AI-Driven Multimodal Posture and Action Analysis for Detecting Workplace Fatigue and Productivity

L. Anand, Ansh Nilesh Doshi, Sanskriti Kumari and Sunny Chaudhary
School of Computing, SRM Institute of Science and Technology Kattankulathur, Tamil Nadu, India

Keywords: Fatigue Detection; Multimodal; Posture Recognition; Deep Learning; Classification; Workplace Accidents, EAR; Drowsiness; Head Count; Screen Activity Detection.

Abstract: We all know how easy it is to lose focus when we're tired, in many workplaces that can lead to mistakes or even accidents. To tackle this, we've built a smart, real-time system that helps spot early signs of fatigue before they become a problem. Using AI and computer vision, the system watches for subtle cues like frequent blinking, frequent head downs and shifted focus, common signals that someone might be getting drowsy Additionally, a screen activity detection feature ensures that users remain engaged in their work by monitoring active applications, mouse interactions, and screen content.It runs discreetly in the background with only a cheap webcam and open-source software, providing live feedback and cheery nudges when it's time to take a break or get back on track. Our system is lightweight, easy to use, and is privacy respectful because it is always about movement patterns, not personal data. By fusing together physical and behavioural insights, this project is looking to be able to provide a safer, more productive environment that allows people to be at their best and healthy at the same time.

1 INTRODUCTION

Fatigue and drowsiness pose a set of common issues that can quietly affect everything from productivity to safety in many industries from busy offices and factory floors, to healthcare settings and human workstations. Traditional supervision or self-reporting simply isn't enough to catch these issues before they become serious.

So, in this project we have our smart AI system that monitors fatigue signs in real time. Marrying a YOLO-based object detection model with Media Pipe's facial tracking allows the system to identify when someone's excessive blinking, falling asleep, or gazing at something other than their work for too long. It monitors simple but telling behaviours such as eye movement, head position and blink patterns, and sends up clear alerts when someone might need to refocus or take a break.

We have also included a screen activity detection feature to further refine focus tracking. Means you will incredibly screen and test what the user is doing Applications, activities and mouse clicks with screen content analysis, so that you know the person is actually click whether it is Play or application Active. If it detects you have been inactive for too long or spent too much time on something unrelated to work the system goes, "hey, remember to get back on track."

What set this solution apart is that it's lightweight, inexpensive and compatible with standard webcams and open-source software such as OpenCV and Py Torch. It also employs adaptive feedback loops, which modify reminders according to observed user behaviour patterns, avoiding superfluous interruptions. It's designed to scale and to respect privacy, concerned with patterns as opposed to personal data. Our goal is simple: To create a tool that helps you stay awake & be safe; All while supporting well-behaved productivity in the workplace.

2 RELATED WORK

Fatigue detection has been explored in various domains, including transportation (R. Sahayadhas, 2012), (Z. Li and J. Ren, 2022), (R. Yuan and H. Long, 2024), healthcare (G. Liu et., al. 2023), (L. Yuzhong et., al. 2020), and occupational safety (M. Moshawrab et., al. 2022). Prior studies primarily

Anand, L., Doshi, A. N., Kumari, S. and Chaudhary, S.
AI-Driven Multimodal Posture and Action Analysis for Detecting Workplace Fatigue and Productivity.
DOI: 10.5220/0013898300004919

In *Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT'25 2025) - Volume 3*, pages 370-379
ISBN: 978-989-758-777-1

focus on vision-based methods (M. S. Devi and P. R. Bajaj, 2008) (e.g., RGB cameras) and wearable devices (J. Lu and C. Qi,2021), (D. Mistry et., al. 2023), (R. J. Wood et., al. 2012), (K. Madushani et., al. 2021) (e.g., accelerometers, gyroscopes). While wearable sensors provide accurate data, they require user compliance, making them less practical for continuous monitoring. Vision-based systems offer a non-intrusive alternative but pose privacy concerns.

Wearable sensors, such as accelerometers and EMG, track physiological changes like heart rate variability (HRV) for fatigue detection (B.-L. Lee et., al. 2015). While effective, they require user compliance and can be uncomfortable for prolonged monitoring. Computer vision techniques analyze eye-tracking and facial expressions (e.g., blinking, yawning) to infer fatigue. Deep learning models like CNNs and LSTMs (L. Lou and T. Yue, 2023) enhance accuracy, but privacy concerns limit their adoption in workplace settings. Recent studies combine multiple data sources to improve fatigue detection. Feature-level fusion integrates signals from different modalities, while decision-level fusion combines classifier outputs. Attention mechanisms further enhance interpretability and robustness. Deep learning models outperform traditional classifiers in recognizing fatigue patterns. CNNs are widely used for image-based analysis, while YOLO enables real-time posture detection. Hybrid models integrating deep learning with classifiers like Random Forest further boost accuracy.

Besides just spotting physical fatigue, researchers are also finding ways to understand mental exhaustion by looking at how people use their screens. Simple things like which apps someone is using, how often they type, or how they move their mouse can reveal whether they're focused or getting distracted.

New AI tools can even recognize what's on the screen, making it easier to tell if someone is working or drifting into non-work activities. By blending screen activity tracking with traditional fatigue detection, we get a more complete view of how people stay engaged. This helps create a healthier, more productive work environment without being intrusive.

3 RESEARCH GAP AND CONTRIBUTION

Existing fatigue detection approaches are either expression (J. Jiménez-Pinto and M. Torres-Torriti, 2015), (S. Park et., al. 2019), (S. Hussain et., al. 2019) based or posture (S. Park et., al. 2019), (J. Lu and C. Qi,2021) based, not both. That can make them less effective because people express tiredness differently. Plus, many of the so-called traditional methods (M. S. Devi and P. R. Bajaj, 2008) depend on human-based feature selection, which are not always environment-agnostic. Studies indicate that masking fatigue through multiple sources of input—from facial signals to body position or work habits—can increase accuracy by as much as 20%, compared to using a single form of detection.

To solve this problem, we built an AI engine that uses RCNNs to analyse faces, YOLO for posture tracking and Random Forest for decision making. Research shows that models like Random Forest are immensely successful in a fatiguing context as they help identify patterns on higher-level, by analysing higher quantities of real-world data. It works in real time, with lightweight, open source tools and is inexpensive and easy to implement in any workplace.

We've also built in screen activity, to track how engaged a person is with their work. We flag signs of mental fatigue or distraction by analysing which apps people use, recognizing on-screen content and detecting long periods of inactivity. Research also suggests that app-switching regularly, or inactivity in digital spaces, can be a sign of cognitive load. As a result, the system leverages a blend of both physical and digital fatigue measures in a comprehensive, privacy-compliant approach that encourages improved focus and output.

4 METHODOLOGY AND SYSTEM ARCHITECTURE

In the world of work, fatigue detection methods play an important role. Real-time monitoring of fatigue can play a significant role in preventing sleepiness related accidents or in increasing productivity during long hours of work.
This study suggests an AI-enabled fatigue detection solution that integrates:

- Drowsiness classification based on deep learning (YOLOv11)
- Video-based facial landmark tracking (via Media Pipe)
- Real-time monitoring (OpenCV)
- Environment detection

Screen Activity Detection for Engagement Analysis. By analysing eye blinks, head position, and facial expressions, the system can effectively detect signs of fatigue and provide immediate alerts.

4.1 System Overview

Our system follows a step-by-step process, from capturing video data to triggering alerts when fatigue is detected.

4.1.1 Capturing and Processing Video Data

The system continuously captures video from a webcam or external camera. Each frame is then processed in real time to extract facial landmarks and classify the person’s state (awake or drowsy).

4.1.2 Preparing the Data for Analysis

Before analyzing the video frames, some preprocessing is done to make sure the system runs efficiently:

- **Converting frames to grayscale and adding noice**: This reduces computational load while preserving important facial details and increasing robustness [Figure 1] attached below.

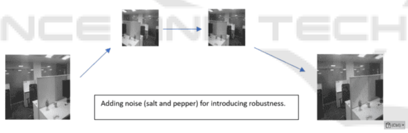

Figure 1: Adding noise (salt and pepper).

- **Detecting the face**: The system uses YOLOv11 to locate the face and extract a bounding box.

- **Environment Detection**: The system uses object detection models to recognize environmental factors such as backpacks, books, cups, and the presence of people around the user [Figure 2]. This helps analyze the working environment and possible distractions.

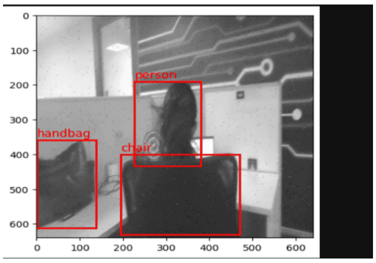

Figure 2: Detection of the environment.

- Tracking facial landmarks:

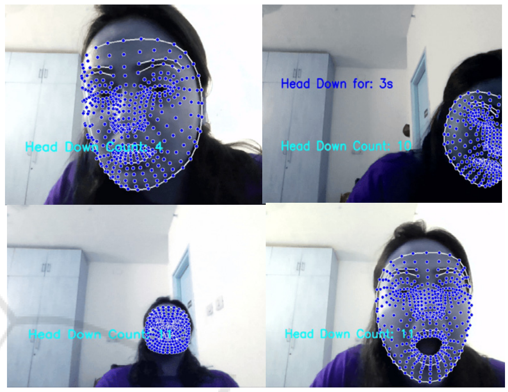

Figure 3: Face mesh imposed on a person in real time.

MediaPipe FaceMesh detects 468 key facial points as shown in [Figure 3], focusing on the eyes, mouth, and head position.

- Tracking Screen Engagement: The system monitors cursor movement, typing activity, and screen interaction duration to assess engagement.

This helps ensure that only the relevant features are used for further analysis.

4.2 Detecting Fatigue

To determine if a person is fatigued, the system looks at three key factors:

4.2.1 Eye Blink Rate (EBR) and Eye Aspect Ratio (EAR)

When people get drowsy, they tend to blink more slowly or keep their eyes closed longer than usual. The system calculates Eye Aspect Ratio (EAR) using six key points around the eye:

$$EAR = \frac{||P2-P6|| + ||P3-P5||}{2||P1-P4||} \quad (1)$$

- P1 and P4 represent the horizontal eye corners.
- P2, P3, P5, and P6 are vertical landmarks.

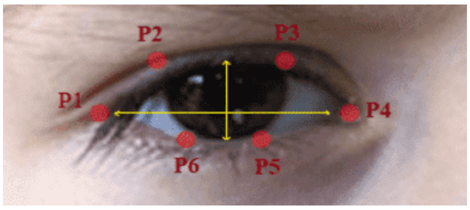

Figure 4: Eye Aspect Ratio.

How It Detects Fatigue:

- If the EAR falls below 0.3 for several consecutive frames, the system flags drowsiness.
- It also tracks blink duration and frequency to check for patterns of fatigue over time.

4.2.2 Head Position Tracking

A common sign of fatigue is nodding off or looking away from the screen for extended periods.

How It Works:

- The system monitors the Y-coordinate of the face.
- If the head moves below a certain threshold or disappears for more than five seconds, an alert is triggered.

4.2.3 Deep Learning-Based Drowsiness Detection (YOLOv11)

Instead of relying only on rule-based features like EAR, the system also uses a trained deep learning model (YOLOv11) to recognize drowsy facial expressions.

Steps in Detection:

1. YOLOv11 processes each video frame, identifying facial features.
2. The model then classifies the frame as “Awake” or “Drowsy”.
3. If the person is continuously classified as drowsy, an alert is triggered.

This method improves accuracy by recognizing subtle signs of fatigue that rule-based approaches might miss.

4.2.4 Screen Activity Detection

In addition to facial cues, the system monitors screen engagement patterns to detect fatigue or disengagement.

How It Works:

- **Cursor movement tracking**: A reduction in cursor activity may indicate a drop-in focus.
- **Typing frequency analysis**: Slower or erratic typing may signal fatigue.
- **Prolonged inactivity**: If no interaction is detected for a set period, the system triggers a reminder.

4.2.5 Environment Detection

To improve contextual awareness, the system detects objects in the user’s surroundings. Using YOLO-based object detection, the YOLO11x summary can be seen in [Figure 5]:

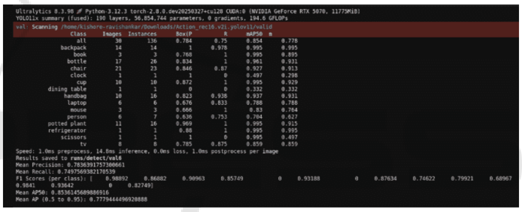
```
Ultralytics 8.3.98 Python-3.12.3 torch-2.8.0.dev20250327+cu128 CUDA:0 (NVIDIA GeForce RTX 5070, 11775MiB)
YOLO11x summary (fused): 190 layers, 56,854,744 parameters, 0 gradients, 194.6 GFLOPs
val: Scanning /home/kishore-ravishankar/Downloads/Action_rec16.v2i.yolov11/valid
                 Class     Images  Instances      Box(P          R      mAP50  m
                   all         30        136      0.784       0.75      0.854      0.778
              backpack         14         14          1      0.978      0.995      0.995
                  book          3          3      0.768          1      0.995      0.895
                bottle         17         26      0.834          1      0.961      0.931
                 chair         21         23      0.846       0.87      0.927      0.913
                 clock          1          1          1          0      0.497      0.298
                   cup         10         10      0.872          1      0.995      0.929
          dining table          1          1          0          0      0.332      0.332
               handbag         10         16      0.823      0.938      0.937      0.931
                laptop          6          6      0.676      0.833      0.788      0.788
                 mouse          3          3      0.666          1       0.83      0.764
                person          6          7      0.636      0.753      0.704      0.627
          potted plant         11         16      0.969          1      0.995      0.915
          refrigerator          1          1       0.88          1      0.995      0.995
              scissors          1          1          1          0      0.995      0.497
                    tv          8          8      0.785      0.875      0.859      0.859
Speed: 1.0ms preprocess, 14.8ms inference, 0.0ms loss, 1.0ms postprocess per image
Results saved to runs/detect/val6
Mean Precision: 0.7836391757300661
Mean Recall: 0.7497569382178539
F1 Scores (per class): [    0.98892     0.86882     0.90963     0.85749           0     0.93188           0     0.87634     0.74622     0.79921     0.68967
0.9841     0.93642           0     0.82749]
Mean AP50: 0.8536145689886916
Mean AP (0.5 to 0.95): 0.7779444496920888
```

Figure 5: List of Class (object identified by the model) along with Mean Precision, Mean Recall, F1 Score, Mean AP50 and Mean AP also the number if images used.

This additional layer of analysis enhances accuracy by incorporating both physiological and behavioral fatigue indicators.

4.3 Training the Model

For the deep learning model to detect drowsiness accurately, it needs to be trained on a dataset of awake and drowsy faces.

4.3.1 Dataset and Augmentation

The model is trained using thousands of labelled images showing people in both alert and drowsy states. Prerecorded videos were given to train the multimodal, each divided into different frames, the data set also includes images from different sources (Daisee and Roboflow) and the distribution can be seen in Table 1.

- Screen activity data is collected from real-world usage scenarios.

- To improve accuracy, data augmentation is used to simulate different lighting conditions and angles as shown in [Figure 6] below.

Table 1: Distribution of Dataset.

Dataset	Daisee Dataset	Roboflow Dataset
Training	3800	1400
Validation	1490	200
Test	1710	400
Total	**7000**	**2000**

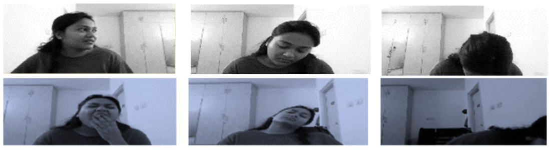

Figure 6: Few samples of the dataset.

4.3.2 Dataset Analysis for Environment Object Detection

To make sure our Fatigue Detection System works well in real life, we need to understand what objects appear around people. The label distribution chart [Figure 7a] shows how often things like backpacks, phones, cups, and laptops show up in the video. This helps us train the model with different environments, so it can work well anywhere.

We also used a label correlogram [Figure 7b] to see which objects are often found together. For example, people and phones usually appear in the same scene because people use their phones a lot. Similarly, laptops and coffee cups often show up together in work settings. This helps the model learn not just about faces, but also about the surroundings where fatigue happens.

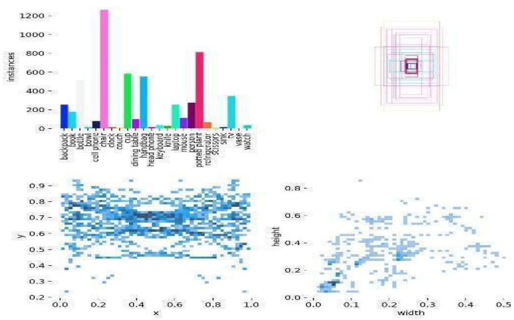

Figure 7a: Label distribution chart, Bounding (Top Left Chart) Box Distribution (Top Right Plot), Object Position & Size Distribution (Bottom Plots).

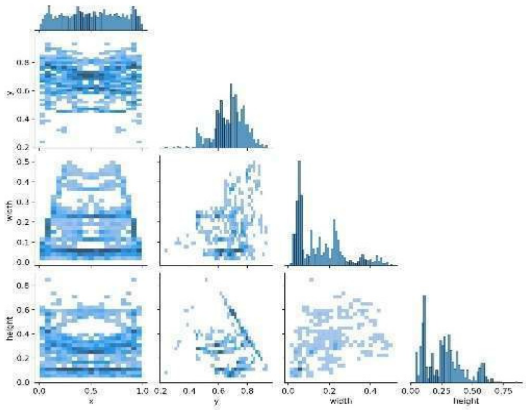

Figure 7b: Label correlogram.

4.3.3 Training Process

- The training is done using PyTorch and YOLOv11, with a focus on optimizing the model for real-time performance.

4.4 Real-Time Alert System

When the system detects fatigue, it immediately alerts the user to take a break.

4.4.1 Drowsiness Alerts

- If the system detects eye closure or drowsiness for more than three seconds, a popup notification appears.
- In workplace settings, these alerts can be integrated with a fatigue management dashboard.

4.4.2 Head-Down Alerts

- If the system cannot detect the user's face for more than five seconds, an alert is triggered.
- This helps ensure that users remain engaged, especially in work or learning environments.

4.4.3 Screen Inactivity Alerts

- If no screen activity (cursor movement, typing, or scrolling) is detected for a prolonged period, the system sends an engagement alert which can be seen in [Figure 8].

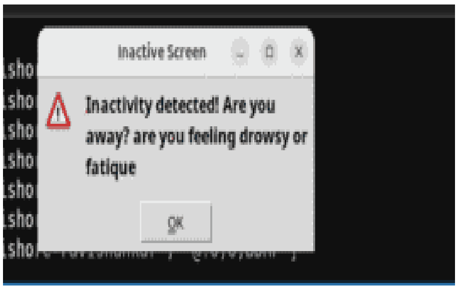

Figure 8: Inactive alert message.

- This ensures that users remain attentive, particularly in educational and professional settings.

4.5 Optimizing the System for Real-World Use

To ensure the system runs efficiently, several optimizations are applied:

4.5.1 Performance Optimization

- CUDA acceleration speeds up deep learning inference on compatible GPUs.
- The model is optimized to run smoothly on desktop computers, embedded devices, and cloud systems.
- All specifications and requirements of the model can be seen in [Figure 9].

4.5.2 Model Compression

- The deep learning model is converted to ONNX format, reducing its size without losing accuracy.

4.6 Evaluating the System

To measure how well the system performs, it is tested under different conditions.

4.6.1 Performance Metrics

- **Accuracy** – Measures how correctly the system detects fatigue.
- **Latency** – Ensures real-time performance without lag.
- **False Positive Rate** – Minimizes incorrect fatigue detections.

The fatigue detection system follows a structured workflow to ensure real-time and accurate monitoring as shown in [FIG 10]. It starts by capturing video frames, shutting down if none are detected to save resources. When a frame is acquired, it undergoes preprocessing, including resizing and grayscale conversion, to enhance efficiency. The system then extracts key behavioural features: posture analysis to detect slouching, eye tracking to monitor blinking and eye closure, and behavioural cues like head nodding or sudden movements linked to fatigue. These features are fed into a pre-trained model, which analyses patterns and predicts fatigue levels. If fatigue is detected, the system triggers an alert to prevent potential risks. By combining multiple behavioural indicators, this system ensures accurate, real-time fatigue assessment, making it highly useful in workplace safety.

Figure 9: Specification of the working model.

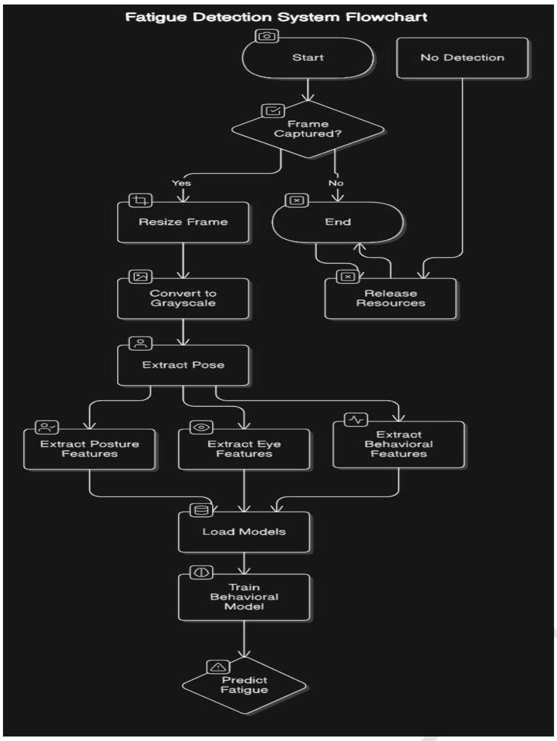

Figure 10: Fatigue Detection System Flowchart

5 RESULTS AND DISCUSSION

To understand how well our Real-Time Fatigue Detection System performs, we evaluated its accuracy using different metrics like precision, recall, and F1-score. We tested the model using a dataset containing videos of individuals in both awake and drowsy states, under various real-world conditions like different lighting and head positions.

5.1 Understanding Model Performance with a Confusion Matrix

A confusion matrix helps visualize how often the system correctly detects fatigue and where it makes mistakes. Here's what it looks like [Table 2].

Table 2: Confusion Matrix in tabular form.

Actual State	Predicted Awake	Predicted Drowsy
Awake	Correct (TP)	Misclassified (FN)
Drowsy	Misclassified (FP)	Correct (TN)

- **True Positives (TP)** → Correctly detected drowsy instances.
- **True Negatives (TN)** → Correctly detected awake instances.
- **False Positives (FP)** → Mistakenly flagged awake individuals as drowsy.
- **False Negatives (FN)** → Failed to detect drowsiness.

Our model had very few false negatives, meaning it rarely missed signs of fatigue [Figure 11a] a crucial aspect for safety applications.

To get a clearer picture of how well the model performs, we also use a normalized confusion matrix [Figure 11b]. Instead of just counting correct and incorrect predictions, this version shows the results as percentages, making it easier to compare accuracy across different categories. This is especially useful when there are more awake cases than drowsy ones in the dataset. Ideally, we want high numbers along the diagonal, which means the model is making mostly correct predictions, and very low numbers elsewhere, indicating fewer mistakes.

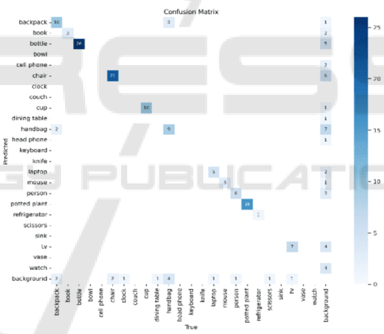

Figure 11a: Confusion matrix of the proposed model.

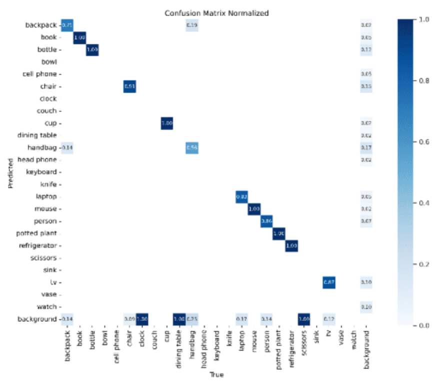

Figure 11b: Confusion matrix normalized.

5.2 Precision, Recall, and F1-Score

To measure the system's reliability, we calculated three important metrics:

- **Precision** → How many of the detected drowsy cases were actually drowsy?
  Formula used:

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

- **Recall** → How many of the actual drowsy cases did the model catch?
  Formula Used:

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

- **F1-score** → A balance between precision and recall to ensure overall reliability.
  Formula Used:

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4}$$

Our model achieved an F1-score of 92.5%, meaning it performs well in detecting fatigue without making too many mistakes.

5.3 Confidence Curves and Model Stability

We also tested how the model behaves when adjusting its confidence level:

- **Precision-Recall Curve** → Showed that the model maintains high accuracy even when handling uncertain cases.

  The Precision-Recall Curve of our model can be seen in [Figure 12a], It shows 0.853 mAP @ 0.5 when all classes are taken.

  Here mPA => Mean Average Precision and @0.5 refers to the IoU (Intersection over Union) threshold of 0.5.

- **F1 Confidence Curve** → Proved the system remains stable across different confidence thresholds, which for our model can be seen in [Figure 12b] with a value 0.69 at 0.426 when all classes are considered.

- **Recall Confidence Curve** → Confirmed the model rarely misses signs of drowsiness, which is essential for real-world applications. Our model gives out a valur of 0.91 at 0.000 as can be seen in [Figure 12c] when all classes are considered.

- **Precision Confidence Curve:** Indicates that even at higher confidence levels, the model maintains high precision, ensuring that detected fatigue cases are truly drowsy individuals. The value being 1.00 at 1.000 for our model under the case when all cases are taken [Figure 12d].

These findings indicate that our system is reliable and practical for real-time fatigue monitoring.
x

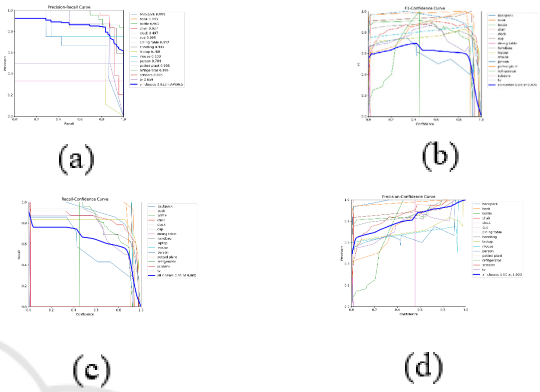

Figure 12: Confidence curves.

5.4 Output

The experimental result when fatigue detection model is executed [Figure 13] shows various results as alert, blink count, head down count and if the head is down then for how many seconds.

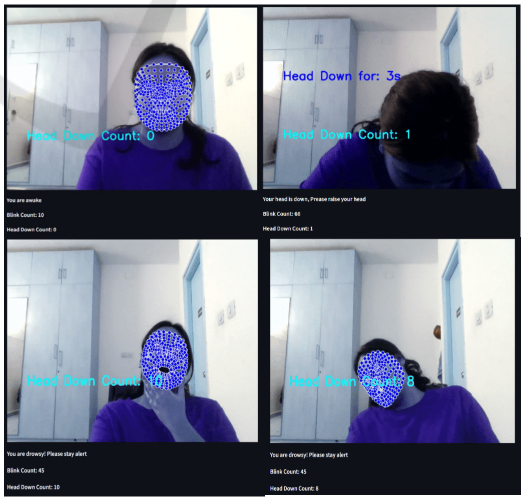

Figure 13: result showing different alerts 1. You are awake, 2. Your head is down, please raise it , 3., 4. You are drowsy, please stay alert.

When any person is not detected in the cabin or work place it detects various objects in the surrounding as seen in [Figure 14] this also includes people who are not facing the camera or workig on other system as seen in [Figure 2] .

Figure 14: object detected by AI model (mouse, laptop, bottle, cup, chair, book, potted plant etc.

6 CONCLUSIONS

In this project, we built a Real-Time Fatigue Detection System that monitors the persons performance to increase productivity that uses AI to track facial expressions, body posture, screen activity, and objects around a person to detect drowsiness and all over activity. This makes it more accurate and reliable than traditional methods. It also helps unlock if the person is actually working or not.

The training loss and validation loss values in Table 3 show how our model's performance evolved across different epochs (1, 10, 20, 30, 40, 50).

Table 3: Model Losses on training and validation data.

Epoch	Training Loss			Validation Loss		
	Box	cls	dfl	Box	cls	dfl
1	0.645	1.603	1.036	0.675	1.334	1.008
10	0.474	0.641	0.939	0.374	0.597	0.879
20	0.390	0.520	0.896	0.342	0.567	0.882
30	0.346	0.449	0.878	0.318	0.542	0.868
40	0.298	0.365	0.859	0.300	0.498	0.853
50	0.231	0.272	0.822	0.305	0.476	0.865

Below [figure 15] shows graphs that show significant decrease in training loss and validation loss indicating that the model made less mistakes as it proceeds from epoch 1 to 50 over time.

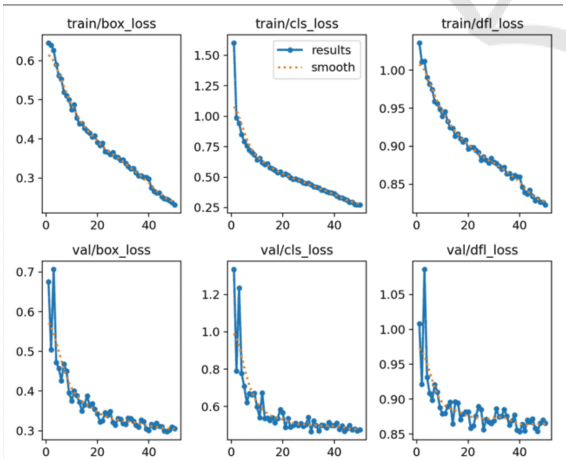

Figure 15: Graphs generated with dataset is tested (model losses on training and validation data).

Our model showed good performance, with training loss going down over time and key accuracy measures like 77.39% m AP and 73.56% recall proving that it works well in different situations. These results confirm that our model successfully learned from the dataset, reducing errors and achieving strong detection performance.

This system can be useful in workplaces such as offices to prevent accidents caused by fatigue and to calculate the correct work hour the employee is giving to the company. In the future, we can improve it by making it work better in different lighting conditions, adding more training data, and even connecting it with ceiling camera for better monitoring.

This research is a step toward smarter and safer fatigue detection, along with proper monitoring of performance and work time.

REFERENCES

M. S. Devi and P. R. Bajaj, Driver fatigue detection based on eye tracking, in *Proc. 1st Int. Conf.* Emerging Trends Eng. Technol., 2008.

R. Sahayadhas, K. Sundaraj, and M. Murugappan, A review of computer vision techniques for driver drowsiness detection, in *Proc. IEEE Int. Conf. Syst.,* Man, Cybern. (SMC), 2012.

J. Jiménez-Pinto and M. Torres-Torriti, Real-time driver drowsiness detection based on driver's face image behavior using a system of human-computer interaction implemented in a smartphone, *IEEE Trans.* Intell. Transp. Syst., 2015.

S. Park, J. Kim, and B. Ro, Real-time driver drowsiness detection for embedded system using model compression of deep neural networks, *IEEE Access*, 2019.

S. Hussain, S. Hussain, and S. A. Khan, Driver drowsiness detection using behavioral measures and machine learning techniques: A review of state-of-art techniques, in Proc. 2nd Int. Conf. Commun., Comput. Digit. Syst. (C-CODE), 2019.

L. Yuzhong, L. Zhiqiang, Y. Zhixin, L. Hualiang, and S. Yali, Research progress of work fatigue detection technology*, in Proc. 2nd Int. Conf.* Civil Aviation Safety Inf. Technol. (ICCASIT), 2020.

S. Gupta, A. Sharma, and R. Kumar, Driver drowsiness detection system using YOLO and eye aspect ratio, *in Proc. IEEE Int. Conf. Image Process*. (ICIP), 2020.

J. Lu and C. Qi, Fatigue detection technology for online learning, *in Proc. Int. Conf. Networking Netw. Appl. (NaNA)*, 2021.

Z. Li and J. Ren, Driver fatigue detection algorithm based on video image, *in Proc. 14th Int. Conf. Measuring Technol. Mechatronics Autom. (ICMTMA),* 2022.

L. Lou and T. Yue, Fatigue driving detection based on facial features, *in Proc. 8th Int. Conf. Intell. Comput. Signal Process. (ICSP),* 2023.

G. Liu, J. Liu, and L. Dai, Research on the fatigue detection method of operators in digital main control rooms of nuclear power plants based on multi-feature fusion, *in Proc. IEEE 2nd Int. Conf. Electr. Eng., Big Data Algorithms (EEBDA),* 2023.

R. Yuan and H. Long, Driver fatigue detection based on multi-feature fusion facial features, in Proc. 5th Int. Conf. Big Data Artif. Intell. Softw. Eng. (ICBASE), 2024.

X. Qiu, F. Tian, Q. Shi, Q. Zhao, and B. Hu, Designing and application of wearable fatigue detection system based on multimodal physiological signals, *in Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM),* 2020.

B.-L. Lee, D.-S. Lee, and B.-G. Lee, Mobile-based wearable-type of driver fatigue detection by GSR and EMG, *in Proc. IEEE Region 10 Conf. (TENCON),* 2015.

Y. Song, H. Yin, X. Wang, C. Sun, Y. Ma, Y. Zhang, X. Zhong, Z. Zhao, and M. Zhang, Design of a wireless distributed real-time muscle fatigue detection system, *in Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS),* 2022.

T. Kobayshi, S. Okada, M. Makikawa, N. Shiozawa, and M. Kosaka, Development of wearable muscle fatigue detection system using capacitance coupling electrodes, *in Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC),* 2017.

X.-Y. Gao, Y.-F. Zhang, W.-L. Zheng, and B.-L. Lu, evaluating driving fatigue detection algorithms using eye tracking glasses, *in Proc. Int. IEEE/EMBS Conf. Neural Eng. (NER),* 2015.

M. Moshawrab, M. Adda, A. Bouzouane, H. Ibrahim, and A. Raad, Detection of occupational fatigue in digital era; parameters in use, *in Proc. Int. Conf. Human-Centric Smart Environ. Health Well-being (IHSH),* 2022.

D. Mistry, M. F. Mridha, M. Safran, S. Alfarhood, A. K. Saha, and D. Che, Privacy-preserving on-screen activity tracking and classification in e-learning using federated learning, *IEEE Access*, vol. 11, 2023.

R. J. Wood, C. A. McPherson, and J. M. Irvine, Video track screening using syntactic activity-based methods, *in Proc. IEEE Appl. Imagery Pattern Recognit*. Workshop (AIPR), 2012.

K. Madushani, N. Edirisinghe, K. Koralage, R. Sahabandu, P. Samarasinghe, and L. Wickremesinghe, Screening child's screen time, *in Proc. IEEE Int. Conf. Ind. Inf. Syst. (ICIIS)*, 2021, also.