

SafeRider: A Multimodal AI Framework for Real-Time Driver Behaviour Monitoring Using Transfer-Learned CNN and ExtraTrees Sensor Fusion

Anushka Bharat Kale

Artificial Intelligence and Data Science Zeal College of Engineering and Research Pune, India

anushkale2005@gmail.com

Abstract — Driver fatigue, distraction, and abnormal driving behaviour remain leading contributors to road accidents worldwide, underlining the urgent need for intelligent monitoring systems that assess driver state from multiple complementary signals [1], [2]. This paper presents *SafeRider*, an AI-driven multimodal driver behaviour monitoring system that fuses a transfer-learned Convolutional Neural Network (CNN) image classifier with an ExtraTrees-based sensor classifier to predict the driver's state in real time. The visual stream is processed by a MobileNetV2 backbone with a custom multilabel head that independently predicts eye state (open/closed) and mouth state (yawning/neutral). In parallel, a 7-feature numerical payload — vehicle telemetry (speed, RPM, throttle, brake, steering angle) and physiological signals (heart rate, stress level) — is enriched with six engineered features and classified into five behavioural categories (*attentive*, *distracted*, *drowsy*, *aggressive*, *eco-driving*) using an ExtraTrees classifier trained on a SMOTE-balanced dataset. A rule-based fusion layer combines the two streams into a graded verdict (*Driver Alert*, *Warning*, *Critical*). Experimental results on a curated dataset of 301 facial images and 1,200 sensor rows demonstrate a multilabel CNN combined-accuracy of **96.05 %** and a sensor classifier hold-out accuracy of **92.92 %** (5-fold CV **94.48 %**), validating that the proposed multimodal fusion approach improves reliability and reduces false predictions compared to single-model systems [8], [17], [19]. The framework is deployed as a production-style Flask service consumed by a React dashboard with text-to-speech alerts, providing an effective foundation for intelligent driver monitoring and advanced driver assistance systems.

Index Terms — Driver monitoring, Multimodal fusion, Transfer learning, MobileNetV2, ExtraTrees, SMOTE, Random Forest, Convolutional Neural Network, Flask, React, Text-to-speech, Road safety, Deep learning, Machine learning.

I. INTRODUCTION

Road safety has become a critical concern due to the rising number of traffic accidents caused by human factors such as driver fatigue, distraction, and abnormal physical conditions. According to the World Health Organisation, over 1.3 million people die in road traffic accidents every year, and human error contributes to more than 90 % of these incidents. Despite continual advances in passive vehicle safety such as airbags, seat belts, and anti-lock braking systems, driver-related errors continue to be the dominant cause of accidents worldwide. Monitoring the driver's state in real time is therefore essential to reduce accident risk and improve overall driving safety.

Traditional driver behaviour monitoring systems typically rely on a single source of information, such as vehicle dynamics, facial expressions, or physiological signals [8]. Systems based solely on vehicle telemetry analyse speed and steering patterns but fail to capture the driver's cognitive condition. Vision-based systems focus on facial cues such as eye closure or yawning but may be affected by lighting conditions and occlusions [9], [13], [15]. Health-based monitoring systems, when used independently, provide physiological insight but lack contextual information about driving behaviour [11], [16]. As a result, unimodal approaches often produce incomplete or unreliable assessments of driver state.

This paper presents *SafeRider*, an AI-driven driver behaviour monitoring system that combines visual information from in-cabin facial images with numerical vehicle and driver-health data to predict the driver's state accurately. The visual stream is analysed by a MobileNetV2 transfer-learned CNN trained in two phases on 301 curated face images;

the model is delivered in two interchangeable variants — a legacy 3-class softmax head and a multilabel head with two independent sigmoid outputs (*eye_open*, *mouth_yawn*). The numerical stream — comprising seven raw sensor parameters and six engineered features — is classified by an ExtraTrees ensemble selected via a 5-fold cross-validation shoot-out against tuned Random Forest, Voting, Stacking, XGBoost, and Gradient Boosting candidates. A rule-based fusion layer integrates both predictions and emits a graded verdict, while a Flask backend with structured logging and request-ID tracing serves a React dashboard that delivers text-to-speech alerts. By leveraging multimodal data fusion, the proposed approach aims to improve reliability, reduce false predictions, and contribute to enhanced road safety and advanced driver assistance systems [8], [17], [19].

II. PROBLEM STATEMENT

Road accidents are frequently caused by driver fatigue, distraction, and abnormal physical conditions, which reduce alertness and increase the likelihood of unsafe driving [1], [8]. Existing driver monitoring systems often rely on a single source of information — such as visual cues, vehicle dynamics, or health parameters — analysed independently. This unimodal approach limits the accuracy and reliability of driver-state assessment, as it fails to capture the complex interaction between driver behaviour, vehicle movement, and physiological condition.

Moreover, systems that focus only on image-based detection may be affected by environmental factors such as low light or occlusion, while numerical-data-based approaches lack direct observation of driver actions. Therefore, there is a need for an intelligent and integrated driver behaviour monitoring system that combines visual information with vehicle and driver health data [17], [19]. Such a system should be capable of analysing multiple data modalities simultaneously and fusing their outputs to predict the driver's state accurately. Addressing this problem can enable early detection of unsafe driving conditions and support the development of advanced driver assistance systems aimed at improving road safety.

III. LITERATURE SURVEY

Driver behaviour monitoring has been an active area of research due to its importance in reducing accident rates. Existing literature can be grouped into vision-based, vehicle-based, health-based, and hybrid monitoring approaches.

Vision-based monitoring systems use computer-vision techniques to analyse facial expressions and head movements for detecting fatigue and distraction. Methods such as eye-closure detection, yawning detection, and head-pose estimation have been widely studied. Convolutional Neural Networks (CNNs) have demonstrated strong performance in extracting spatial features from facial images and identifying drowsiness-related patterns [13], [15]. However, vision-only systems often suffer from limitations related to lighting conditions, camera-angle variation, and occlusions such as sunglasses, which can reduce detection accuracy.

Vehicle-based monitoring systems analyse driving patterns using parameters such as speed, steering angle, lane deviation, and braking behaviour. These systems are commonly used in telematics and fleet-management applications [4], [5]. While vehicle data provide valuable insight into driving dynamics, they do not directly reflect the driver's physical or cognitive state; unsafe conditions such as fatigue may therefore go undetected when relying solely on vehicle parameters.

Health-based and biometric monitoring approaches focus on physiological signals such as heart rate, body temperature, blood pressure, and electrocardiogram (ECG) readings to identify fatigue, stress, or abnormal driver conditions [11], [16]. These systems capture the driver's physical state effectively but, when used independently, lack contextual information about driving behaviour and visual cues. Practical deployment may also be affected by sensor availability and user comfort.

Hybrid and multimodal systems have therefore emerged as the most promising direction. Recent works combine multiple data sources — facial cues, vehicle telemetry, and physiological signals — and report improved accuracy and robustness over unimodal baselines [17], [19]. The *SafeRider* framework proposed in this paper extends this line of work by combining a transfer-learned MobileNetV2 image classifier with an SMOTE-balanced ExtraTrees sensor classifier, and integrating the two streams through a deterministic fusion ladder that produces a graded driver-state verdict.

IV. PROPOSED SOLUTION

The proposed solution addresses the limitations of unimodal driver monitoring systems by introducing an AI-driven multimodal framework for accurate driver behaviour assessment. The system integrates visual information obtained from driver facial images with numerical data derived from vehicle dynamics and driver health parameters to provide a comprehensive evaluation of the driver's state.

The proposed system processes facial images using a **MobileNetV2** transfer-learned CNN that is fine-tuned in two phases on a curated face dataset. The CNN is available in two variants: a 3-class softmax head (*closed_eyes*, *open_eyes*, *yawn*) and a multilabel head with two independent sigmoid outputs (*eye_open* and *mouth_yawn*), which provides finer-grained drowsiness cues. Test-time augmentation (10 augmented passes plus 1 clean pass) is applied at inference to further stabilise the prediction. In parallel, a 7-feature numerical payload — vehicle parameters (speed, RPM, throttle, brake, steering angle) and physiological signals (heart rate, stress level) — is enriched with six engineered features (speed-to-RPM ratio, throttle-brake balance, steering severity, physiological load, aggression index, fatigue index) and analysed using an **ExtraTrees classifier** trained on a SMOTE-balanced dataset to assess driving behaviour and physiological condition.

To enhance reliability, the outputs of both models are combined using a rule-based fusion layer. The fusion logic considers both visual indicators (eye and mouth state) and the sensor verdict (one of *aggressive*, *attentive*, *distracted*, *drowsy*, *eco-driving*) to generate a final driver state classification. If both modalities indicate unsafe behaviour, the system flags a *Critical* state. If a single modality detects abnormality, a *Warning* state is generated. When both modalities indicate normal conditions, the driver state is classified as *Driver Alert* (safe). This fusion-based decision-making process improves accuracy and reduces false predictions compared to single-modality systems.

V. SYSTEM ARCHITECTURE

The proposed AI-driven driver behaviour monitoring system is designed using a multimodal architecture that integrates visual information with numerical vehicle and driver health data. The overall system architecture, illustrated in Fig. 1, consists of multiple interconnected modules responsible for data acquisition, preprocessing, model-based analysis, fusion logic, and final decision making.

[Insert Fig. 1 — System Architecture of SafeRider]

Description: Two parallel pipelines (image and numerical) feed a fusion layer. Left pipeline: webcam → face crop (Haar cascade) → 224×224 RGB preprocessing → MobileNetV2 transfer-learned CNN → eye-state / mouth-state outputs. Right pipeline: 7 raw sensor features → engineered features → StandardScaler → ExtraTrees classifier → 5-class label. Both outputs flow into a rule-based fusion layer that emits a final verdict (Safe / Warning / Critical), which the Flask backend returns to the React dashboard for visual and text-to-speech alerting.

A. Data Acquisition Module

The system acquires input data from two primary sources.

Visual Data Acquisition — Facial images of the driver are captured by a camera positioned inside the vehicle. The image dataset includes different driver states such as *open eyes*, *closed eyes*, and *yawning*, which represent visual indicators of alertness and fatigue.

Numerical Data Acquisition — Numerical inputs include vehicle parameters (speed, RPM, throttle position, brake position, steering angle) and driver-health parameters (heart rate, stress level). Collectively, these seven raw features summarise the driver's driving behaviour and physiological condition.

B. Preprocessing Module

To ensure effective model performance, the acquired data undergo preprocessing before being passed to the learning models.

Image Preprocessing — A Haar-cascade face detector first crops the face region; the cropped image is resized to 224×224 pixels and rescaled to the $[-1, 1]$ range expected by MobileNetV2.

Numerical Data Preprocessing — Missing or inconsistent values are imputed, six engineered features are derived from the seven raw inputs, and the resulting 13-feature vector is standardised using StandardScaler. During training, SMOTE ($k = 5$) is applied only on the training fold to address class imbalance without leaking into validation.

C. Image-Based Behaviour Analysis

The preprocessed facial image is fed into a MobileNetV2-based CNN whose ImageNet-pretrained backbone is paired with a custom classifier head consisting of a global average pooling layer, batch normalisation, three dropout-regularised dense layers ($192 \rightarrow 64 \rightarrow$ output units), and an output layer that is either softmax (3 classes) or a pair of sigmoids (multilabel). The network is trained in two phases: in Phase 1 the backbone is frozen and only the head is trained for up to 60 epochs with Adam (learning rate 1×10^{-3}); in Phase 2 the top 25 % of the backbone is unfrozen and the network is fine-tuned for up to 40 epochs with Adam (1×10^{-5}), keeping batch-normalisation layers locked. At inference, 10-pass test-time augmentation is applied to stabilise predictions.

D. Numerical Data Analysis

The preprocessed numerical data are processed using an ExtraTrees classifier. This ensemble model was selected from a 5-fold cross-validation shoot-out involving tuned Random Forest, soft VotingClassifier, StackingClassifier, XGBoost, HistGradientBoosting, and GradientBoosting candidates. ExtraTrees achieved the highest mean CV accuracy and was selected as the production estimator. The model classifies the driver's condition into five categories: *attentive*, *distracted*, *drowsy*, *aggressive*, and *eco-driving*.

E. Fusion Logic and Decision Module

The fusion module integrates the outputs of the CNN and ExtraTrees models using a deterministic priority-ordered rule set. *Risky* sensor labels are defined as *drowsy*, *distracted*, and *aggressive*. The fusion ladder is evaluated top-down and the first matching rule wins:

1. eyes closed + yawning + risky sensor → **Critical: Severe Drowsiness**
2. eyes closed + yawning → **Critical: Driver Drowsy**
3. eyes closed + risky sensor → **Critical: Driver Drowsy**
4. eyes closed → **Warning: Eyes Closed**
5. yawning + risky sensor → **Warning: Yawning + Risky Pattern**
6. yawning → **Warning: Yawning Detected**
7. risky sensor only → **Warning: Risky Driving Pattern**
8. otherwise → **Driver Alert (Safe)**

F. Driver State Prediction

Based on the fusion logic, the final driver-state prediction is emitted along with a severity tag (*safe*, *warning*, *critical*) which the frontend uses to drive colour-coded UI cues and a text-to-speech alert. The modular architecture supports real-time inference and is suitable for integration with advanced driver assistance systems.

VI. METHODOLOGY

The proposed methodology develops an intelligent driver behaviour monitoring system by integrating visual and numerical data through a multimodal learning approach [8]. The pipeline consists of five major stages: data collection, preprocessing, model development, fusion-logic implementation, and driver-state prediction.

A. Data Collection

Image Dataset — A curated dataset of 301 facial images covering *closed eyes*, *open eyes*, and *yawning* states is used to train and evaluate the CNN. The images are split 80/20 into training and validation subsets.

Numerical Dataset — A vehicle dataset of 1,200 rows is constructed, each row containing the seven raw features described in Section V-A together with one of five class labels (*attentive*, *distracted*, *drowsy*, *aggressive*, *eco-driving*). The dataset is stratified-split 80/20 into training and hold-out sets.

B. Data Preprocessing

Image preprocessing consists of Haar-cascade face cropping, resize to 224×224 , and rescaling to $[-1, 1]$. Numerical preprocessing consists of missing-value imputation, engineered-feature computation, StandardScaler normalisation, and SMOTE oversampling ($k = 5$) on the training fold only.

C. Model Development

CNN-Based Image Classification — A MobileNetV2 backbone pretrained on ImageNet is paired with a custom head (GAP $\rightarrow$ BatchNorm $\rightarrow$ Dropout(0.45) $\rightarrow$ Dense(192, ReLU, $L2 = 5 \times 10^{-4}$) $\rightarrow$ Dropout(0.30) $\rightarrow$ Dense(64, ReLU, $L2 = 5 \times 10^{-4}$) $\rightarrow$ Dropout(0.20) $\rightarrow$ Dense output). Two head variants are produced: a 3-class softmax for legacy compatibility and a multilabel pair of sigmoid outputs for fine-grained eye and mouth analysis. Training uses sparse categorical (or binary) cross-entropy with class-weighted loss, EarlyStopping (patience = 15 in Phase 1, 10 in Phase 2), and ModelCheckpoint. The phase that achieves the higher validation accuracy is kept as the production model.

[Insert Fig. 2 — CNN Architecture for Facial Behaviour Detection]

Description: MobileNetV2 backbone block followed by GlobalAveragePooling $\rightarrow$ BatchNorm $\rightarrow$ Dropout/Dense layers (192 $\rightarrow$ 64) $\rightarrow$ output layer (softmax or sigmoid pair). Annotate input shape $224 \times 224 \times 3$ and output labels.

ExtraTrees-Based Numerical Classification — A Pipeline of StandardScaler $\rightarrow$ SMOTE ($k = 5$) $\rightarrow$ ExtraTreesClassifier is trained on the engineered 13-feature vector and compared against tuned Random Forest, soft VotingClassifier, StackingClassifier, XGBoost, HistGradientBoosting, and GradientBoosting candidates using stratified 5-fold cross-validation. ExtraTrees is selected as the best estimator and saved to disk.

[Insert Fig. 3 — ExtraTrees Sensor Classifier Pipeline]

Description: Input feature vector (7 raw + 6 engineered) $\rightarrow$ StandardScaler $\rightarrow$ SMOTE oversampling $\rightarrow$ ExtraTrees ensemble (multiple decision trees) $\rightarrow$ majority vote $\rightarrow$ class label.

D. Fusion Logic Implementation

The fusion logic implements the priority-ordered rule set documented in Section V-E. The deterministic ladder is preferred over a learned meta-classifier because it (i) is interpretable to safety reviewers, (ii) requires no additional labelled fusion data, and (iii) can be hot-swapped via configuration without retraining either base model.

E. Driver State Prediction and Output

The final driver-state prediction is generated from the fusion ladder and returned to the frontend together with model confidences, raw probabilities, and a request-correlation ID. The multimodal approach improves detection accuracy and robustness compared to single-modality systems, making it suitable for real-time driver monitoring applications.

VII. IMPLEMENTATION

The proposed SafeRider system is implemented as a Flask-based Python backend serving a React frontend. The overall implementation is divided into six main stages: data collection and preparation, model training, backend integration, frontend development, fusion-logic implementation, and testing.

A. Data Collection and Preparation

Because the collection of real driver monitoring data is constrained by privacy and safety, simulated and publicly available open-source datasets are used. The visual dataset contains 301 labelled facial images for *closed eyes*, *open eyes*, and *yawning*; the numerical dataset contains 1,200 rows of vehicle and physiological features. Both datasets are preprocessed as described in Section VI-B.

B. Model Training

The CNN is implemented in TensorFlow-Keras (TF-CPU 2.21.0) using transfer learning on MobileNetV2 with the two-phase schedule described in Section VI-C. The ExtraTrees classifier is implemented in scikit-learn (≥ 1.5) with the imbalanced-learn Pipeline to make SMOTE participate in cross-validation correctly. Trained artefacts are

persisted to disk: `driver_multilabel_cnn.keras` (and a softmax fallback `driver_custom_cnn.keras.h5`) for the CNN, and `random_forest_driver.pkl` for the sensor model. Companion metadata JSON files (`cnn_metadata.json`, `cnn_multilabel_metadata.json`, `rf_metadata.json`) record per-class metrics, training-time hyperparameters, and feature importances.

C. Backend Integration (Flask API)

The Flask backend is the main processing unit of the system. It loads both trained models once at startup with a warm-up pass and exposes five HTTP endpoints:

Endpoint	Method	Purpose
/	GET	Service metadata
/health	GET	Model registry, classes, accuracy metrics
/predict_image	POST	CNN visual classification
/predict_numerical	POST	ExtraTrees sensor classification
/predict_fusion	POST	Combined visual + sensor verdict

Production hardening includes env-driven configuration (host, port, CORS origins, thresholds), structured logging via a rotating file handler (2 MB per file, 5 backups), per-request X-Request-ID correlation, an 8 MB upload size cap, extension and MIME validation, request-scoped filenames (no raw filename trust), and centralised error handlers that map `ValidationError` → 400 and any other exception to a generic 500 without stack-trace leakage. Responses are returned as JSON for direct consumption by the frontend.

D. Frontend Development

The frontend is a React single-page dashboard served on port 3000. Users can capture an image from the in-cabin webcam (or upload one) and enter numerical sensor values through accessible progress-filled sliders. The dashboard sends multipart POST requests to the `/predict_fusion` endpoint and renders the response in real time. Prediction results, CNN confidence, ExtraTrees label, and a final verdict (*Driver Alert*, *Warning*, *Critical*) are displayed with colour-coded cues (green / yellow / red), pulse animations on critical states, an aria-live region for screen readers, and a text-to-speech announcement of the verdict.

E. Fusion Logic Implementation

Fusion is implemented as a deterministic ladder in `utils/inference.py`. The multilabel path is preferred and uses CNN-provided eye and mouth states together with the ExtraTrees label; the legacy softmax path is retained as a fallback and uses a configurable CNN-confidence threshold (default 0.7). The full priority-ordered rule set is documented in Section V-E.

F. Testing and Validation

The complete system is tested using held-out data and end-to-end smoke tests. Backend responses are verified using Postman, and real-time results are checked through the React UI. Integration tests confirm correct verdict generation across all eight fusion branches. The integrated system achieves an overall fusion accuracy of approximately **96 %** on held-out multilabel test data, confirming effective driver-state detection and real-time alert generation.

VIII. DESIGN AND RESULT

A. CNN Model Results

The MobileNetV2-based CNN is used to classify driver facial states. Two evaluation modes are reported. The legacy 3-class softmax variant achieves a plain validation accuracy of **95.56 %**; applying 10-pass test-time augmentation keeps the same operating point. The multilabel variant achieves an eye-open accuracy of **97.51 %**, a mouth-yawn accuracy of **97.71 %**, and a *both-correct* combined accuracy of **96.05 %** on the held-out validation split.

[Insert Fig. 4 — Confusion Matrix of CNN Model (multilabel head)]

Description: 4×4 matrix or two 2×2 matrices showing eye-state (open vs closed) and mouth-state (yawn vs neutral) confusion counts on the validation set.

B. ExtraTrees Sensor Model Results

The ExtraTrees classifier classifies driving behaviour into five categories. Across the model shoot-out, ExtraTrees achieved the highest 5-fold cross-validation accuracy of **94.48 %**, narrowly outperforming a tuned Random Forest (94.06 %), the StackingClassifier (94.06 %), VotingClassifier (93.89 %), XGBoost (93.81 %), HistGradientBoosting (93.31 %), and GradientBoosting (92.22 %). On the stratified hold-out test set, the selected ExtraTrees model achieves an accuracy of **92.92 %**. The top three feature importances are *stress_level* (14.90 %), *physio_load* (14.70 %), and *heart_rate* (11.99 %), confirming that the engineered physiological features carry substantial predictive signal.

[Insert Fig. 5 — Confusion Matrix of the ExtraTrees Sensor Classifier]

Description: 5×5 confusion matrix over classes aggressive, attentive, distracted, drowsy, eco-driving computed on the 240-row hold-out test set.

Table I. Summary of model performance.

Model	Dataset	Metric	Score
MobileNetV2 (softmax)	301 face images (3 classes)	Validation accuracy	95.56 %
MobileNetV2 (multilabel)	301 face images (eye + mouth)	Combined accuracy	96.05 %
ExtraTrees (SMOTE)	1,200 rows (5 classes)	5-fold CV	94.48 %
ExtraTrees (SMOTE)	1,200 rows (5 classes)	Hold-out test	92.92 %
Fused verdict	Multimodal	End-to-end	≈ 96 %

IX. CONCLUSION

This paper presents *SafeRider*, an AI-based hybrid system for real-time driver behaviour monitoring that fuses transfer-learned visual analysis with engineered-feature sensor analysis. The combination of a MobileNetV2 CNN with a multilabel head for facial cues and an SMOTE-balanced ExtraTrees classifier for vehicle and physiological data enables accurate detection of driver states such as *attentive*, *drowsy*, *distracted*, *aggressive*, and *eco-driving*.

The deterministic fusion ladder improves accuracy and reduces false detections compared to single-modality baselines [17], [19], while remaining fully interpretable to safety reviewers. The Flask backend efficiently manages predictions with production-style hardening, and the React frontend provides a clear, accessible, and audibly alerting user interface. Experimental results achieve a CNN combined accuracy of **96.05 %**, a sensor classifier hold-out accuracy of **92.92 %**, and an end-to-end fusion accuracy of approximately **96 %**, demonstrating the system's reliability and effectiveness for driver-safety applications.

X. FUTURE SCOPE

Although the current system uses simulated and curated open-source data, several enhancements can be made for real-world deployment:

9. **Integration with Real-Time Sensors** — Use IoT devices such as smartwatches, ECG sensors, and OBD-II vehicle telematics to collect live driver data.
10. **Advanced Deep Learning Backbones** — Replace MobileNetV2 with EfficientNetV2 or a video-based ConvNeXt model to improve robustness under varying lighting and environmental conditions.
11. **Edge and Cloud Deployment** — Deploy the system on edge devices such as Raspberry Pi or NVIDIA Jetson for in-vehicle processing, with cloud support for fleet-scale data analysis.

12. **Adaptive and Continual Learning** — Enable the system to learn continuously and adapt to different drivers and driving behaviours via on-device fine-tuning.

13. **ADAS Integration** — Connect the system with Advanced Driver Assistance Systems to automatically trigger safety actions such as speed limitation, lane-keep assist, or emergency braking [6], [7].

XI. REFERENCES

- [1] A. G. Wheaton, "Drowsy driving and risk behaviors—10 states and Puerto Rico 2011–2012," *MMWR Morb Mortal Wkly Rep*, vol. 63, no. 26, pp. 557–562, 2014.
- [2] NHTSA. Accessed: Feb. 27, 2024. [Online]. Available: <https://www.nhtsa.gov/risky-driving/drowsy-driving>
- [3] Canalys. Accessed: Feb. 20, 2024. [Online]. Available: <https://www.canalys.com/newsroom/canalys-autonomous-driving-starts-to-hit-mainstream-as-35-million-new-cars-had-level-2-features-in-q4-2020>
- [4] O. Carsten, F. C. H. Lai, Y. Barnard, A. H. Jamson, and N. Merat, "Control task substitution in semiautomated driving: Does it matter what aspects are automated?" *Hum. Factors*, vol. 54, no. 5, pp. 747–761, Oct. 2012, doi: 10.1177/0018720812460246.
- [5] J. W. Jenness, N. D. Lerner, S. Mazor, J. Osberg, and S. Tefft, "Use of advanced in-vehicle technology by young and older early adopters. Survey results on adaptive cruise control systems," NHTSA, U.S. Dept. Transp., Washington, DC, USA, Tech. Rep. DOT HS 810 917, 2008.
- [6] IDTechEx. Accessed: Feb. 15, 2024. [Online]. Available: <https://www.idtechex.com/ja/research-article/regulations-drivers-for-mandating-driver-monitoring-systems/30322>
- [7] U.S. Congress, S. 1406. Accessed: Feb. 15, 2024. [Online]. Available: <https://www.congress.gov/bill/117th-congress/senate-bill/1406/text>
- [8] M. Q. Khan and S. Lee, "A comprehensive survey of driving monitoring and assistance systems," *Sensors*, vol. 19, no. 11, p. 2574, Jun. 2019, doi: 10.3390/s19112574.
- [9] S. Titare, S. Chinchghare, and K. N. Hande, "Driver drowsiness detection and alert system," *Int. J. Sci. Res. Comput. Sci., Eng. Inf. Technol.*, vol. 7, pp. 583–588, Jun. 2021, doi: 10.32628/cseit2173171.
- [10] C. Solomon and Z. Wang, "Driver attention and behavior detection with Kinect," *J. Image Graph.*, vol. 3, no. 2, pp. 1–6, 2015.
- [11] J. D. Ortega, N. Köse, P. N. Cañas, M.-A. Chao, A. Unnervik, M. Nieto, O. Otaegui, and L. Salgado, "DMD: A large-scale multi-modal driver monitoring dataset for attention and alertness analysis," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Jan. 2020, pp. 387–405, doi: 10.1007/978-3-030-66823-5_23.
- [12] B. K. Savas and Y. Becerikli, "Real time driver fatigue detection system based on multi-task ConNN," *IEEE Access*, vol. 8, pp. 12491–12498, 2020, doi: 10.1109/ACCESS.2020.2963960.
- [13] E. Vural, "Drowsy driver detection through facial movement analysis," in *Proc. Int. Workshop Hum.-Comput. Interact.*, vol. 4796, Cham: Springer, pp. 6–18, doi: 10.1007/978-3-540-75773-3_2.
- [14] K. Dwivedi, K. Biswaranjan, and A. Sethi, "Drowsy driver detection using representation learning," in *Proc. IEEE Int. Advance Comput. Conf. (IACC)*, Feb. 2014, pp. 995–999, doi: 10.1109/IADCC.2014.6779459.
- [15] W. Zhang, Y. L. Murphey, T. Wang, and Q. Xu, "Driver yawning detection based on deep convolutional neural learning and robust nose tracking," in *Proc. IJCNN*, Jul. 2015, pp. 1–8, doi: 10.1109/ijcnn.2015.7280566.
- [16] Y. Ji, S. Wang, Y. Zhao, J. Wei, and Y. Lu, "Fatigue state detection based on multi-index fusion and state recognition network," *IEEE Access*, vol. 7, pp. 64136–64147, 2019, doi: 10.1109/ACCESS.2019.2917382.

- [17] K. Zhang, S. Wang, N. Jia, L. Zhao, C. Han, and L. Li, "Integrating visual large language model and reasoning chain for driver behavior analysis and risk assessment," *Accident Anal. Prevention*, vol. 198, Apr. 2024, Art. no. 107497, doi: 10.1016/j.aap.2024.107497.
- [18] J. Chen, S. Dey, L. Wang, N. Bi, and P. Liu, "Attention-based multi-modal multi-view fusion approach for driver facial expression recognition," *IEEE Access*, vol. 12, pp. 137203–137221, 2024, doi: 10.1109/ACCESS.2024.3462352.
- [19] J.-W. Kim, C. I. Nwakanma, D.-S. Kim, and J.-M. Lee, "Intelligent face recognition on the edge computing using neuromorphic technology," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2021, pp. 514–516, doi: 10.1109/ICOIN50884.2021.9333967.
- [20] S. Y. Nikouei, Y. Chen, S. Song, R. Xu, B.-Y. Choi, and T. Faughnan, "Smart surveillance as an edge network service: From haar-cascade, SVM to a lightweight CNN," in *Proc. IEEE 4th Int. Conf. Collaboration Internet Comput. (CIC)*, Oct. 2018, pp. 256–265, doi: 10.1109/CIC.2018.00042.
- [21] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.
- [22] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, Apr. 2006, doi: 10.1007/s10994-006-6226-1.
- [23] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.
- [24] GeeksforGeeks, "Introduction to Convolutional Neural Networks." [Online]. Available: <https://www.geeksforgeeks.org/machine-learning/introduction-convolution-neural-network/>
- [25] GeeksforGeeks, "Random Forest Algorithm in Machine Learning." [Online]. Available: <https://www.geeksforgeeks.org/machine-learning/random-forest-algorithm-in-machine-learning/>