted Driver Recognition with Distillation-Based Neural Architecture Search and Knowledge Transfer. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 764–777. <https://doi.org/10.1109/TITS.2022.3217342>.
25. Li, Z.; Zhao, X.; Wu, F.; Chen, D.; Wang, C. A Lightweight and Efficient Distracted Driver Detection Model Fusing Convolutional Neural Network and Vision Transformer. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 19962–19978. <https://doi.org/10.1109/TITS.2024.3447041>.
26. Gu, S.; Wen, B.; Chen, S.; Li, Y.; Qi, G.; Shuai, L.; Zhu, Z. Driver Distraction Detection Based on Adaptive Tiny Targets and Lightweight Networks. *Signal Process. Image Commun.* **2025**, *138*, 117342. <https://doi.org/10.1016/j.image.2025.117342>.
27. Rampavan, M.; Ijjina, E.P. An Improved Genetic Algorithm Based Deep Learning Model with You Only Look Once Framework for Driver Distraction Detection. *Eng. Appl. Artif. Intell.* **2025**, *160*, 111932. <https://doi.org/10.1016/j.engappai.2025.111932>.
28. Debsi, A.; Ling, G.; Al-Mahbashi, M.; Al-Soswa, M.; Abdullah, A. Driver Distraction and Fatigue Detection in Images Using ME-YOLOv8 Algorithm. *IET Intell. Transp. Syst.* **2024**, *18*, 1910–1930. <https://doi.org/10.1049/itr2.12560>.
29. Xin, S.; Zhang, S.; Xu, W.; Yang, Y.; Zhang, X. Deep Learning-Based Eye Tracking System to Detect Distracted Driving. *Meas. Sci. Technol.* **2024**, *35*, 096303. <https://doi.org/10.1088/1361-6501/ad4e51>.
30. Lu, M.; Hu, Y.; Lu, X. Driver Action Recognition Using Deformable and Dilated Faster R-CNN with Optimized Region Proposals. *Appl. Intell.* **2020**, *50*, 1100–1111. <https://doi.org/10.1007/s10489-019-01603-4>.
31. Mittal, H.; Verma, B. CAT-CapsNet: A Convolutional and Attention Based Capsule Network to Detect the Driver's Distraction. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 9561–9570. <https://doi.org/10.1109/TITS.2023.3266113>.
32. Gong, Y.; Shen, X. An Algorithm for Distracted Driving Recognition Based on Pose Features and an Improved KNN. *Electronics* **2024**, *13*, 1622. <https://doi.org/10.3390/electronics13091622>.
33. Tango, F.; Botta, M. Real-Time Detection System of Driver Distraction Using Machine Learning. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 894–905. <https://doi.org/10.1109/TITS.2013.2247760>.
34. Liu, T.; Yang, Y.; Huang, G.-B.; Yeo, Y.K.; Lin, Z. Driver Distraction Detection Using Semi-Supervised Machine Learning. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1108–1120. <https://doi.org/10.1109/TITS.2015.2496157>.
35. Aksjonov, A.; Nedoma, P.; Vodovozov, V.; Petlenkov, E.; Herrmann, M. Detection and Evaluation of Driver Distraction Using Machine Learning and Fuzzy Logic. *IEEE Trans. Intell. Transp. Syst.* **2019**, *20*, 2048–2059. <https://doi.org/10.1109/TITS.2018.2857222>.
36. Liu, H.; Fu, C.; Zhou, Y.; Shi, P.; Jiang, C. A Two-Level Fusion CNN Model for Classifying Metro Drivers' Distractions with Functional near-Infrared Spectroscopy and Electrocardiography Signals. *Ergonomics* **2025**, *1*–25. <https://doi.org/10.1080/00140139.2025.2605050>.
37. Noh, B.; Park, M.; Han, Y.; Kim, J. A Multi-Modal Approach for Detecting Drivers' Distraction Using Bio-Signal and Vision Sensor Fusion in Driver Monitoring Systems. *Eng. Appl. Artif. Intell.* **2025**, *161*, 112265. <https://doi.org/10.1016/j.engappai.2025.112265>.
38. Mou, L.; Chang, J.; Zhou, C.; Zhao, Y.; Ma, N.; Yin, B.; Jain, R.; Gao, W. Multimodal Driver Distraction Detection Using Dual-Channel Network of CNN and Transformer. *Expert Syst. Appl.* **2023**, *234*, 121066. <https://doi.org/10.1016/j.eswa.2023.121066>.
39. Elhenawy, M.; Masoud, M.; Haworth, N.; Young, K.; Rakotonirainy, A.; Grzebieta, R.; Williamson, A. Detection of Driver Distraction in the Australian Naturalistic Driving Study Videos Using Pre-Trained Models and Transfer Learning. *Transp. Res. Part F Traffic Psychol. Behav.* **2023**, *97*, 31–43. <https://doi.org/10.1016/j.trf.2023.06.016>.
40. Hasan, M.Z.; Chen, J.; Wang, J.; Rahman, M.S.; Joshi, A.; Velipasalar, S.; Hegde, C.; Sharma, A.; Sarkar, S. Vision-Language Models Can Identify Distracted Driver Behavior From Naturalistic Videos. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 11602–11616. <https://doi.org/10.1109/TITS.2024.3381175>.
41. Zhang, Z.; Yang, L.; Lv, C. Highly Discriminative Driver Distraction Detection Method Based on Swin Transformer. *Vehicles* **2024**, *6*, 140–156. <https://doi.org/10.3390/vehicles6010006>.
42. Tarvainen, A.; Valpola, H. Mean Teachers Are Better Role Models: Weight-Averaged Consistency Targets Improve Semi-Supervised Deep Learning Results. *arXiv* **2018**, arXiv:1703.01780. <https://doi.org/10.48550/arXiv.1703.01780>.
43. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Müller, A.; Nothman, J.; Louppe, G.; et al. Scikit-Learn: Machine Learning in Python. *arXiv* **2018**, arXiv:1201.0490. <https://doi.org/10.48550/arXiv.1201.0490>.
44. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. <https://doi.org/10.1145/3065386>.

45. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and <0.5MB Model Size. *arXiv* **2016**, arXiv:1602.07360. <https://doi.org/10.48550/arXiv.1602.07360>.
46. Ma, N.; Zhang, X.; Zheng, H.-T.; Sun, J. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. *arXiv* **2018**, arXiv:1807.11164. <https://doi.org/10.48550/arXiv.1807.11164>.
47. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016*; IEEE: New York, NY, USA, 2016; pp. 770–778. <https://doi.org/10.1109/CVPR.2016.90>.
48. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *Int. J. Comput. Vis.* **2020**, *128*, 336–359. <https://doi.org/10.1007/s11263-019-01228-7>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.