RESEARCH ARTICLE

JENER Journal of Empirical and Non-Empirical Research
Volume 2, Issue 5, Page 502-507 | Article No. 59
DOI: 10.4898/jener.v2i5.a59
ISSN: 7895-9216
Journal homepage: https://jenerjournal.com
Journal email: editor@jenerjournal.com

JENER Journal
JENER Journal of Empirical and Non-Empirical Research

Cubo: Edge AI Device System for Real-Time Driver Distraction Detection Using Geometric Rules and Quantum-Optimized CNNs

Dhruva Valluru[1] & Amogh Gotaparthy[2]

[1]Enloe High School Morrisville, USA
[2]Panther Creek HS Morrisville, USA

ARTICLE INFORMATION

Article history:
Published: May 2026

Keywords:
CNN, Deep Learning, Multi-Layered, Embed- ded AI, QSGD, Distraction Detection, Driver Monitoring

ABSTRACT

Driver distraction is a significant factor in global road deaths, causing over 1,000 fatalities every day. Existing detection technologies are expensive and mostly limited to high- end vehicles. This study proposes a widely deployable, non- intrusive system combining facial landmark tracking, object detection, and deep learning to monitor driver distraction. Using a multi-layered model based on MediaPipe, YOLOv5, and MobileNet CNN, trained on over 14,000 frames and optimized with QSGD, the system achieved 88.1% accuracy with strong temporal consistency. This embedded solution offers real-time, scalable distraction detection to improve road safety.

1. Introduction

Distracted driving remains a widespread and deadly issue worldwide, contributing to approximately 20–30% of all road traffic crashes globally, which translates to over 400,000 deaths annually [1] [2]. Visual, manual, and cognitive distractions—particularly mobile phone use—significantly impair a driver's reaction time, focus, and decision-making, making the road environment increasingly hazardous.

While existing strategies have limited success in reducing distracted driving, new laws are expected to mandate the integration of passive driver monitoring systems in vehicles, presenting promising opportunities for this technology. Cognitive issues such as inattentional blindness and overload significantly affect teen drivers, who are especially vulnerable due to inexperience and frequent mobile device use. We are currently in talks with policymakers and school transportation safety boards in North Carolina, such as the Wake County Public School System (WCPSS), to pilot these systems with high-risk populations. Discussions with vehicle manufacturers, such as GM, are also ongoing to align with future safety standards.

Current in-vehicle systems, including Advanced Driver Assistance Systems (ADAS) and lane departure interventions, largely address the consequences of driver distraction—such as unintentional drifting or loss of vehicle control—rather than targeting the root cause: the distracted driver [3]. This reactive approach has proven insufficient, underscoring the growing need for a proactive solution that prevents distraction through early behavioral intervention.

To address this need, we developed a computational machine learning system named Cubo, powered by a combination of computer vision and neural networks to monitor driver behavior non-intrusively. This system utilizes facial landmark detection and hand tracking to identify signs of distraction, such as phone usage or gaze aversion from the road, and provides instant audio alerts to re-engage the driver. By targeting distraction at its source, Cubo aims to prevent crashes before they happen, ultimately saving lives and shaping the future of safe driving.

2. Methodology

2.1 Data Pre-Processing

2.1.1 Frontal Face and Eye Detection (FFED)

We obtained our data from YouTube, the DMD VicomTECH Driver Monitoring Dataset, and Kaggle. We obtained video files in .mp4 format from YouTube and the DMD VicomTECH dataset, and image files from Kaggle (a well-known platform for data science and machine learning). We used a dataset of video footage from different environments to train the model to recognize variations in lighting, vehicle types, and driver demographics. The videos and images were used to train and support our custom-built pipeline algorithm for Frontal Face and Eye Detection (FFED) [4] [5], which recognizes key facial and optical features of the driver. Our FFED pipeline automatically classified every frame of each video, assigning a clear label to each frame based on the detected behavior, such as "distracted," "looking at the road," and "using phones." Through this process, individual frames were extracted and organized into structured datasets optimized for model training and inference. We used a machine learning framework called MediaPipe to extract relevant visual features from the frames. We modified the framework to detect features of the frames in two separate Cascade Classifier Frameworks, which are subparts of the FFED pipeline. The first classifier was used to detect and crop face regions, and the second classifier was used to detect eye regions for further analysis [6]. Manual quality control checks were performed to ensure accurate bounding boxes and consistent image quality, resulting in a refined set of approximately 8,750 processed images drawn from our total dataset of 14,320 frames, which are divided between distracted and attentive driving categories.

2.1.2 Object Detection and Hand Placement (ODHP)

The facial and eye detection capabilities served as a foundation for our team to develop another customized pipeline algorithm called Object Detection and Hand Placement (ODHP), which monitored both phone usage and hand interaction. The pipeline runs YOLOv5, which operates at high speed and produces accurate results on edge devices, namely the Raspberry Pi [7]. The YOLOv5 model analyzes video frames of the driver, detecting mobile phones by drawing boxes, labeling, and dis- playing respective confidence scores [8]. The ODHP pipeline generates two main outputs after phone detection: a boolean flag indicating the presence of a phone, as well as a positional matrix containing box coordinates. MediaPipe tracks the 21 hand landmarks of drivers (i.e., fingers, palms, and wrist points), which provide three-dimensional spatial information about hand posture and position. The processed data from YOLOv5 and MediaPipe systems undergo normalization and reorganization into structured arrays for additional process- ing. To determine driver phone usage, the system uses the Euclidean distance measurement between phone box center coordinates and hand landmark positions:

$$d = \sqrt{(x_2 - x_1)2 + (y_2 - y_1)2 + (z_2 - z_1)2}$$

The hand landmark coordinates (x_1, y_1, z_1) are used to calculate distance alongside the phone bounding box center coordinates (x_2, y_2, z_2). This proximity-based method provides an estimate of phone holding distance based on hand positions, with a primary focus on the fingertips and wrist areas. The system tested this method with thousands of frames across different lighting conditions, vehicle types, and driver behaviors to establish a reliable threshold of 0.05 in normalized coor- dinates [8]. When the Euclidean distance between the phone and hand landmarks falls below the established threshold, the system identifies driver phone use by triggering a distraction alert that records the incident and produces an immediate audio notification. The ODHP pipeline determines both the presence of mobile devices and their usage by drivers through its combination of object detection and hand placement analysis methods.

2.2 Model Training and Evaluation (Model One)

2.2.1 Facial Attention Analysis Using MediaPipe Face Mesh

To expand on the previously established FFED algorithm, we incorporated MediaPipe's Face Mesh framework into our facial detection system. This framework detects 468 high- resolution 3D facial landmarks per frame at 30 FPS, which allows real-time gaze direction and head pose orientation analysis [9]. These landmarks include key points of the pupils (landmarks 468 & 469), inner and outer eye corners (33, 133, 362, 263), nose tip (1), nasal bridge (6, 168, 197), and jawline/chin base (152, 205, 234, 356, 454). These are then transformed into structured coordinate arrays for each frame. The system generates two orientation vectors from these points: (1) the gaze vector running from the pupil center (468, 469) to the inner eye corners (33, 263) and (2) the head pose vector extending from the nose tip (1) to the chin base
(152). The system maintains continuous vector comparisons against a forward-facing baseline, which was established under neutral head and eye conditions [10].

2.22 Defining Distraction Parameters and Threshold Calibration

From analysis through extracted frames, we established angular thresholds of ±15.2° horizontally and ±9.7° vertically from the baseline gaze direction. When either gaze or head pose breaches the angular thresholds, the system flags the frame as potentially distracted [11]. When a phone is detected (as determined by the ODHP pipeline algorithm), we initiate a gaze-to-object correlation step. This involves projecting a gaze ray from the midpoint of the pupils using the gaze vector and determining whether it intersects the phone's bounding box as computed by YOLOv5. A ray–box intersection test determines if the phone is the subject of visual attention. If this intersection is valid and maintained for over 2 seconds, the event is classified as an active distraction.

2.2.3 Multi-Layered Ensemble Model Combining Face, Eye, Object, and Hand Classification

The model operates all its components (phone detection algorithm, hand landmark tracker, and facial analysis pipeline) simultaneously on each video frame. The system integrates all outputs into a single, multi-layered evaluation framework that performs continuous driver distraction assessment. The synchronized frame- by-frame operation generates a binary output indicating the driver's distraction status as True or False, providing an efficient real-time safety monitoring solution [12]. This geometric rule-based process of Cubo's system represents "Model One."

2.3 Integration of Deep Learning to Enhance Model Accuracy (Model Two)

The limitations of Model One's geometric & rule-based approach were addressed by developing Model Two. Model Two is a data-driven framework that would be used to opti- mize distraction detection on top of Model One. Model Two enhances the initial FFED and ODHP pipelines by utilizing three specialized convolutional neural networks (CNNs) with a MobileNetV3-Large backbone. The design is made to balance accuracy and computational efficiency, making deployment on edge devices like Raspberry Pi 5 easier. Model Two learns complex spatial and appearance features directly from images, enabling it to handle variations in lighting, occlusion, and driver behavior better, making it a more adaptable addition to Model 1 [13].

2.3.1 Dataset Preparation for CNN Training

Training for Model Two was conducted using a master dataset of 14,320 labeled frames curated from diverse driving environments. Through the master dataset, 3,360 face crops, 2,100 eye crops, and 1,725 hand crops were extracted by the FFED pipeline, including face and eye regions isolated by Dlib's HOG + SVM detector and hand crops generated from landmark-driven bounding boxes, which were determined on wrist and fingertip coordinates. Images and crops were standardized to 224×224 pixels, sorted into three RGB channels, and converted to a 0-1 pixel range to ensure consistency. Finally, targeted data augmentation, random brightness and contrast shifts (±20%), Gaussian blur (σ=1.2), rotational jitter (±15°), and horizontal flips, were used upon datasets to improve generalization and simulate real-world variations [14].

2.3.2 Modular CNN Training with Optimization Techniques

Each CNN classifier was trained independently on its re- spective region, i.e., face, eyes, or hands, using binary cross- entropy loss with distraction labels encoded as 0 (attentive) and 1 (distracted). Initial training employed classical optimizers such as Adamax, SGD, RMSprop, and Adam to establish convergence baselines; however, final optimization utilized Quantum Stochastic Gradient Descent (QSGD), a quantum- inspired method that reduces gradient precision to acceler- ate learning and reduce memory footprint [15] [16]. Early stopping halted training after five stagnant epochs to prevent overfitting, enabling efficient capture of various distraction cues across drivers. To assess stability over time, we used the Temporal Consistency Rate (TCR), which measures the frequency with which predictions align with the majority decision within a five-frame window [17].

2.3.3 Stacking Geometric and Deep Learning for Accuracy

When deployed in a sequential ensemble architecture, the system requires both Model One and Model Two to classify a frame as distracted for a positive detection, minimizing false positives while preserving true positive detections. This approach utilized a 70/30 weight split on a labeled test set of 1,368 frames. This hybrid approach combines Model One's geometric rule-based precision with Model Two's CNN-based adaptability. Overall, this combined architecture provides a scalable, real-time solution that effectively detects driver dis- traction in challenging, real-world environments [12].

3. Results

3.1 Face + Phone Detection + Hand MediaPipe (Model One)

Model One used geometric and spatial logic by combining MediaPipe Face Mesh, YOLOv5, and MediaPipe Hands to detect distraction. It computed gaze vectors and head orienta- tion from facial landmarks, identified phones with YOLOv5, and tracked hand positions to determine phone-hand proximity using Euclidean distance. Frames were flagged as distracted when gaze angles exceeded the set limits (as shown in Table 2) and phone-hand distance dropped below the 0.05 threshold, which yielded the highest accuracy (as indicated in Table 1). These parameters achieved 91.3% detection accuracy with 4.5% false positives, promising reliability across a wide range of driving conditions. However, Model One achieved a 31.4% detection accuracy under baseline conditions, demonstrating subpar temporal consistency but multi-angle robustness.

3.2 MobileNetV3 + Quantum QSGD Optimization (Model Two)

Model Two used a deep learning approach to classify distraction based on face, eye, and hand regions. During evaluation, this model achieved frame-by-frame distraction classification accuracies ranging from 62.47% to 71.4%, de- pending on environmental complexity. The CNN optimizers were trained with binary cross-entropy and augmented data to simulate diverse real-world conditions. Using a softmax confidence threshold of 0.65, the QSGD optimizer achieved an F1 score of 0.86 (as shown in Table 3), indicating a strong balance between precision and recall. Additionally, the model's Temporal Consistency Rate (TCR) reached 87.5%, significantly outperforming the 52.3% TCR of Model One alone, further solidifying its ability to provide stable and reliable predictions over time.

3.3 Multi-Layered Ensemble Model System (Model 1 + Model 2)

When the geometric rule-based Model One and the deep learning-based Model Two were combined in a sequential en- semble configuration, overall system performance significantly improved. This method resulted in a significant accuracy boost of 88.1% (as shown in Table 4) on a 1,368-frame dataset, compared to 30–40% with Model One alone. The ensemble achieved an F1 score of 0.86, demonstrating a strong balance between precision and recall. We discarded predictions with Softmax scores below 0.65 to reduce false alerts. Moreover, it maintained the high TCR of 87.5% and further reduced spurious detections caused by non-distracting glances at mir- rors or dashboard elements. This hybrid approach effectively leveraged both the interpretability of geometric rules and the adaptability of CNNs, enhancing robustness in real-world conditions.

3.4 Edge Device System Deployment

The full Cubo system, combining both geometric and CNN models, was deployed on a Raspberry Pi 5, a cost-effective edge AI platform. The software pipeline is integrated with an accelerometer, enabling distraction detection only when the vehicle is in motion, thereby conserving resources and ensuring contextual relevance. The hardware system runs in real-time, powered by a standard vehicle plug, and offers easy, non-invasive installation, making it a scalable solution for both personal and commercial driver monitoring [7].

4. Discussion

4.1 Highlights

Cubo effectively combines geometric rule-based analysis and deep learning to accurately detect driver distraction. The ensemble of Model One and Model Two achieved an accuracy of 88.1% and an F1 score of 0.86, demonstrating improved performance over geometric-only methods. A Temporal Con- sistency Rate (TCR) of 87.5% indicates stable detection across sequential frames, thereby reducing false positives from brief, non-distracted behaviors. Confidence thresholding further re- fines predictions by filtering uncertain outputs, enhancing re- liability. These results confirm that integrating explicit spatial reasoning with learned visual features strengthens the system's adaptability to real-world variability.

4.2 Limitations

However, challenges remain in achieving consistent detec- tion under diverse lighting, occlusion, and driver demographic conditions. Static thresholds limit Model One's geometric approach and cannot capture nuanced distraction behaviors, necessitating the development of a CNN-based Model Two. The performance of Model Two depends heavily on the quality and diversity of

labeled training data, which currently lacks representation of all real-world scenarios. Hardware limitations on embedded platforms require balancing model complexity and inference speed, restricting the use of more advanced algorithms. Moreover, rapid hand movements, occlu- sions, and similar driver gestures can occasionally lead to false positives or negatives. Future improvements should focus on expanding datasets, enhancing occlusion handling, and adapt- ing models for real-time embedded deployment. Overall, Cubo provides a robust foundation but requires ongoing refinement for comprehensive, real-world application.

5. Related Works

Other similar works are limited in important ways. For ex- ample, Yongsheng Zhang et al. examined right-turn distracted driving behavior in 581 clips from the SHRP2 naturalistic driving dataset. However, their method utilized only shallow machine learning methods, including logistic regression and random forests, on predefined external environmental features (e.g., intersection type, bike lanes, level of traffic volume) [18]. Moreover, labels were manually annotated, and distraction was inferred from the ecological context rather than driver behavior, which limited the real-time application. Cubo is the most relevant, as it is capable of real-time, correlated facial and hand landmarking, and object detection to provide a direct measure of visual and mental distraction while driving on the roads.

6. Conclusion

In this work, we developed and evaluated Cubo, an AI- powered distracted driving detection system that leverages a multi-layered approach combining geometric rule-based mod- els with advanced deep learning classifiers. Our system inte- grates facial landmark analysis, hand and phone detection, and a novel CNN architecture based on MobileNetV3, optimized with Quantum Stochastic Gradient Descent, to deliver robust, real-time distraction monitoring suitable for deployment on embedded edge devices. Empirical results demonstrate that the hybrid ensemble of Model One's spatial heuristics and Model Two's learned visual representations significantly im- proves detection accuracy and temporal consistency compared to geometric-only methods. This comprehensive framework offers a non-invasive, scalable solution for early identifica- tion of driver distraction, aiming to reduce traffic accidents by enabling timely interventions. Future directions include expanding dataset diversity, refining temporal modeling tech- niques, and integrating additional sensor modalities further to enhance system reliability and adaptability across varying driving environments. Ultimately, Cubo represents a promising step toward safer roads through intelligent, proactive driver behavior monitoring.

7. Figures and Tables

Fig. 1: The bottom left picture displays a camera frame. The top left pictures show the cropped face and hand/phone captures. The right picture shows processed facial landmarks, hand landmarks, and a phone bounding box.

Fig. 2: The bottom left picture displays a camera frame. The top left pictures show the cropped face and hand captures. The right picture shows processed facial landmarks and hand landmarks.

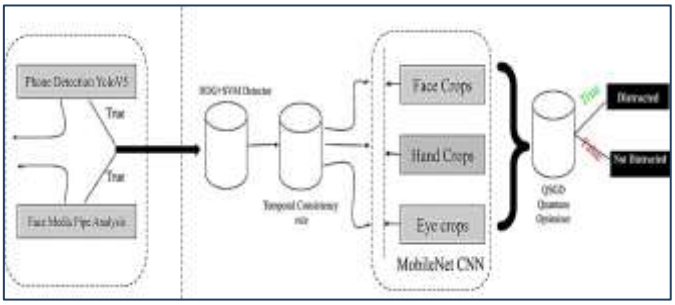

Fig. 3: Multi-Layered Ensemble Model UML Diagram

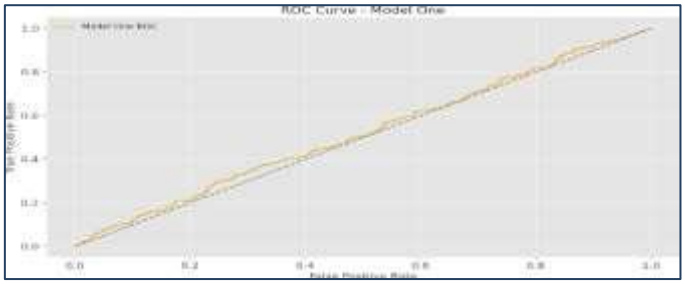

Fig. 4: ROC Curve for Model One Performance

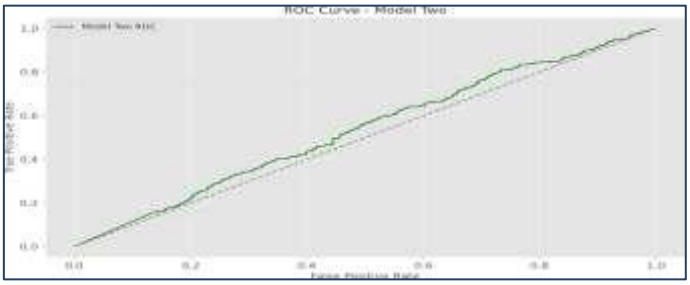

Fig. 5: ROC Curve for Model Two Performance

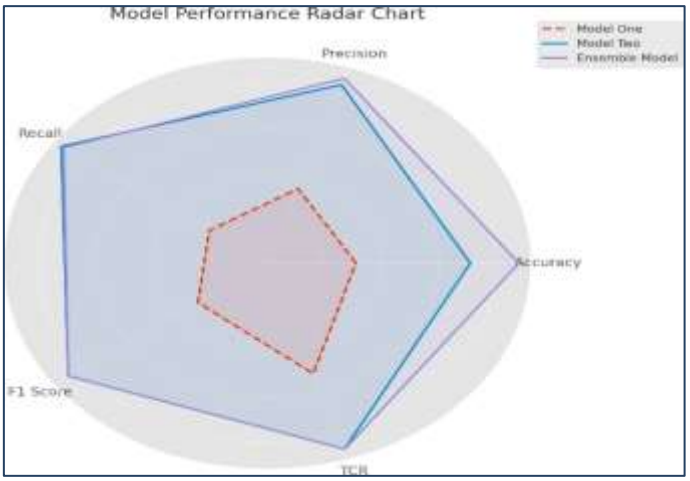

Fig. 6: Radar Chart for All Models Performance

Table I: Effect of Euclidean Distance Threshold on Accuracy

Euclidean Distance Threshold	Accuracy (%)
0.10	74.82
0.08	81.27
0.06	85.93
0.05	89.40
0.04	84.12
0.03	79.96

Table II: Detection Accuracy at Varying Angular Thresholds

Gaze Threshold (° H/V)	Calibration Vector	Accuracy (%)
±10 / ±7.5	Nose–Chin	83.24
±12 / ±8.5	Pupils–Eye Corners	87.11
±15.2 / ±9.7	Pupils–Eye Corners	91.30
±18 / ±11	Nose–Chin	85.67
±20 / ±12.5	Pupils–Eye Corners	79.34

Table III: Face/Eye/Hand Classification Results

Optimizer	Accuracy (%)	Recall	F1 Score
Adam	81.54	0.804	0.812
SGD	77.38	0.776	0.778
RMSprop	80.02	0.791	0.798
QSGD	88.10	0.874	0.860

Table IV: Combined Model Performance at Varying Weights

Model Weights (Rule/CNN)	Accuracy (%)	Recall	F1 Score
50 / 50	84.67	0.823	0.809
60 / 40	86.92	0.861	0.845
70 / 30	88.10	0.868	0.856
90 / 10	85.34	0.849	0.832

References

[1] World Health Organization. Global status report on road safety 2023: Distraction and crashes. https://www.who.int/publications/ global-status-report-road-safety-2023, 2023. Accessed: Jul. 24, 2025.

[2] National Highway Traffic Safety Administration. Distracted driving fact sheet. https://www.nhtsa.gov/risky-driving/distracted-driving, 2025. Accessed: Jul. 24, 2025.

[3] S. Yang and R. Parry. Cell phone use while driving: Risk implications for organizations. *IEEE Technology and Society Magazine*, 33(4):65–72, 2014.

[4] Kumar, A. Gupta, and A. Chauhan. Driver distraction detection. *International Journal of Innovative Research in Technology (IJIRT)*, 11(12):1603–1613, May 2025.

[5] Ezzouhri, Z. Charouh, M. Ghogho, and Z. Guennoun. Howdrive 3d: Driver distraction dataset. IEEE Dataport, 2021. Accessed: Sept. 22, 2021.

[6] R. Florez, F. Palomino-Quispe, A. B. Alvarez, R. J. Coaquira-Castillo, and J. C. Herrera-Levano. A real-time embedded system for driver drowsiness detection based on visual analysis of the eyes and mouth using convolutional neural network and mouth aspect ratio. *Sensors*, 24(11):6261, 2024.

[7] T. Wang, J. Guo, B. Zhang, G. Yang, and D. Li. Deploying ai on edge: Advancement and challenges in edge intelligence. *Mathematics*, 13(11):1878, 2025.

[8] G. Peserico and A. Morato. Performance evaluation of yolov5 and yolov8 object detection algorithms on resource-constrained embedded hardware platforms for real-time applications. In *Proc. IEEE Interna- tional Conference*, pages 1–7, 2024.

[9] B. N, K. Bhuvana, K. Supraja, and H. Tella. Enhancing hand gesture recognition with mediapipe and svm model on custom dataset. In *International Conference on Computing, Communication, and Intelligent Systems (ICCUBEA)*, pages 1–4, Pune, India, 2023.

[10] Y. Wu and Q. Ji. Robust facial landmark detection under significant head poses and occlusion. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3658–3666, 2015.

[11] K. Wilson and A. Garcia. Benchmarks for in-vehicle real-time distrac- tion detection using embedded platforms. *Journal of Real-Time Image Processing*, 19(2):150–160, 2022.

[12] M. Fresta, A. Patti, F. Cacace, D. Giordano, A. Chella, and F. Furnari. Deep learning-based real-time driver cognitive distraction detection. *IEEE Access*, 13:26589–26607, 2025.

[13] M. Rybczak and K. Kozakiewicz. Deep machine learning of mobilenet, efficient, and inception models. *Algorithms*, 17(3):96, 2024.

[14] Z. Wang, S. H. Nelaturu, and S. Amarasinghe. Accelerated cnn training through gradient approximation. In *Proc. 2nd Workshop on Energy Efficient Machine Learning and Cognitive Computing (EMC2)*, pages 31–35, 2019.

[15] Ayush. Optimizers in deep learning: A detailed guide. https://www. analyticsvidhya.com/blog/2025/04/optimizers-in-deep-learning/, 2025.

[16] Accessed: Jul. 28, 2025.

[17] D. Alistarh, D. Grubic, R. Tomioka, and M. Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, pages 1709–1720, 2017.

[18] H.-S. Kim, M. Son, M. Kim, M.-J. Kwon, and C. Kim. Breaking tem- poral consistency: Generating video universal adversarial perturbations using image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4302–4311, 2023.

[19] L. Bin, R. Yue, and Y. Zhang. The influence of different factors on right-turn distracted driving behavior at intersections using naturalistic driving study data. *IEEE Access*, PP:1, Sept. 2019.