

# Sistema de Alerta de Sonolência para Motoristas com aprendizado de máquina em borda

Hyago V. L. B. Silva - 922; Felipe A. P. de Figueiredo; Samuel B. Mafra

*Instituto Nacional de Telecomunicações - INATEL*

Santa Rita do Sapucaí - MG

hyago.silva@mtel.inatel.br; felipe.figueiredo@inatel.br; samuelbmafra@inatel.br

**Abstract**—A sonolência e a má postura ao volante são fatores críticos que contribuem para a ocorrência de acidentes, principalmente em rodovias e no setor de transporte de cargas. Motoristas de caminhões frequentemente enfrentam longas jornadas e recorrem a substâncias estimulantes, como anfetaminas, em uma tentativa de manter-se acordados, o que aumenta os riscos à saúde e à segurança. Este artigo apresenta um sistema de baixo custo baseado em visão computacional, que utiliza bibliotecas de Machine Learning, como MediaPipe, dlib e YOLOv8, para monitorar em tempo real os sinais de fadiga e postura inadequada de motoristas, basicamente analisando sonolência pelos olhos, boca e postura. Este sistema propõe um comitê de 3 modelos, que estão concentrados em uma Raspberry Pi 5, utilizando uma câmera de 720p, conectada com um acelerador Edge TPU. O sistema alcança com YOLOv8 98,5% de acurácia, com 99,2% de mAP, para todas as duas classes. Os outros dois modelos do comitê, que são Dlib e MediaPIPE, auxiliam no comitê, para uma dupla verificação sobre inclinação para frente e olhos fechados durante um longo período. Por meio da análise de indicações faciais e posturais, o sistema detecta olhos fechados por períodos prolongados, sonolência como bocejar durante um longo período, e a postura, caso haja inclinação para frente ou para trás, e lados, emitindo alerta visuais e sonoros.

## I. INTRODUCTION

Os acidentes de trânsito estão entre as principais causas de mortalidade global, com a sonolência ao volante contribuindo para cerca de 20% a 30% dos incidentes fatais. No Brasil, motoristas profissionais enfrentam jornadas exaustivas, muitas vezes ultrapassando os limites regulamentares, levando ao uso de anfetaminas como solução paliativa para manter a atenção, apesar dos sérios riscos à saúde, como dependência e morte súbita [1], [2]. Esse cenário reflete não apenas um problema de saúde pública, mas também uma preocupação crescente em diferentes setores econômicos e sociais.

Estudos realizados no Brasil destacam a prevalência de acidentes envolvendo caminhões, frequentemente atribuídos à fadiga e à sonolência dos motoristas. Cerca de 30% dos acidentes rodoviários são causados por sonolência ao volante, como apontado pela Agência Nacional de Transportes Terrestres (ANTT) [3]. Dispositivos como câmeras de fadiga têm emergido como soluções tecnológicas promissoras, proporcionando monitoramento em tempo real do estado de alerta do motorista. Esses sistemas são capazes de detectar sinais de fadiga, como o fechamento prolongado dos olhos, e emitir alerta para evitar acidentes [4].

O impacto dessas tecnologias é evidenciado em estudos de caso que analisaram a implementação de câmeras de fadiga em

operações logísticas. Por exemplo, uma análise longitudinal de 18 meses mostrou uma redução de até 91% nas infrações relacionadas à fadiga, além de economia financeira significativa para empresas do setor [4]. Isso ressalta não apenas a eficácia dessas tecnologias, mas também a necessidade de promover mudanças culturais e operacionais que garantam segurança nas estradas e melhorem a qualidade de vida dos motoristas. Assim, o avanço na adoção de soluções tecnológicas reforça a importância de abordar de forma integrada os fatores de risco no transporte rodoviário de cargas no Brasil [4].

Embora veículos modernos já possuam tecnologias avançadas, como sensores infravermelhos para monitorar o estado do motorista, ou mesmo sistemas autônomos que podem assumir o controle do veículo, esses recursos são limitados a uma pequena parcela da população devido aos custos elevados [5]. A ausência de soluções acessíveis, não muito complexas ou custosas, para auxiliar os motoristas menos expostos a acidentes causados por sonolência e má postura são mais escassas [6].

Neste cenário, surge a oportunidade de se utilizar técnicas de visão computacional (CV) e aprendizado de máquina (ML), embarcado em sistemas simples, para criar soluções acessíveis e eficazes, capazes de detectar sinais de fadiga e má postura [7]. Este artigo propõe um sistema baseado em detecção facial e corporal, que utiliza modelos pré-treinados para monitorar os movimentos oculares e a posição corporal do motorista, além de avaliar a utilização de cinto de segurança, emitindo alerta em tempo real para prevenir acidentes.

O sistema proposto utiliza técnicas de (CV) para monitorar em tempo real o estado do motorista, detectando sinais de sonolência, má postura e possibilitando a identificação do uso de cinto de segurança. Baseado em bibliotecas como OpenCV, MediaPipe e dlib, YOLO (You-Only-Look-Once).

Algumas condições perigosas, como olhos fechados por longos períodos ou inclinação excessiva para frente, são detectadas, alerta visuais e sonoros são acionados. Com hardware acessível, como câmeras de 720p e dispositivo embarcado utilizando um Raspberry Pi, esta solução viabiliza a implementação de baixo custo, promovendo segurança no trânsito e auxiliando motoristas a evitar acidentes relacionados à fadiga, distração e negligência no trânsito.

## II. TRABALHOS RELACIONADOS.

Em [8], desenvolveu-se uma solução baseada em visão computacional para detectar estados de sonolência em motoristas. Utilizando o método Viola-Jones integrado à biblioteca OpenCV, o sistema classifica os estados do condutor (acordado, sonolento ou dormindo) com base no tempo de fechamento dos olhos, e emite alerta sonoros por meio de um buzzer. O dispositivo foi implementado em um Raspberry Pi 4 Model B, acoplado a uma câmera infravermelha para funcionar em condições de baixa luz. Com um custo acessível de R\$ 598,34, o sistema foi testado com cinco voluntários em diferentes condições de iluminação, obtendo uma acurácia geral de 96,83% em 727 amostras. O trabalho destacou-se por explorar componentes de baixo custo e tecnologias amplamente acessíveis, tornando a solução viável para aplicações práticas. Apesar disso, foram observados atrasos no processamento das imagens e a possibilidade de falsos positivos em situações como ressecamento ocular.

No artigo [9], apresenta-se um sistema híbrido que combina métodos clássicos de visão computacional com redes neurais convolucionais (CNNs) para identificar estados de sonolência e distração em motoristas. A abordagem utiliza o algoritmo Viola-Jones para detectar o rosto e os olhos, enquanto uma CNN, implementada em TensorFlow, classifica os olhos como abertos ou fechados. A solução foi testada utilizando o conjunto de dados UTA Real-Life Drowsiness Dataset, com 60 participantes, alcançando acurácia de 95,41% em um vídeo teste e 84,59% em outro. O sistema opera com pré-processamento das imagens, classificação do estado dos olhos e emissão de alerta baseados em um contador de quadros que rastreia períodos de sonolência. Entre os desafios, destaca-se a dependência do Viola-Jones, que apresentou falhas na detecção de olhos em alguns casos, e problemas causados por reflexos em óculos.

Em [10], se apresenta um sistema de detecção de sonolência em motoristas utilizando a arquitetura MobileNet-SSD, embarcado na Raspberry Pi 3. O modelo foi treinado com 4500 imagens anotadas, divididas em quatro classes: olhos abertos, fechados, bocejo, não bocejo. Foi utilizado transfer learning a partir de um modelo pré-treinado no MS COCO. Avaliado com a métrica PASCAL VOC, o sistema alcançou um mAP de 98,01%, destacando-se em condições de iluminação normal. O sistema apresentado no artigo enfrenta algumas limitações importantes. Primeiramente, sua performance é prejudicada em condições de baixa luminosidade ou em situações em que há reflexos de luz direta na lente da câmera. Isso se deve ao fato de o conjunto de dados de treinamento conter poucas imagens com essas características, limitando sua capacidade de generalização. O processamento do modelo demonstrou um limite de quadros, alcançando 7 a 8 FPS.

No artigo [11], demonstra-se uma solução para detectar fadiga facial em motoristas usando o modelo YOLOv8. Essa abordagem se destaca pela precisão e capacidade de processamento em tempo real, sendo aplicável em cenários de transporte inteligente para prevenir acidentes causados

por sonolência ao volante. O YOLOv8 incorpora avanços como o módulo C2f, que melhora a extração de recursos, e o uso de Generalized Focal Loss (GFL) e Distribution Focal Loss (DFL), que refinam a precisão das caixas de predição e classificação. O modelo também utiliza transfer learning para ajuste a tarefas específicas, reduzindo o tempo de treinamento e otimizando o desempenho. Apesar dos bons resultados, o modelo enfrenta limitações em ambientes de iluminação adversa e fundos complexos. Sugere-se que fusões multimodais, integrando dados fisiológicos e comportamentais, podem melhorar a precisão.

Este artigo reúne um conjunto de modelos para auxiliar na detecção em tempo real de sonolência ou indisposição em motoristas, utilizando uma análise de comitê com dois modelos. Estes modelos estão embarcados em uma Raspberry Pi 5 com uma câmera de 720p conectada ao dispositivo. O sistema fica analisando o motorista em dois casos, para três características, mas que podem ser combinadas. Esses casos são, primeiro, com os olhos fechados, boca aberta e inclinação do corpo demonstrando indisposição, e a segunda combinação, em estado de alerta, olhos abertos, boca fechada e inclinação correta do corpo. Caso seja detectado sonolência ou estado de indisposição devido a estas características acima, o sistema aciona um alarme, alertando o motorista.

A diferença do que é utilizado neste trabalho em relação à arquitetura do trabalho [9] e aos demais é de uma análise de comitê de dois modelos, de um modelo pré-treinado através das bibliotecas *dlib* e *MediaPIPE*, com detecção de sonolência através da inclinação do corpo e abertura ou fechamento dos olhos. Juntamente com YOLOv8, que analisa a boca e os olhos em relação à abertura e o fechamento de ambos. É utilizado um Raspberry PI 5 Model B, juntamente com uma câmera 720p, para detectar estas indicações.

Devido aos trabalhos possuírem problemas na detecção dos olhos e no processamento dos dados, foi utilizada uma verificação dupla nas detecções dos olhos fechados e abertos, e uma otimização no dispositivo, utilizando uma Edge TPU Coral aceleradora USB. Também foi necessariamente feita a quantização do modelo do YOLOv8, para o formato edgetpu, para poder ser processado com uma taxa de quadro maior, próxima do tempo real.

## III. METODOLOGIA

A arquitetura do sistema propõe um comitê de três modelos para análise de sonolência, indisposição ou detecção de fadiga nos motoristas. Foram utilizados dois modelos pré-treinados disponíveis pelas bibliotecas que o Python oferece.

### A. Arquitetura do sistema.

Estes modelos ou módulos instalados no Python processam os dados analisando a distância entre os pontos dos olhos, utilizando a *Dlib*. O módulo responsável por fazer a verificação da postura do motorista, se está inclinada para frente, para trás ou ereta, é o *MediaPIPE*, que faz a análise da distância entre a face e o meio dos ombros do motorista, ou seja, ponto central entre os ombros.

Este sistema analisa todas as três características, boca, olhos e postura. O comitê se organiza analisando duas vezes o caso dos olhos fechados e também a postura incorreta, contando frames e um score com *dlib* e *MediaPIPE*, juntamente com a detecção do Yolo. A postura é analisada através do MediaPIPE e o Yolo. O Yolo fica responsável pela análise da abertura da boca, do fechamento dos olhos, e pela postura também, caso se incline para trás, mas principalmente focado no rosto todo. O Yolo atua como modelo final de verificação do contexto total. Veja como o comitê se organiza para poder validar o resultado de forma mais assertiva com a figura 1, abaixo.

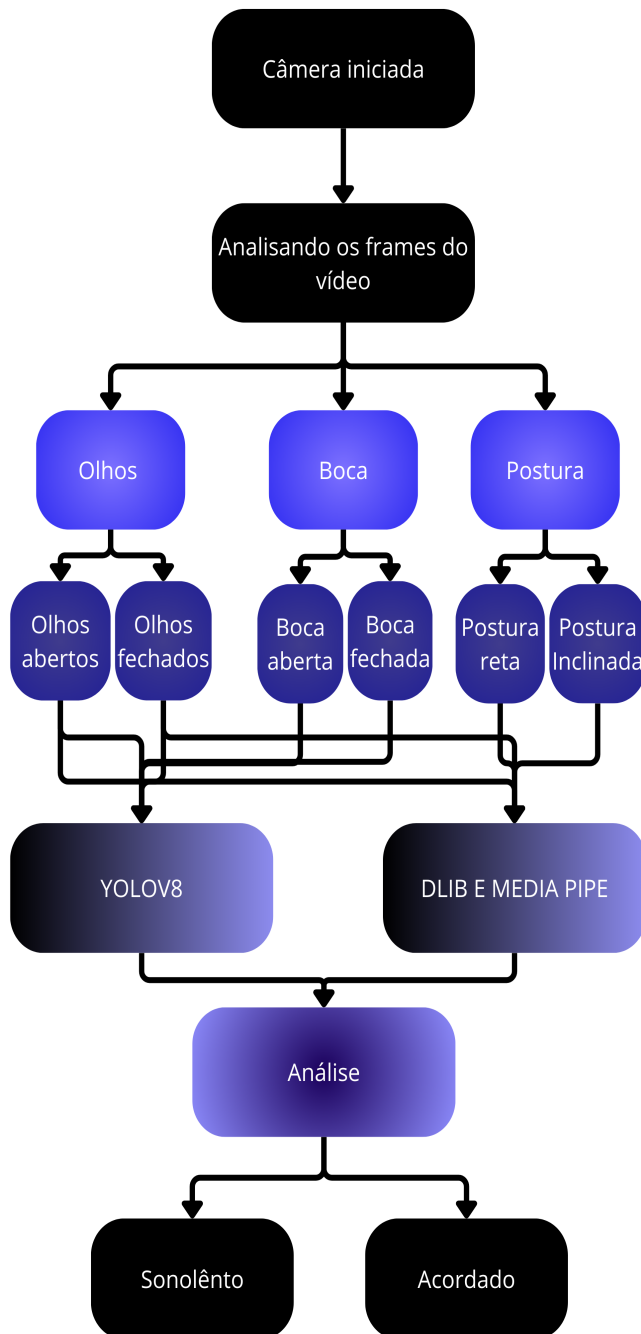


Fig. 1. Arquitetura do comitê de modelos

## B. *Dlib* e *Media PIPE*, Módulos nativos do Comitê.

Ambas as bibliotecas que trazem nativamente a vetorização do corpo e da face são rápidas e processam rapidamente os dados, o que auxilia na verificação mais rápida de sonolência. Veja a figura 2, que representa os pontos que o módulo *MediaPIPE* e o modelo pré-treinando com *Dlib* estão configurados para analisar nos frames para este sistema em específico. Onde, DSC (Distance-sholders-center) e BR (Blinking-Ratio) são métricas medidas respectivamente pelos módulos e modelos, *MediaPIPE* com DSC, e *Dlib* com BR.

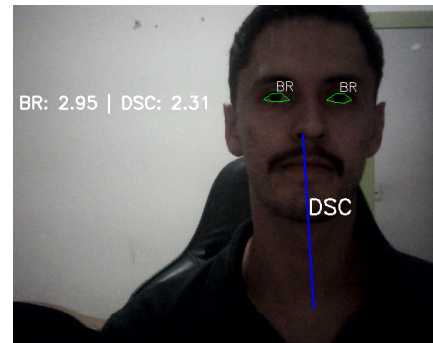


Fig. 2. Representação da implementação das bibliotecas *Dlib* e *MediaPIPE*, analisando respectivamente as métricas BR e DSC

A métrica BR é definida como a razão entre o comprimento horizontal e o comprimento vertical do olho, sendo utilizada para identificar se os olhos estão piscando lentamente ou permanecendo semicerrados, o que é característico de sonolência. Um limiar foi estabelecido com base em testes, via uma base de dados online no Kaggle [12], considerando, valores acima de 4,5 como indicativos de sonolência. Já a métrica DSC mede a distância entre o nariz e o ponto médio entre os ombros do motorista, permitindo identificar mudanças de postura associadas à fadiga, como inclinações para frente ou para os lados. Valores acima de 3 indicam comportamentos relacionados à sonolência.

## C. Base de dados e treinamento do modelo YOLOv8

E o último modelo, YOLOv8, que faz a detecção dos olhos, bocas e postura relacionados à sonolência, analisando ambos, abertos ou fechados, ou a inclinação. O YOLOv8 foi treinado para verificar os olhos também, devido ao problema enfrentado nos artigos mencionados. O modelo estará fazendo a verificação final antes da análise final. O YOLOv8 foi treinado utilizando um conjunto de bases de dados do Roboflow [13]. No total, foram 12143 imagens na base de dados, a divisão foi de 80% para treinamento e 20% para validação. A figura 3, em seguida, demonstra a distribuição da base de dados em relação às imagens, onde *Drowsy* significa sonolência e *Awake* significa alerta.

Nas outras características, como boca aberta ou fechada, e inclinação do corpo, cada modelo tem sua resposta específica. Mas o YOLO desempenha um papel crucial na verificação dupla de olhos, fechados ou abertos, além de analisar a postura, também caso se incline para trás ou para os lados, participando

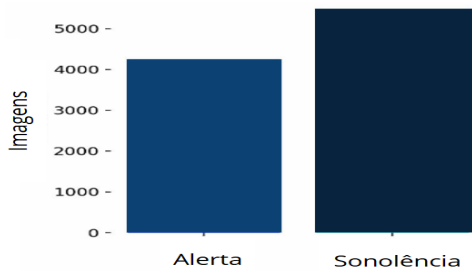


Fig. 3. Base de dados Roboflow [13]

no comitê como um modelo decisivo para a análise final. Analisando a parte e em conjunto, ou seja, o modelo que analisa a face toda e detecta sonolência ou alerta. O modelo YOLO foi treinado com imagens de 640 pixels, usando lotes de tamanho 32, e convergiu em cerca de 446 épocas antes do treinamento ser interrompido devido à estabilização das métricas de precisão, recall, F1-score e mAP.

Veja em seguida o modelo com Yolov8, analisando a face toda, verificando sinais de sonolência. É verificada a face toda, os olhos, a boca e a inclinação, detectando indícios de cansaço. Mas esta verificação não leva em conta especificamente cada parte do corpo, ela analisa a face toda. Abaixo, a figura 4 demonstra estes aspectos de sonolência, nos seguintes casos: olhos fechados, boca aberta, inclinação para trás ou para os lados. E no outro caso, a classe acordada, ou alerta, nos casos: olhos abertos, boca cerrada e postura correta.

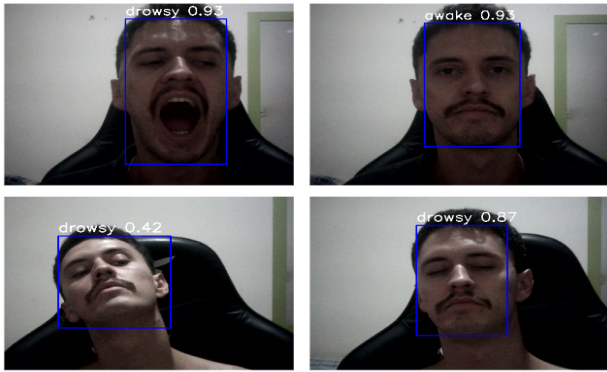


Fig. 4. Detecção da face com segundo modelo do Yolov8.

Para garantir uma resposta eficiente em tempo real, o sistema aciona um alarme sonoro quando 20 frames consecutivos apresentam olhos fechados, boca aberta ou inclinação corporal que ultrapasse os limiares estabelecidos. O alarme, que dura de 2 a 3 segundos, busca alertar o motorista e prevenir acidentes causados pela sonolência. Esta abordagem combina a análise de características faciais e posturais para oferecer um sistema robusto e eficaz na detecção de fadiga ao volante.

A seguir, é demonstrado como o código utiliza o comitê dos 3 modelos e módulos nativos do Python para prever e detectar sinais de sonolência. Veja o pseudocódigo que representa apenas os primeiros modelos deste comitê, no caso o conjunto com *Yolov8n*, *Dlib* e *MediaPIPE*:

---

#### Algorithm 1 Sistema de Alerta de Sonolência e Má Postura

---

```

while captura de vídeo aberta do
  frame ← capturar da câmera (RGB)
  faces ← detectar faces (Dlib)
  pose ← detectar postura (MediaPipe)
  YOLO_results ← executar detecção (YOLO)
  for detecção em YOLO_results do
    classe, confiança, coordenadas ← extrair atributos
    if classe == 'drowsy' then
      Incrementar class_count
      if class_count ≥ limiar and alerta == False then
        thread ← (play_alert_sound)
        alert_triggered ← True
      end if
    else
      class_count, alert_triggered ← Reset
    end if
    Adicionar bounding box e classe ao frame
  end for
  for face in faces do
    landmarks ← pontos faciais
    Calcular razão de piscada:
    D_horizontal ← distância lateral do olho
    D_vertical ← distância vertical do olho
    Razão de piscada ← D_horizontal / D_vertical
    if razão de piscada > limiar then
      frames_closed_eyes ← ++1
    else
      frames_closed_eyes ← Reset
    end if
    if frames_closed_eyes ≥ limiar then
      alerta ← frame
      thread ← (play_alert_sound)
    end if
  end for
  if pose detectada then
    Calcular inclinação corporal:
    nariz ← calcular
    Distância ombro-esquerdo ← calcular
    Distância ombro-direito ← calcular
    Distância meio-ombro ← calcular
    DSC ← (Distância meio-ombro, nariz)
    if DSC > limiar then
      frames_leaning_forward ← ++1
    else
      frames_leaning_forward ← Reset
    end if
    if frames_leaning_forward ≥ limiar then
      alerta ← frame
      thread ← (play_alert_sound)
    end if
  end if
  Mostrar frame atualizado
  if ESC pressionado then
    Encerrar captura
  end if
end while

```

---

A justificativa da utilização do modelo YOLOv8, se deve à precisão que o modelo entrega, e à adaptabilidade da exportação do modelo para ser compatível com edgetpu, além de ter um bom resultado no benchmarking. No modelo do YOLO é analisada a detecção dos olhos fechados ou abertos e a detecção de boca aberta ou fechada, mas em conjunto ele analisa a face toda. O YOLO atua como o modelo decisivo para a análise final. Três datasets foram unidos em um dataset apenas, para maior quantidade de imagens, analisando a face, desde olhos e boca, e a disposição do motorista no carro, analisando se está com olhos em alerta, boca normal, sem bocejo e também a posição da pessoa no banco do motorista [13].

1) *Cálculos Matemáticos:* A razão do piscar dos olhos é calculada como a razão entre a distância horizontal entre os cantos dos olhos e a distância vertical entre os centros superior e inferior dos olhos. A distância horizontal é obtida através dos pontos faciais com 68 pontos, da biblioteca Dlib. A distância horizontal é maior do que a distância vertical. Sendo que a distância vertical é a que se altera durante o piscar dos olhos, assim é possível ter um valor mais normalizado utilizando esta razão. E também uma quantidade mínima de tempo de frame, que é mantida quando os olhos não simplesmente estão piscando, mas sim fechados. Quando ocorre a piscada, basicamente a distância vertical fica pequena, portanto no denominador temos um valor pequeno, o que retorna um valor acima de um limiar, o que basicamente detecta a piscada.

Para se calcular a distância entre o ponto médio dos olhos, na parte vertical, para encontrar o valor de BR, e também o ponto médio entre o ombro esquerdo e direito, para encontrar o valor de DSC, são utilizadas as equações 1 e 2. Ou seja, o ponto médio do ombro e o nariz possuem um ponto x e y.

$$P_{\text{médio}} = \left( \frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right) \quad (1)$$

$$DSC = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (2)$$

Após o cálculo do ponto médio, são traçadas linhas com a distância euclidiana na equação 2. Com estas linhas é obtida a distância horizontal e vertical dos olhos, com isto é possível calcular BR através da equação 3 abaixo.

$$BR = \frac{\text{DistânciaHorizontal}}{\text{DistânciaVertical}} \quad (3)$$

Esses cálculos todos, em conjunto, formam o que é denominado de características que empregam e demonstram sonolência do usuário de forma rápida. Por fim, o modelo com YOLO auxilia a estes cálculos serem mais efetivos, sendo o modelo decisivo no comitê que faz a análise completa e final do sistema.

#### IV. RESULTADOS

Os testes realizados demonstraram a eficácia do sistema em detectar indícios de sonolência e má postura, isto se deve a testes em vídeos, como a base de dados utilizada para uma breve comparação, analisando momento de sonolência e não.

A base de dados está referenciada no Kaggle, [12] e pode ser acessada abertamente. Esta base auxiliou na análise do limiar a ser escolhido para os módulos nativos ou importados através do Python, Dlib e MediaPIPE.

Os casos foram monitorados analisando os seguintes parâmetros:

##### A. Alerta de Sonolência

- **Tempo de Detecção:** 4 segundos (equivalente a 120 frames).
- **Condição:** Olhos fechados por 5 frames consecutivos.
- **Ação:** Alerta visual e sonoro.

##### B. Alerta de Má Postura

- **Tempo de Detecção:** 3 segundos (equivalente a 90 frames).
- **Condição:** Distância nariz-ombros menor que o limiar de 0.95 metros.
- **Ação:** Alerta visual e sonoro.

Os resultados indicam que o sistema é robusto para variações de luz e postura, desde que a posição inicial seja configurada corretamente. Veja a figura 5, mostrando alguns casos de sonolência e indisposição que atuam em conjunto.

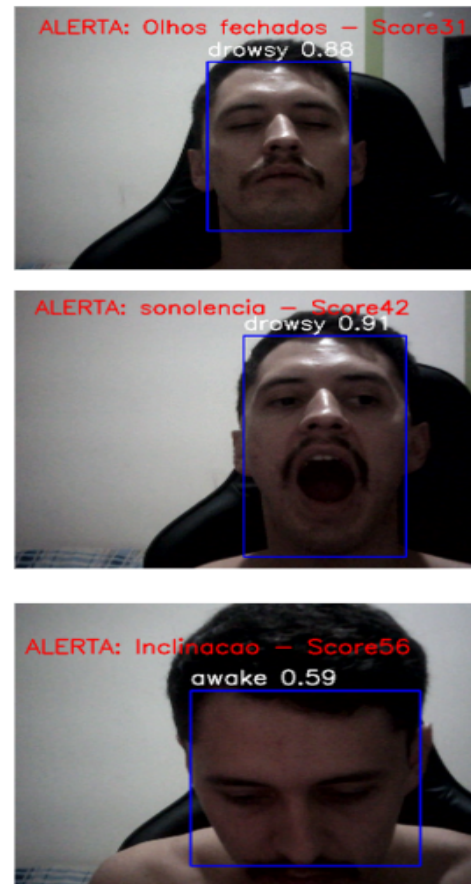


Fig. 5. Alerta de sonolência, com Dlib, MediaPIPE e YOLOv8

O modelo do YOLO se demonstra muito eficiente para detecção de sonolência. Alguns casos específicos, como



inclinação para frente, não são captados, porém, inclinação para trás, olhos fechados ou boca fechada, o Yolo detecta perfeitamente. A acurácia na matriz de confusão para predição de alerta foi de 98% e de sonolência foi de 98%. Veja na figura 6 abaixo a matriz de confusão do resultado do treinamento do YOLOv8.

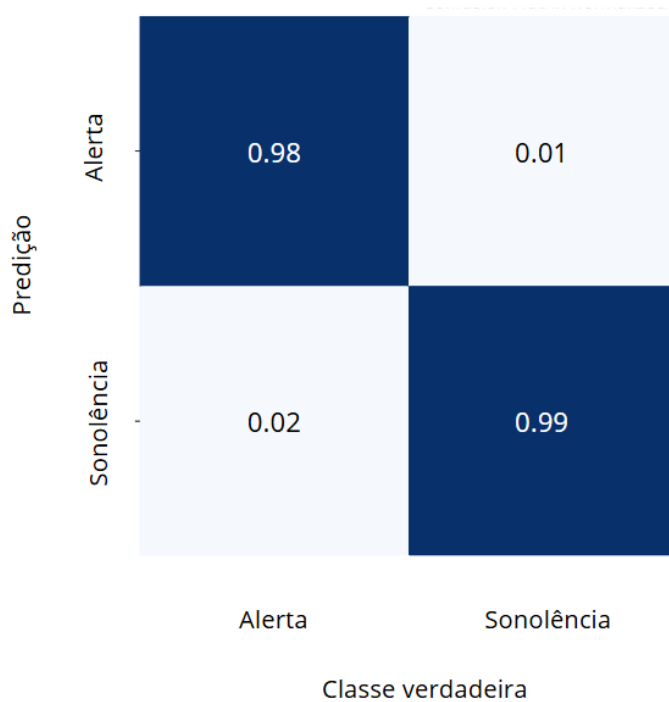


Fig. 6. Matriz de confusão YOLOv8

Na matriz de confusão no resultado, o YOLOv8 participa na detecção de alerta, quando está com a postura correta, sem olhos fechados, por muito tempo, quando está com a boca fechada sem estar bocejando, e também quando a postura está corretamente fixa no banco do carro, com os olhos atentos para os lados e para frente. Agora, por outro lado, a outra classe, que seria Sonolência, é quando está o contrário do que foi comentado anteriormente, olhos fechados, bocejando e inclinando, demonstrando cansaço excessivo. Conforme a matriz de confusão, a acurácia do modelo com o resultado da validação foi de aproximadamente de 98,5%.

O desempenho do modelo também pode ser visto no gráfico em sequência. Mesmo que a base de dados para treinamento do YOLOv8 ainda esteja bem balanceada, é necessário avaliar outras métricas para entender corretamente se o modelo estaria performando corretamente. Essas métricas são, F1-Score, precision, recall e mean average Precision, mAP-50. A curva de precisão e recall ajuda a entender a relação entre a capacidade do modelo de identificar corretamente as amostras positivas (recall) e a proporção de previsões positivas corretas feitas pelo modelo (precisão).

No gráfico da figura 7, apresentado, é possível observar que o modelo mantém altos valores de precisão e recall para ambas as classes (awake e drowsy), com um mAP@0.5 geral

de 0.992. Isso indica que o modelo está conseguindo detectar corretamente a maioria das amostras com um bom equilíbrio entre precisão e recall.

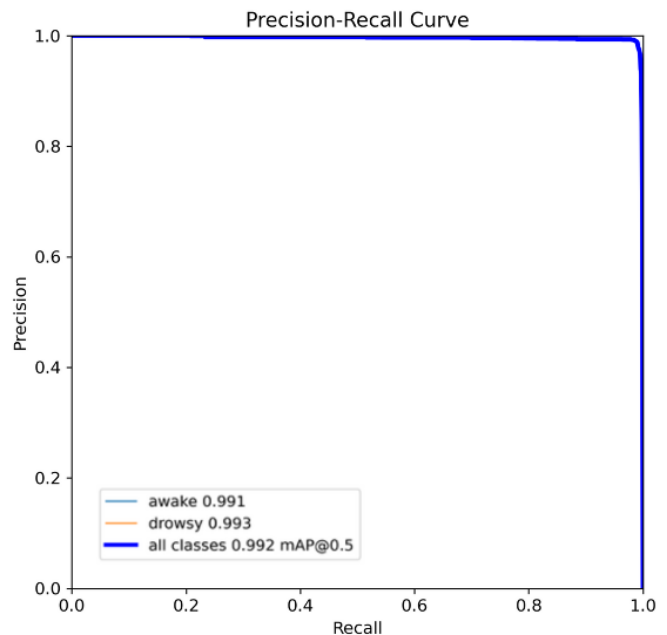


Fig. 7. Curva de Precisão-Recall YOLOv8

Outras métricas avaliadas são as que têm relação com o dispositivo que irá rodar o comitê dos modelos e o sistema. Essas métricas são: O consumo de CPU, RAM, o tamanho do modelo em si, a quantidade de FPS atingida total pelo modelo. Essas métricas podem ser vistas abaixo na tabela I. Em seguida, o desempenho do modelo, é validado, sendo compilado, junto da Raspberry Pi 5, o dispositivo acelerador Edge TPU. E as métricas são similares às métricas anteriores, e o desempenho deste foi escolhido apenas o peso float32, isto vem por padrão quando exporta o formato como *edgetpu* para poder quantizar com o formato *edgetpu* e exportar este.

TABLE I  
MÉTRICAS DE DESEMPENHO NO DISPOSITIVO RASPBERRY PI 5 MODEL B

	FPS	CPU	RAM
float32	2	25%	630 MB
float16	3.5	23%	610 MB
int8	4	20%	600 GB
onnx	5	21%	605 GB
edgetpu	6	15%	480 GB

O desempenho não é muito alto devido à exportação do modelo ter sido em 640 pixels, e principalmente, por ter outros dois modelos e módulos que estão competindo pelo uso do dispositivo, isto faz com que o desempenho não seja tão efetivo.

O sistema foi codificado para rodar no Streamlit, e mostrar os alarmes escritos quando é detectado alguma sonolência ou extremo cansaço na pessoa, ou motorista, veja na figura 8,

a seguir, sobre o aplicativo em Streamlit para poder verificar isto.

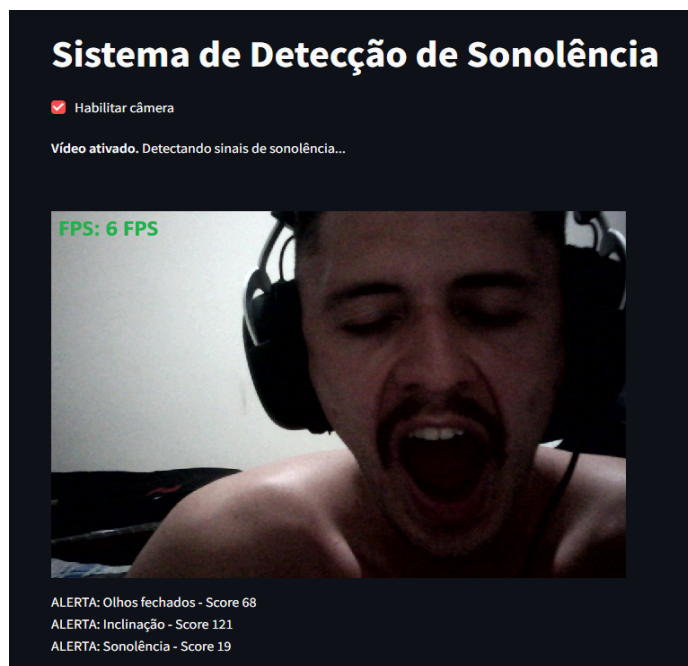


Fig. 8. Aplicativo em Streamlit para detecção de sonolência

Foi extraído de uma captura de vídeo do dataset mencionado, e obtido e salvo em csv, para poder analisar sobre as features importantes. Foi feito a diminuição de dimensionalidade para analisar, usando PCA em 2 dimensões, para avaliar sobre as métricas, como elas se comportam, se estariam os dados mais separados do que unidos. Esta figura abaixo representa um dado de teste usado de vídeo para representar a separação dos dados, em uma dimensionalidade inferior. Veja na figura 9 a seguir.

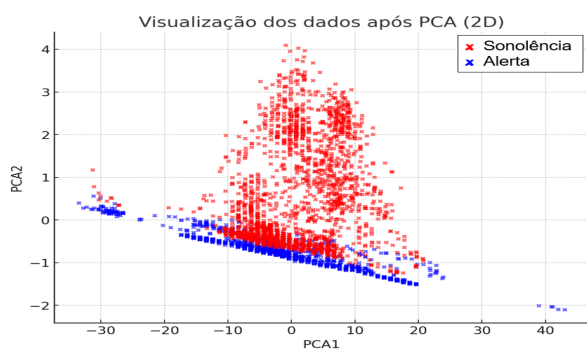


Fig. 9. PCA em duas dimensões para análise das features em relação a um vídeo de teste em um frame

Este mapa de características pode ser usado para treinar um modelo de rede neural de acordo com frames do vídeo. Veja que as características importantes são apenas duas que descrevem bastante sobre sonolência e acordado. Destas, poderia separar clusters para treinamento de classificação de

sonolência, que poderia auxiliar no comitê dos modelos. Isto é um trabalho futuro, detalhe a ser debatido e incorporado no projeto.

## V. CONCLUSÃO

Este sistema apresenta uma solução acessível e eficaz para monitorar sinais de sonolência e má postura em motoristas. Em contraste com tecnologias caras, esta abordagem utiliza ferramentas de código aberto e hardware acessível, permitindo sua implementação em larga escala, especialmente para motoristas profissionais no Brasil. Proporciona maior segurança no trânsito, reduzindo acidentes causados por sonolência e distração. O sistema desempenha bem com a utilização do modelo exportado como int8, onnx e edgetpu. O desempenho demonstra a detecção correta da sonolência quando a pessoa ou o motorista demonstra sinais de sonolência com os olhos, lentamente fechando, com também o bocejo caso a pessoa esteja começando a ter sinais de cansaço, e também a postura relacionada a um cansaço extremo. A integração com Streamlit resultou em uma aplicação que poderia ser aplicada ao mobile, ou sistema integrado com um carro, diretamente. Para uma melhoria futura, poderia ser feita a integração com sistemas automotivos para desaceleração automática, e estacionamento seguro de acordo com carros inteligentes ou outra aplicação que conseguisse monitorar o trânsito ao redor do veículo do motorista. Otimizações também poderiam ser feitas, utilização de resolução de modelo menor para acelerar mais ainda o processamento dos dados, por exemplo, 240, ou 192. Outro ponto que é um possível problema são as detecções do modelo para condições adversas, como baixa luz, entre outras.

## VI. AGRADECIMENTO

This work was partially supported by CNPq (Grant References 311470/2021-1 and 403827/2021-3), by the projects XGM-AFCCT-2024-2-5-1, XGM-FCRH-2024-2-1-1, and XGM-AFCCT-2024-9-1-1 supported by xGMobile - EMBRAPII-Inatel Competence Center on 5G and 6G Networks, with financial resources from the PPI IoT/Manufatura 4.0 from MCTI grant number 052/2023, signed with EMBRAPII, by RNP, with resources from MCTIC, Grant No. 01245.020548/2021-07, under the Brazil 6G project of the Radiocommunication Reference Center (Centro de Referência em Radiocomunicações - CRR) of the National Institute of Telecommunications (Instituto Nacional de Telecomunicações - Inatel), Brazil, and by Fapemig (PPE-00124-23, APQ-00810-21, APQ-04523-23, APQ-05305-23, and APQ-03162-24).

## REFERENCES

- [1] M. R. Alves, "Fatores associados a acidentes de trânsito em pedestres: Rondônia, amazônia ocidental brasileira, 1998-2022," *GEOGRAFIA*, vol. 49, no. 1, pp. 317–337, 2024.
- [2] A. C. F. d. Oliveira and J. Q. Pinheiro, "Indicadores psicossociais relacionados a acidentes de trânsito envolvendo motoristas de ônibus," *Psicologia em Estudo*, vol. 12, pp. 171–178, 2007.
- [3] K. R. CUNHA, "Análise de acidentes de trânsito de caminhoneiros na perspectiva da análise da atividade," 2016.
- [4] G. I. d. F. TIOSSO, "Transporte rodoviário de carga: a importância da utilização de câmeras de fadiga no monitoramento do transporte rodoviário brasileiro," 2024.

- [5] H. Silva, *Tecnologias e aplicações práticas de mobilidade*. Editora Senac São Paulo, 2023.
- [6] M. M. Tavares, *Desenvolvimento de um sistema electrotêxtil para aferição objectiva dos movimentos corporais durante o sono: correlação com a polissonografia e a qualidade do sono*. PhD thesis, Universidade da Beira Interior (Portugal), 2021.
- [7] D. Mikelsten, *Automação e Tecnologias Emergentes*, vol. 3. Cambridge Stanford Books, 2009.
- [8] L. C. Guimarães, “Geração de dispositivo sinalizador de sonolência para conduta de veículos a fim de conter acidentes de trânsito,” 2021.
- [9] G. N. Astrissi *et al.*, “Detecção de distração e sonolência em condutores veiculares utilizando redes neurais convolucionais,” 2023.
- [10] M. T. A. Dipu, S. S. Hossain, Y. Arafat, and F. B. Rafiq, “Real-time driver drowsiness detection using deep learning,” *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 7, 2021.
- [11] C. Zhou, Y. Zhao, S. Liu, Y. Zhao, X. Li, and C. Cheng, “Research on driver facial fatigue detection based on yolov8 model,” 2024.
- [12] R. Ghoddoosian, M. Galib, and V. Athitsos, “A realistic dataset and baseline temporal model for early drowsiness detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.
- [13] projectsautonomos, “deteccao de sonolencia dataset.” <https://universe.roboflow.com/projectsautonomos/deteccao-de-sonolencia-7h4d7> , nov 2024. visited on 2024-11-24.