International Journal of Science, Strategic Management and Technology
Volume **02** Issue **05** May-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

Drowsiness Detection using Generative AI: A Practical Approach for Real-Time Driver Safety

Raj Shekhar Mishra
BTech(Information Technology) Department of Information Technology
G. Noida - 201310, India
mishrarajshekhar153@gmail.com

Mr. Ankur Chaudhary
Assistant Professor
Department of Information Technology
G. Noida - 201310, India

https://doi.org/10.55041/ijsmt.v2i5.253

Cite this Article: Mishra, R. S. (2026). Drowsiness Detection using Generative AI: A Practical Approach for Real-Time Driver Safety. International Journal of Science, Strategic Management and Technology, *02*(05). https://doi.org/10.55041/ijsmt.v2i5.253

***Abstract*—**Let me start with a simple truth — drowsy driving kills. Every year, thousands of people lose their lives because a driver fell asleep at the wheel. Traditional drowsiness detection systems try to solve this problem, but they have a fundamental flaw — they treat every driver the same way. Fixed thresholds for eye closure and yawning simply do not work for everyone. A driver with naturally small eyes gets false alerts constantly. Another driver who is genuinely drowsy but has wide eyes slips through. In this paper, we present a completely different ap- proach using Generative AI, specifically a Conditional Generative Adversarial Network (cGAN). Our system learns each driver's unique alert face during the first few minutes of driving. Then it continuously compares their real-time face against a personalized baseline. If something looks off — even subtle drowsiness before the eyes fully close — the system triggers an alert. We trained and tested this system on the NTHU Drowsy Driver Dataset, and the results were impressive — 94.8% detection accuracy with only 2.1% false positives. The system runs in under 150 milliseconds per frame on standard hardware. This is not just another drowsiness detector. This is a system that actually adapts to the person behind the wheel.

Index Terms—Drowsiness Detection, Generative AI, GAN, Driver Safety, Real-time Monitoring, Conditional GAN, Facial Recognition.

I. INTRODUCTION

Let me share something that keeps me awake at night — the sheer number of preventable accidents caused by drowsy driving. According to the National Highway Traffic Safety Administration, around 100,000 police-reported crashes every year involve a drowsy driver. That is more than 270 crashes every single day. And the really sad part? About 1,550 people die in these crashes annually. Thousands more are injured.

I have been on long drives where I felt my eyelids getting heavy. Most of us have. The problem is universal, and every- one thinks they can beat it. "I will just push through," they say. Sometimes they do. Sometimes they do not.

The research community has been working on this problem for years. Most systems use a camera to watch the driver's face. They track how long the eyes stay closed — a metric called PERCLOS. They count yawns. They monitor head position. But here is the problem — they use the same thresholds for everyone. Eyes closed for more than 2 seconds? Trigger alert. But what about a driver who naturally blinks slowly? They get false alerts constantly and eventually turn the system off. What about a driver who is genuinely drowsy but keeps their eyes partially open? The system never triggers.

This is where my thinking took a different direction. What if a system could learn what *your* alert face looks like? Not an average person's alert face — your specific face. Your eye shape, your blink pattern, your natural facial expressions. Then it could detect when you deviate from your own normal.

That is exactly what generative AI enables. Generative Adversarial Networks (GANs) are really good at learning the distribution of data. For us, that means learning the distribution of your alert face. Once the system knows what you normally look like, it can tell when something is wrong — even subtle changes that happen before your eyes fully close.

In this paper, I present a drowsiness detection system built on Conditional GANs. It personalizes to each driver, handles natural variations gracefully, and achieves state-of- the-art accuracy. My goal is simple — to build something that could actually save lives on the road.

II. PROBLEM STATEMENT

After spending months studying existing drowsiness detec- tion systems and testing them on real data, I identified six fundamental problems that nobody has solved properly:

First — fixed thresholds do not work for everyone. Let me give you a concrete example. One of my test subjects has naturally narrow eyes because of their facial structure. The traditional PERCLOS algorithm flagged this person as drowsy 40% of the time when they were perfectly alert. On the other hand, another subject could be genuinely drowsy — barely keeping their eyes open — and the system would not trigger because their eyes still had a larger opening than the first subject's normal eyes. The same threshold simply cannot work for both people.

Second — lighting changes break traditional systems. Driving is not a controlled laboratory environment. The sun moves behind clouds. You drive under a bridge. You enter a tunnel. You drive toward the setting sun. Traditional computer vision models panic in these situations. Eye detection fails. Feature extraction becomes noisy. False alerts spike.

Third — glasses and occlusions are not handled well. Millions of drivers wear glasses. Some wear sunglasses. Some wear masks. Some have beards that partially cover their mouth. Traditional eye-blink detection algorithms often fail completely when glasses introduce glare or reflections. Yawning detection fails when a beard covers the mouth.

Fourth — existing systems do not adapt over time. Drowsiness is progressive. A driver becomes more tired gradually over a long journey. But traditional systems treat each moment independently. They have no memory, no sense of accumulating fatigue. A slightly drowsy driver at hour 4 of a road trip might look similar to a fully alert driver at hour 1. The system misses the context.

Fifth — false alert fatigue is a real problem. I spoke with several fleet managers who had deployed drowsiness detection systems. Their drivers hated them. The systems triggered constantly for false reasons — a driver adjusting the radio, turning to talk to a passenger, scratching their face. After a week, everyone turned the alerts off. A safety system that drivers disable is worse than no system at all.

Sixth — no personalization. This is the heart of the problem. Every existing system I could find uses a one-size-fits-all model. Train on 100 drivers, deploy on the 101st. But every driver has unique facial features, unique blinking patterns, unique expressions. A system that does not personalize cannot perform optimally for anyone.

These six problems convinced me that traditional ap- proaches have hit a wall. We needed something fundamentally different — something that could learn, adapt, and personalize. That is why I turned to Generative AI.

III. LITERATURE REVIEW

Before building my own system, I spent considerable time understanding what others had already done. The good news is that many brilliant researchers have worked on this problem. The bad news is that none of them had solved the personalization challenge yet.

A. *Traditional Vision-Based Approaches*

The earliest drowsiness detection systems used simple computer vision techniques. They would detect the driver's face using Haar cascades, then apply color thresholding to find the eyes. The eye closure ratio was calculated by measuring the height-to-width ratio of the eye region. When this ratio fell below a threshold for a certain number of consecutive frames, an alert triggered.

A 2004 study by the Federal Highway Administration established PERCLOS (Percentage of Eye Closure) as a reliable drowsiness metric. They found that a PERCLOS value above 0.15 — meaning the eyes are closed more than 15% of the time — correlates strongly with drowsiness. This finding influenced almost every system that came after.

But PERCLOS has limitations. It measures only the eyes. A drowsy driver might keep their eyes open but lose focus, slow their reaction time, or make poor decisions. PERCLOS misses these early warning signs.

B. *Machine Learning Approaches*

When deep learning became popular, researchers applied it to drowsiness detection. A 2020 study used a Convolutional Neural Network to classify individual frames as alert or drowsy, achieving about 85% accuracy. Later studies used temporal models like LSTMs to consider sequences of frames, pushing accuracy to around 90%.

But these models still had the same fundamental limitation — they were trained on generic data and did not personalize.

C. *The Geometry-Based Alternative*

Some researchers took a different approach. Instead of training neural networks, they used facial landmarks detected by libraries like Dlib or MediaPipe. They computed geometric ratios — the Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR) — and set thresholds on these ratios.

EAR is calculated as the distance between the upper and lower eyelids divided by the distance between the left and right eye corners. When a person closes their eyes, EAR drops dramatically. MAR works similarly for yawning.

These geometric approaches are simple, fast, and interpretable. But they still face the threshold personalization problem. Every driver has slightly different baseline EAR values.

D. *The Emergence of Generative AI*

Generative AI entered the scene around 2014 when Ian Goodfellow introduced Generative Adversarial Networks. The idea was brilliant — pit two neural networks against each other. A generator creates fake data, and a discriminator tries to tell real from fake. They train together, and eventually the generator creates remarkably realistic synthetic data.

Researchers have applied GANs to many problems — generating realistic face images, translating images from day to night, even creating art. But surprisingly few have applied them to drowsiness detection.

A 2022 study from China was the first to explore GANs for driver drowsiness. They trained a standard GAN on alert face images and used the discriminator's output as a drowsiness indicator. Their accuracy reached 92%, promising but still room for improvement. They also did not personalize to individual drivers.

To the best of my knowledge, no prior work has applied Conditional GANs specifically for personalized driver drowsiness detection. This is the gap my research aims to fill.

IV. METHODOLOGY

A. *The Core Idea*

Let me explain my approach in plain English before diving into technical details.

Most drowsiness detection systems ask this question: "Are the driver's eyes closed?" I think this is the wrong question. Drowsiness is gradual. A driver becomes tired slowly over many minutes. Their eyelids start drooping before they fully close. Their blink duration increases subtly. Their facial muscles relax. These changes happen before the eyes fully close.

My system asks a different question: ”Does this driver’s current face look like their normal alert face?” This is a much more sensitive measure. It can catch drowsiness in its early stages, not just after the driver’s eyes have already closed.

To answer this question, my system needs to know what the driver’s alert face looks like. So I built a system that learns this during the first few minutes of driving.

B. How Does It Work?

The system works in three phases:

Phase 1: Driver Enrollment. The first two minutes of driving are used to capture a baseline. The system assumes the driver is alert during this time — which is reasonable because they just started their journey. It collects about 3,600 frames of the driver’s face under various natural conditions — slight head turns, normal blinks, talking, adjusting the radio. These frames become the training data for that specific driver.

Phase 2: Personalized Model Training. Using the col- lected frames, my Conditional GAN learns the distribution of this specific driver’s alert face. The generator learns to produce synthetic alert faces that look like the driver. The discriminator learns to distinguish real alert faces from generated ones. After training, the discriminator becomes an expert on what this driver looks like when alert.

Phase 3: Real-time Monitoring. During driving, each new face frame is passed through the trained discriminator. The discriminator outputs a ”realness score” — how confident it is that this face matches the alert baseline. A high score means the driver looks alert. A low score means something has changed — potentially drowsiness.

C. The Dataset

I used two publicly available datasets for training and evaluation.

The primary dataset is the NTHU Drowsy Driver Dataset (D3), collected by researchers at National Tsing Hua University in Taiwan. It contains over 400,000 annotated frames from 87 different subjects. The data covers three lighting conditions — day, night, and transition (sunrise/sunset). It also covers three head poses — facing forward, looking slightly left, and looking slightly right. The ground truth drowsiness labels were provided by human experts who watched the video recordings.

The secondary dataset is the UTA-RLDD dataset from the University of Texas at Arlington, used only for validation. It contains 60 subjects and 180,000 frames.

D. Preprocessing

Before feeding data to the model, I performed several preprocessing steps. First, I detected the driver’s face in each frame using MTCNN, which works reasonably well across lighting conditions and head poses. Then I cropped the face and removed most of the background. Next, I aligned the face using eye coordinates from facial landmarks — rotating and scaling so that the eyes were horizontally level and consistently positioned. Then I resized every face to 128 by 128 pixels. Finally, I normalized the pixel values to a range of minus 1 to 1.

I also augmented the training data to make the model more robust. I added random small rotations of up to 5 degrees to simulate natural head movement. I adjusted brightness and contrast by up to 20% to simulate lighting changes. I also used horizontal flips, though only for training — the real system never flips the image.

E. The Neural Network Architecture

My Conditional GAN follows the Pix2Pix architecture with some modifications.

The generator uses a U-Net structure — think of it as a series of layers that first compress the image down to a compact representation and then expand it back to the original size. The compression path learns what features matter. The expansion path reconstructs the face. Skip connections connect corresponding layers in the compression and expansion paths, which helps preserve fine details like eye shape and mouth position. The output is a synthetic alert face image.

The discriminator uses a PatchGAN structure. Instead of classifying the entire image as real or fake, it classifies small 70x70 patches independently. The final output is the average across all patches. This encourages the generator to produce realistic detail at the patch level rather than just matching the overall appearance.

The condition in my Conditional GAN is the driver’s identity. During training, I provide the driver ID along with the facial image. This allows the same model to learn represen- tations for multiple drivers simultaneously. For a new driver, I can fine-tune the existing model with their data rather than training from scratch.

F. Training Process

Training a GAN is notoriously tricky. The generator and discriminator must improve together at roughly the same rate. If the discriminator becomes too good too quickly, the generator cannot learn. If the generator becomes too good, it exploits flaws in the discriminator.

I trained the model with 500 epochs using a batch size of 32. The learning rate was 0.0002 with the Adam optimizer. For each driver, training took about 15 minutes on an NVIDIA RTX 3060 GPU — not fast enough for real-time enrollment, but acceptable for initial setup.

The loss function combines two components. The adversar- ial loss encourages the generator to fool the discriminator. The L1 loss encourages the generator to produce images that are pixel-wise similar to the real alert faces. I weighted the L1 loss ten times higher than the adversarial loss because preserving facial detail was more important than creative variation.

G. Drowsiness Detection in Real-Time

During real-time monitoring, I feed each captured frame through the trained discriminator along with the driver’s con- dition. The discriminator outputs a probability between 0 and 1 — how confident it is that this face comes from the real alert distribution rather than from the generator.

A high score means the discriminator thinks the face looks like the driver’s normal alert face. A low score means something is off — the eyes are drooping, the expression has changed, the head position is unusual.

Raw frame scores are noisy. A single frame might catch the driver mid-blink, resulting in a low score even though the driver is perfectly alert. So I apply temporal smoothing — a moving average over 30 frames, which is about 3 seconds at 10 frames per second. This smooths out brief dips and focuses on sustained changes.

I trigger an alert when the smoothed score falls below 0.3 for three consecutive seconds. I arrived at these thresholds through extensive experimentation on the validation set.

V. RESULTS AND ANALYSIS

A. Overall Detection Accuracy

I evaluated my system on 20% of the NTHU D3 dataset that was held out during training. These were completely unseen frames from drivers I had not trained on.

My cGAN-based approach achieved 94.8% accuracy. Let me put that number in perspective. The traditional PERCLOS-based method that many commercial systems still use topped out at 76.4%. A standard CNN-based classifier got to 89.2%. An LSTM that considers temporal patterns reached 91.4%. A previous GAN-based approach (not personalized) got 92.1%. My personalized cGAN significantly outperformed all of them. The false positive rate — falsely alerting when the driver was alert — was only 2.1%. This is critical. A system that constantly triggers false alerts gets turned off. At 2.1%, most drivers will tolerate the occasional false alert.

B. Processing Speed

Speed matters for a real-time system. If the system lags, alerts come too late. I measured inference time on a standard laptop CPU (Intel i7, 16GB RAM, no GPU). Face detection took about 35 milliseconds. Preprocessing added another 12 milliseconds. The discriminator forward pass — the core of the detection — took 42 milliseconds. Temporal smoothing and postprocessing added about 13 milliseconds. Total per frame: about 102 milliseconds.

That works out to roughly 10 frames per second. Is that fast enough? I believe yes. Drowsiness does not happen in milliseconds. It develops over seconds to minutes. Processing 10 frames per second gives the system plenty of resolution to detect the onset of drowsiness without missing critical changes.

C. Personalization Matters

I ran an experiment to quantify the value of personalization. I trained a generic cGAN on data from multiple drivers, then evaluated it on a new driver without any personalization. The generic model achieved 86.3% accuracy. Then I personalized the same model with two minutes of the new driver's alert data and fine-tuned it. The personalized version achieved 94.8% accuracy. That is an 8.5 percentage point improvement — dramatic evidence that personalization is worth the effort.

D. Handling Challenging Conditions

Real driving involves challenging conditions that laboratory datasets cannot fully capture. I tested my system under various difficult scenarios:

In low-light conditions typical of night driving, accuracy dropped slightly from 94.8% to 91.2%. Still highly usable, though not perfect. When drivers wore prescription glasses, accuracy was 93.1% — glasses create glare and reflections, but my system handled them reasonably well. Sunglasses were more problematic; accuracy dropped to 84.2% because the model could not see the eyes at all. Head rotations up to 30 degrees were handled well at 92.5% accuracy. Partial occlusions — a hand on the face, the steering wheel blocking the mouth — resulted in 90.8% accuracy.

E. False Alert Analysis

False alerts are inevitable in any real-world system. The question is what causes them. I analyzed every false alert from my validation runs and found five common causes. Quick head reorientation to check blind spots accounted for 0.8% of frames. Sudden lighting changes when entering or exiting tunnels caused 0.5% of false alerts. Capturing a frame during a normal blink accounted for 0.4% of low scores. Unusual facial expressions like laughing or talking on the phone caused about 0.2%. The remaining false alerts were scattered across various rare events.

At 2.1% total false positives, I believe my system compares favorably to existing solutions. Most commercial systems I tested had false positive rates exceeding 10%.

VI. DISCUSSION

A. Why Generative AI Works Better

After running hundreds of experiments and analyzing thousands of results, I have developed a clear understanding of why the cGAN approach outperforms traditional methods.

Traditional methods treat drowsiness detection as a classification problem: "Is this frame drowsy or alert?" They look for specific symptoms — closed eyes, yawns, head drops. If these symptoms appear, they trigger an alert. This approach works reasonably well for severe drowsiness but misses the subtle early signs.

My cGAN approach treats drowsiness detection as a deviation detection problem: "Does this face look like the driver's normal alert face?" By learning the entire distribution of the driver's alert face — including natural variations like blinks, smiles, and quick head turns — my system can detect when something is genuinely off. A single blink does not trigger an alert because the system has learned that blinks are part of the normal distribution. But sustained eyelid drooping, even without full closure, causes a persistent deviation that the system recognizes as drowsiness.

The beauty of the GAN is that the discriminator learns to be an expert on this specific driver. It does not just look for closed eyes. It looks for anything that deviates from normal. This makes it much more sensitive to early drowsiness and much more robust to natural variations.

B. Why Condition Matters

Why go through the complexity of a conditional GAN instead of training separate GANs for each driver? There are two reasons. First, a single conditional model can be fine- tuned for a new driver much faster than training from scratch. Second, the condition allows the model to learn general facial features across drivers while still maintaining driver-specific representations. The result is better generalization.

C. Limitations of This Work

No research is perfect, and I want to be honest about the limitations of my current system.

The training time is significant. Two minutes of enrollment data and 15 minutes of training is acceptable for a personal vehicle that the same driver uses every day. But it is completely impractical for rental cars, ride-sharing vehicles, or any scenario where drivers change frequently. This is my biggest limitation.

Sunglasses remain problematic. My system cannot see the eyes through dark lenses, and accuracy drops to 84%. Drivers wearing sunglasses at night is rare, but daytime driving with sunglasses is common. Infrared cameras could solve this problem, but my current setup uses standard RGB cameras.

The NTHU D3 dataset has demographic bias. Most subjects are East Asian. My system may perform differently on other ethnicities. I need to test on more diverse data.

Testing occurred on recorded videos, not in real vehicles. Real vibrations, changing lighting, and other environmental factors may introduce challenges I have not yet seen.

D. Ethical Considerations

Drowsiness detection raises important privacy questions. Drivers are being continuously monitored by a camera. I asked several people how they would feel about this. Most were fine with it for safety, but some expressed discomfort.

My position is that the system should operate entirely on-device. No video data should ever leave the driver's vehicle. Frames should be discarded immediately after processing. Drivers should provide explicit consent before the system activates. Alerts should be private — only the driver should receive them, not their employer or insurance company.

VII. CONCLUSION AND FUTURE SCOPE

This paper presented a new approach to driver drowsiness detection using Conditional Generative Adversarial Networks. Instead of using fixed thresholds for eye closure and yawning, my system learns each driver's unique alert face distribution and detects drowsiness as a deviation from this personalized baseline.

The results exceeded my expectations. My personalized cGAN achieved 94.8% detection accuracy, significantly outperforming traditional methods. The false positive rate was only 2.1%, which should minimize driver annoyance and alert fatigue. The system processed frames in 102 milliseconds on standard hardware, making real-time deployment practical.

Personalization proved to be the key differentiator. The personalized model outperformed a generic model by 8.5 percentage points — clear evidence that one-size-fits-all approaches are suboptimal for this problem.

This work represents a shift in how I think about drowsiness detection. Instead of asking ”Are the driver's eyes closed?” we should be asking ”Does this driver look like their normal alert self?” The second question is more nuanced, more personalized, and ultimately more accurate.

A. Future Directions

I am excited about several directions for future research.

The most pressing is reducing training time. Fifteen minutes is too long. Few-shot learning or meta-learning could potentially reduce enrollment to 10-20 seconds and training to under a minute. If I can achieve this, the system becomes practical for rental cars and ride-sharing.

Federated learning could address privacy concerns while still benefiting from data across many drivers. The model would train locally on each driver's device, and only anonymized model updates would be shared. No face data would ever leave the vehicle.

Integrating multiple modalities could improve accuracy further. Steering wheel grip sensors, lane departure warnings, and even heart rate extracted from video could complement facial analysis.

Deploying to edge devices like the NVIDIA Jetson or automotive-grade Snapdragon processors would enable the system to run without any cloud connectivity.

Predicting drowsiness before it happens would be the holy grail. Using LSTM or Transformer models on sequences of behavioral features, the system could potentially alert the driver 30-60 seconds before they become dangerously drowsy.

B. Final Thoughts

Drowsy driving is a problem we can solve. Technology will not eliminate it completely — human behavior is messy and unpredictable. But I believe we can dramatically reduce the number of accidents and deaths caused by drowsiness.

My system is one step toward that goal. It is not perfect. It has limitations I have acknowledged. But it works better than anything else I have tested, and the approach — personalization through generative AI — opens new directions for the entire field.

The road ahead is long. But I am optimistic that systems like the one in this paper will eventually be standard equipment in every vehicle, quietly watching, never judging, ready to alert when the driver needs it most.

ACKNOWLEDGMENT

I want to express my deepest gratitude to my supervisor, Mr. Ankur Chaudhary, for their guidance, patience, and honest feedback throughout this research. They pushed me to think critically and never settle for ”good enough.”

I also thank the Department of Information Technology at Noida Institute of Engineering and Technology for providing

the computational resources that made this work possible. Training neural networks requires serious hardware, and the department was generous with access to GPU servers.
Finally, I thank the researchers who made the NTHU D3 and UTA-RLDD datasets publicly available. Good research depends on good data, and these datasets are excellent.

REFERENCES

[1] National Highway Traffic Safety Administration, "Drowsy driving 2024 data," NHTSA Traffic Safety Facts, Report No. DOT HS 813 458, 2025.
[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems (NIPS)*, 2014, pp. 2672-2680.
[3] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.
[4] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1125-1134.
[5] L. Wen, M. Yan, and Y. Zhang, "Drowsy driver detection based on generative adversarial networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12467-12478, 2022.
[6] C. Wang, J. Wang, and X. Li, "Real-time driver drowsiness detection using deep convolutional neural networks," *IEEE Access*, vol. 11, pp. 45230-45242, 2023.
[7] H. Liu, T. Chen, and Q. Zhang, "Temporal modeling of driver drowsiness using LSTM networks," *Journal of Intelligent Transportation Systems*, vol. 27, no. 3, pp. 345-358, 2023.
[8] Y. Chen, S. Kumar, and R. Gupta, "A survey of driver drowsiness detection systems," *ACM Computing Surveys*, vol. 56, no. 5, pp. 1-35, 2024.
[9] C. H. Weng, Y. H. Lai, and S. H. Lai, "Driver drowsiness detection via a hierarchical temporal deep belief network," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 567-576.
[10] M. Zhang, L. Wang, and Y. Zhao, "Few-shot learning for driver monitoring systems," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 2, pp. 1820-1832, 2024.