

# Protótipo de um sensor de fadiga com detecção de pontos faciais

## *Prototype of a fatigue sensor with facial point detection*

Luiz Inácio Chaves<sup>1</sup>  
 Wylliam Salviano Gongora<sup>2</sup>  
 Michelli Cristina Galli<sup>3</sup>  
 Thiago Berticelli Ló<sup>4</sup>  
 André Luiz do Nascimento Moura<sup>5</sup>  
 Gustavo Henrique Silveira Manoel<sup>6</sup>  
 Jorge Luiz de Oliveira Pinto<sup>7</sup>  
 Pedro Henrique Mauloni<sup>8</sup>

### Resumo

No Brasil, muitos acidentes de trânsito são registrados anualmente. Estatísticas indicam que cerca de 60% desses acidentes decorrem de sono e cansaço. O uso da tecnologia reduz em até 86% dos potenciais acidentes. Diante disso, um protótipo de um sistema embarcado foi desenvolvido, visando à detecção da fadiga. O objetivo é detectar o sinal de exaustão por meio de um sistema embarcado. A partir disso, o sistema emite um alerta, sugerindo ao condutor que pare e descanse. O módulo é composto por um sistema embarcado do modelo *Raspberry Pi 3*, utilizando a câmera *Pi Camera V2*, um LED e um *Buzzer*. A linguagem *Python* foi utilizada, acrescida das bibliotecas *OpenCV* e *DLIB*. Após a análise do sistema, a métrica  $R^2$  dos testes, variando o *Frame check* foi 0,981, já para o *Frame width* foi 0,8779. Os resultados obtidos com o protótipo demonstraram latência de resposta inferior a dois (2) segundos.

**Palavras-chave:** acidentes de trânsito; detecção facial; sensor de fadiga.

### Abstract

*In Brazil, many traffic accidents are recorded annually. Statistics indicate that about 60% of these accidents are due to sleep and fatigue. The use of technology reduces potential accidents by up to 86%. In view of this, a prototype of an embedded system was developed, aiming at fatigue detection. The objective is to detect the exhaustion signal through an embedded system. From this, the system issues an alert, suggesting the driver to stop and rest. The module consists of an embedded system of the Raspberry Pi 3 model, using the Pi Camera V2 camera, an LED and a Buzzer. The Python language was used, plus the OpenCV and DLIB libraries. After analyzing the system, the  $R^2$  metric of the tests varying the Frame check was 0.981, while for the Frame width it was 0.8779. The results obtained with the prototype showed response latency of less than two (2) seconds.*

**Keywords:** traffic accident; facial detection; fatigue sensor.

1 Doutorando em Engenharia Elétrica pela Faculdade de Engenharia Elétrica e Computação pela Universidade Estadual de Campinas (FEEEC-UNICAMP), Campinas, SP, Brasil. Professor do Instituto Federal do Paraná Campus Assis Chateaubriand (IFPR), Assis Chateaubriand, PR, Brasil. ORCID: <https://orcid.org/0000-0001-5012-1351>. E-mail: [luiz.chaves@ifpr.edu.br](mailto:luiz.chaves@ifpr.edu.br)

2 Doutor em Engenharia Elétrica pela Universidade São Paulo (USP), São Paulo, SP. Professor do IFPR, Assis Chateaubriand, PR. ORCID: <https://orcid.org/0000-0001-7943-8141>. E-mail: [wylliam.gongora@ifpr.edu.br](mailto:wylliam.gongora@ifpr.edu.br)

3 Doutora em Estudos da Linguagem pela Universidade Estadual de Londrina (UEL), Londrina, PR. Professora do IFPR, Assis Chateaubriand, PR. ORCID: <https://orcid.org/0009-0002-5277-4625>. E-mail: [michelli.galli@ifpr.edu.br](mailto:michelli.galli@ifpr.edu.br)

4 Doutor em Engenharia Agrícola pela Universidade Estadual do Oeste do Paraná (UNIOESTE), Cascavel, PR. Professor do IFPR, Cascavel, PR. ORCID: <https://orcid.org/0000-0002-1457-8810>. E-mail: [thiago.lo@ifpr.edu.br](mailto:thiago.lo@ifpr.edu.br)

5 Acadêmico em Física pela Universidade Rural Federal de Pernambuco (UFRPE) e técnico em Eletromecânica pelo IFPR. ORCID: <https://orcid.org/0009-0000-4771-5124>. E-mail: [erdnamour@gmail.com](mailto:erdnamour@gmail.com)

6 Acadêmico em Engenharia Mecânica pela Fundação Assis Gurgacz (FAG), Cascavel, PR e técnico em Eletromecânica pelo IFPR. ORCID: <https://orcid.org/0009-0007-0988-8199>. E-mail: [gustavomanoel.busiquiasilveira@gmail.com](mailto:gustavomanoel.busiquiasilveira@gmail.com)

7 Acadêmico em Ciências da Computação pela UNIOESTE e técnico em Eletromecânica pelo IFPR. ORCID: <https://orcid.org/0009-0002-1756-9298>. E-mail: [jorgeoliveira21082004@gmail.com](mailto:jorgeoliveira21082004@gmail.com)

8 Acadêmico em Engenharia Elétrica pela FAG e técnico em Eletromecânica pelo IFPR, Iracema do Oeste, PR. ORCID: <https://orcid.org/0009-0009-9339-0848>. E-mail: [pedromauloni@gmail.com](mailto:pedromauloni@gmail.com)



## 1 Introdução

Uma das maiores causas de acidentes de trânsito do mundo é o sono ao volante. Ao ano, cerca 1,35 milhões de vidas são perdidas mundialmente em decorrência dos acidentes de trânsito. Entre os principais motivos estão a ingestão de álcool, a falta de atenção ao dirigir, o excesso de velocidade e a sonolência (WORLD HEALTH ORGANIZATION, 2018). No Brasil, em 2021, segundo um levantamento feito pelo Painel da Conferência Nacional do Transporte (CNT), com dados da Polícia Rodoviária Federal (PRF), 64.539 acidentes rodoviários foram registrados (CONFEDERAÇÃO NACIONAL DO TRANSPORTE, 2022).

Uma alternativa para auxiliar o motorista ao volante é a utilização da tecnologia para detecção de fadiga, cujas características faciais são detectadas por sensores e recomendam a parada para um descanso, evitando que o motorista durma ao volante. Essa tecnologia se caracteriza pelo sensor de fadiga encontrado, principalmente, em veículos *premium* e ainda pouco utilizado.

O sensor de fadiga é um sistema que pode analisar o comportamento do condutor e reconhecer, quando há sinais de sono ou cansaço, emitindo um alerta em tempo real, a fim de ajudar o motorista a permanecer focado (TIEMI, 2021). Há uma variedade de sensores de fadiga no mercado, desde os mais simples até os mais completos, capazes de identificar até sete sinais no comportamento que podem causar acidentes, tais como: fadiga; distração; uso do celular; distância perigosa entre os veículos; fumar enquanto dirige; não utilização do cinto de segurança e troca de pista sem usar um alerta.

O código base do sensor de fadiga foi idealizado por Adrian Rosebrock no seu artigo *Facial landmarks with DLIB, OpenCV, and Python*; posteriormente modificado por Angel Jude Suárez (2021), com o objetivo de detectar a sonolência, algumas alterações foram feitas nos alertas e na distância euclidiana dos olhos. Na face, os pontos de referência podem ser detectados de diversas formas, mas, essencialmente, todos os métodos são para verificar e identificar as regiões da boca, olhos, sobrancelhas, nariz e queixo. As bibliotecas *DLIB* e *OpenCV* foram utilizadas com o intuito de localizar o rosto e, em seguida as coordenadas, para então mapear as estruturas faciais (ROSEBROCK, 2017).

Em 2012, o sistema chegou ao Brasil trazido pela Volkswagen, nomeado inicialmente de “detector de fadiga”, cujos princípios de funcionamento eram o

ângulo de esterçamento do volante, a pressão aplicada nos pedais e as acelerações em volta do veículo. Esse sistema faz uma comparação dos dados coletados, desde o início da viagem, e, no caso de detecção de fadiga, sugere que o condutor faça uma pausa (PARANHOS, 2019).

Diante do exposto, este trabalho teve como objetivo desenvolver um sistema embarcado de baixo custo para veículos, capaz de realizar o reconhecimento facial, a fim de evitar acidentes causados por sonolência. Sendo assim, um dos propósitos iniciais foi o entendimento da linguagem *Python*, para a programação do sensor, bem como a determinação de metodologias adequadas para o projeto. Em seguida, a partir de testes, verificou-se a versatilidade do sistema com intuito de diminuir as falhas e os erros, para então configurar o projeto como uma ferramenta capaz de garantir maior segurança aos consumidores, durante a condução, além do bom custo-benefício e da confiabilidade na utilização, através da comparação com ferramentas homologadas e comercializadas no mercado.

## 2 Materiais e métodos

### 2.1 Fluxograma do trabalho

Para atender aos critérios de execução, o módulo computadorizado é composto por um sistema embarcado, modelo *Raspberry Pi 3*, que, por meio de uma câmera integrada, permite a detecção da fadiga. A instrumentalização do sistema engloba uma simulação computacional, utilizando a linguagem de programação em *Python* para compilação do reconhecimento facial, com a aplicação da biblioteca *OpenCV*, uma das principais linguagens usadas para o processamento de imagens em tempo real, reconhecimento de objetos, análise estrutural entre outros.

Abaixo serão apresentadas breves explicações sobre os blocos do fluxograma da figura 1:

Conforme exposto, cada etapa configura:

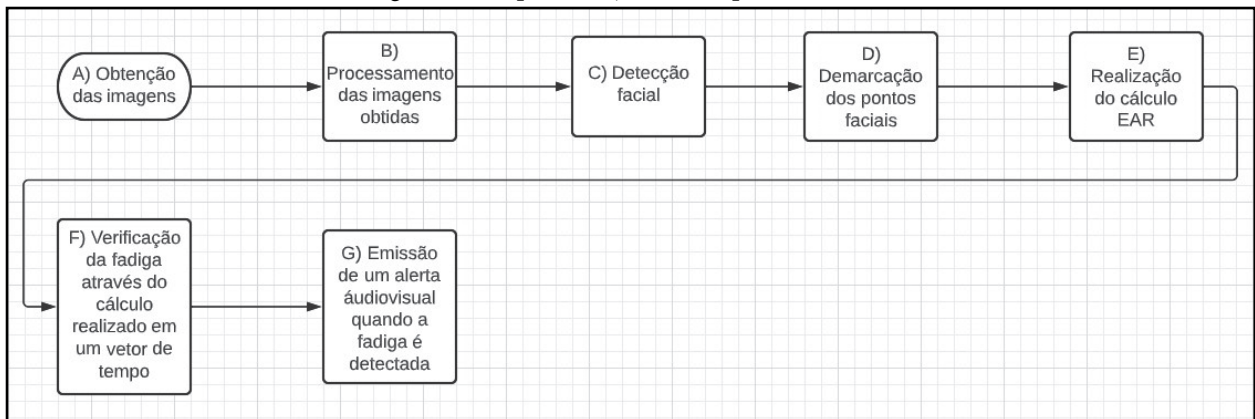
- A. obtenção de imagens: este bloco tem a responsabilidade de adquirir as imagens, a partir de uma câmera;
- B. processamento das imagens obtidas: nesse bloco, as imagens obtidas são processadas; no processamento são feitos os ajustes, as correções e as conversões;
- C. detecção facial: é efetuado o algoritmo de Viola-Jones para o reconhecimento da face, com auxílio da biblioteca *OpenCV*;
- D. demarcação dos pontos faciais: os pontos usados de referência são localizados para a

- demarcação do rosto e também são realizados os ajustes da imagem;
- E. realização do cálculo EAR (*Eye Aspect Ratio*): é realizado o cálculo da relação do aspecto do olho humano, para que o código possa fazer a detecção da fadiga;
- F. verificação da fadiga, a partir do cálculo realizado em um vetor de tempo: quando

realizado o cálculo, passa-se para a próxima etapa que verifica se a relação do aspecto do olho está abaixo do que é predefinido como “normal”;

- G. emissão de uma alerta audiovisual, quando a fadiga é detectada: nesse bloco são executados os sinais de alerta visuais e sonoros ao motorista, no caso de detecção da fadiga.

Figura 1 – Representação das etapas do sistema



Fonte: Os autores (2022).

## 2.2 Python

*Python* é uma linguagem de programação de alto nível. Apresenta uma sintaxe clara e consistente, considerada uma linguagem livre, pois pode ser usada em qualquer aplicação, seja de jogos 2D até inteligência artificial e *Machine Learning* (PARANHOS, 2019). Consiste no aprendizado pelas máquinas, a partir de respostas e dados obtidos por associações.

A linguagem de programação *Python* está presente no *Raspberry Pi*, por meio do IDE (*Integrated development and learning environment*). Possui as versões *Python 2* e *Python 3*, a diferença entre elas está relacionada, principalmente, à sintaxe (PARANHOS, 2019).

## 2.3 Algoritmo de Viola-Jones

A técnica de reconhecimento de padrões, conhecida como algoritmo de Viola-Jones, foi desenvolvida por Paul Viola e Michael Jones e consiste em encontrar partículas de imagens que apresentam diferentes contrastes. Essas frações são denominadas de *Características de Haar* e são encontradas, durante o treinamento de algum algoritmo. O algoritmo possui, como principal característica, um bom resultado na detecção de objetos e faces. Para extrai-las, normalmente as imagens são convertidas em tons de cinza

(PAIXÃO, 2018). Como mostra a figura 2, o classificador Haar utiliza alguns recursos para a extração de características de uma imagem e, posteriormente, serve como base no processo de detecção de objetos e/ou face.

O classificador utiliza a cascata Haar: um montante de características representadas por bordas, linhas e centro, cujas combinações são denominadas *features Haar*. Utilizando-se essas *features*, é possível a detecção de rostos e objetos que, por meio de padrões retangulares nos dados obtidos, analisa as diferentes intensidades das regiões.

A face é identificada apenas se passar pela cascata inteira, caso contrário, será rejeitada, como ilustra a figura 3.

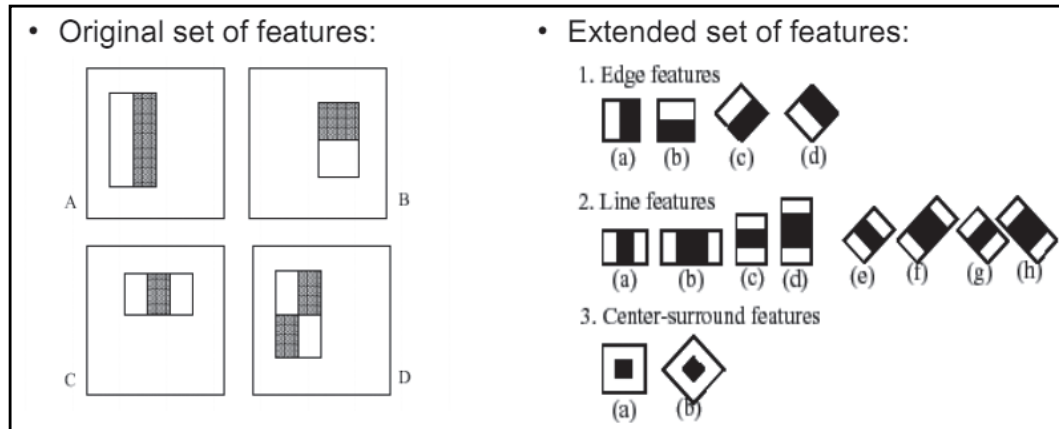
## 2.4 Detecção dos pontos de referência faciais

O objetivo da detecção de pontos de referência faciais é identificar, automaticamente, a localização de pontos chave do rosto em vídeos ou imagens faciais. Esses pontos chave são pontos dominantes, descrevendo uma localização única de algum componente facial ou um ponto conectando esses pontos dominantes, em volta do contorno facial e dos componentes faciais (WU; JI, 2018). Essa técnica de análise é utilizada para

localizar e representar regiões características do rosto, tais como: olhos, sobrancelhas, nariz, boca e maxilar. As principais aplicações desses pontos

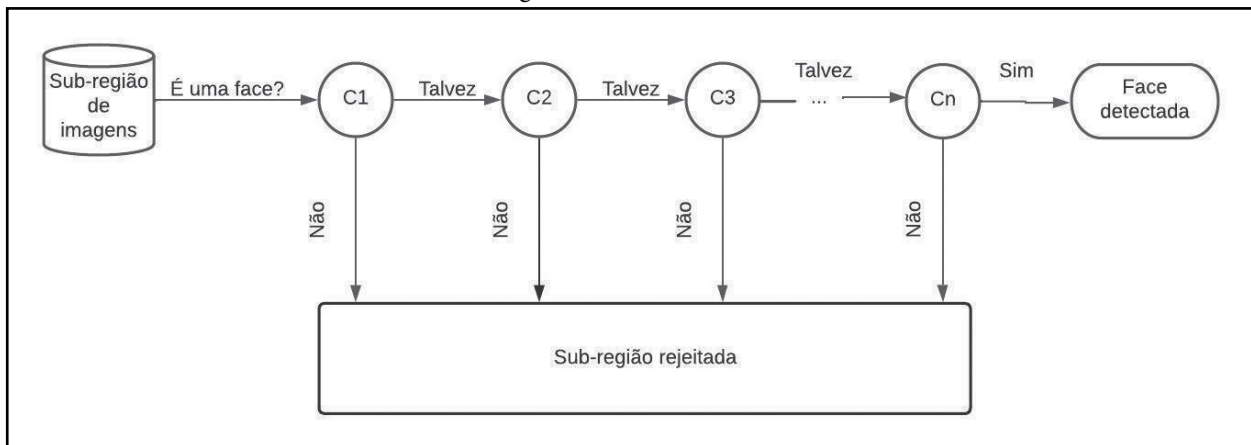
faciais são para a detecção de piscadas, estimação de pose da cabeça, alinhamento de rosto e entre outras (ROSEBROCK, 2017).

Figura 2 – Classificador Haar



Fonte: Paranhos (2019).

Figura 3 – Cascata Haar



Fonte: Os autores (2022).

Para a detecção, é utilizada a biblioteca *DLIB* que integra um conjunto de ferramentas no qual foi usado um detector de pontos faciais que estima a localização de 68 coordenadas (x, y), mapeando a estrutura facial do rosto, como mostra a figura 4. Essas coordenadas são parte de um modelo de predição de formas chamado *iBUG300-W*. Nele o detector de pontos faciais da biblioteca é treinado (ROSEBROCK, 2017). Os dados das coordenadas são armazenados em um vetor que pode ser usado apenas ao selecionar o intervalo que a aplicação necessita (FERREIRA; MARQUES; BRAZ JUNIOR, 2018). Esse vetor é assim determinado:

- boca: [49, 68];
- nariz: [28, 36];

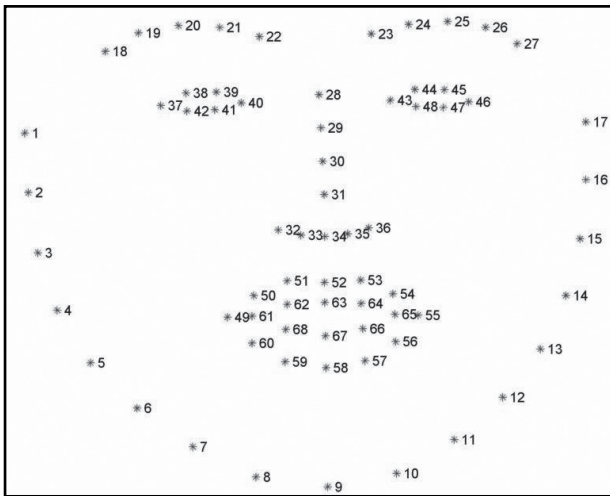
- olho direito: [37, 42];
- olho esquerdo: [43, 46];
- sobrancelha direita: [18, 22];
- sobrancelha esquerda: [23, 27];
- mandíbula: [1, 17].

### 2.5 Eye Aspect Ratio

Cada pessoa tem um padrão de piscadas que se difere um pouco um do outro, essa variação está ligada à velocidade de fechamento e abertura do olho, a intensidade ao apertar o olho e a duração da piscada; uma piscada dura aproximadamente de 100 a 400 milissegundos (SOUKUPOVÁ; CECH, 2016).



Figura 4 – Pontos de referência faciais



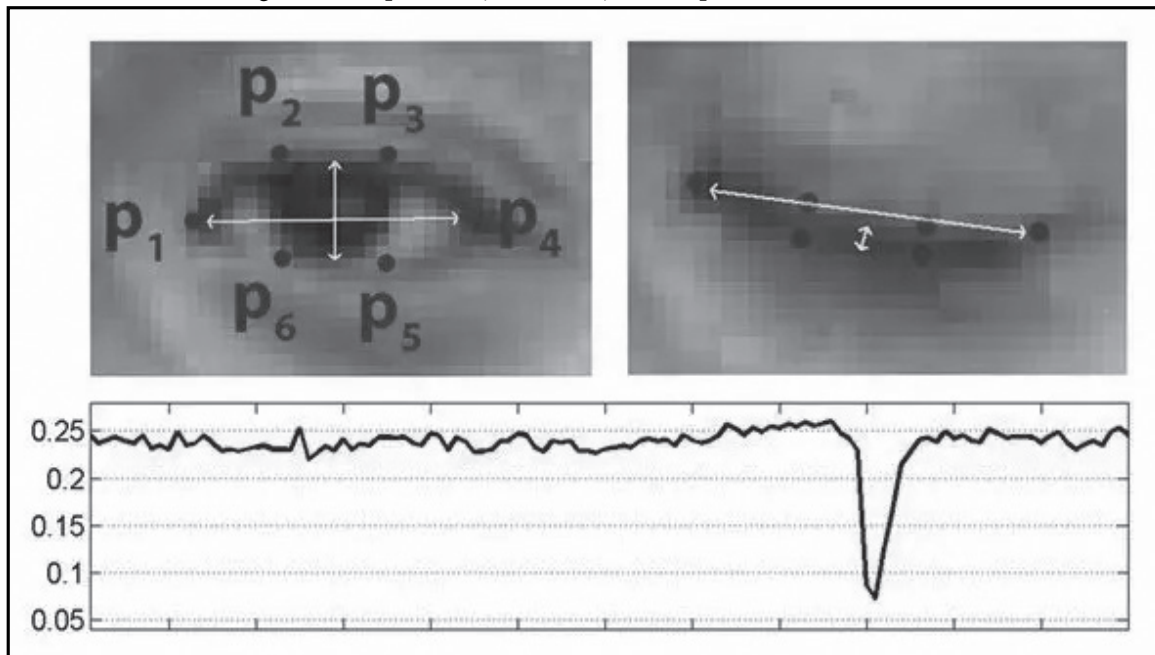
Fonte: Sagonas *et al.* (2013).

A relação de aspecto do olho (EAR – *Eye Aspect Ratio*) é uma quantidade escalar obtida dos pontos de referências faciais de um rosto em alguma imagem, encontrando a distância das coordenadas dos olhos correspondentes, substituindo-a na seguinte fórmula (KUWAHARA *et al.*, 2022):

$$EAR = \frac{|p2-p6| + |p3-p5|}{2|p1-p4|} \quad \text{Eq. (1)}$$

Nessa equação, os pontos de p1 a p6 correspondem às coordenadas dos pontos faciais, referentes aos olhos, retratados em um plano 2D, figura 5. O piscar dos olhos é realizado simultaneamente e a média do EAR de ambos os olhos é calculada. Por essa fórmula de cálculo ser simples, possui um bom desempenho em tempo real, no entanto, se houver um movimento súbito do rosto ou, se estiver longe da câmera, a precisão diminui consideravelmente (KUWAHARA *et al.*, 2022).

Figura 5 – Representação da relação de aspecto do olho (EAR)



Fonte: Soukupová e Cech (2016, p. 1).

## 2.6 Capacidade de detecção

A capacidade de detecção do sistema se dá pelo tamanho da janela em que ocorre o sinal visual de “Alerta” e a demarcação dos olhos. A distância de detecção pode ser alterada ao mudar o tamanho do *frame*.

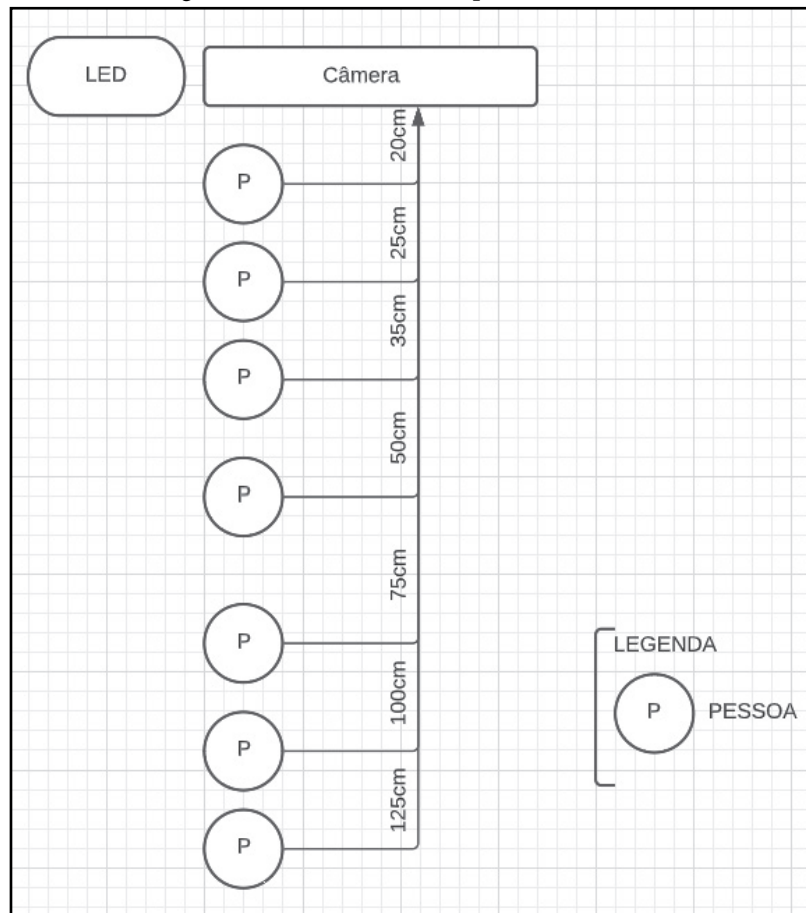
Quanto à verificação do tempo de processamento em diferentes tipos de *hardware*, foi utilizado um computador *desktop* com configuração mediana, o *Raspberry Pi Model 3 B+* e, em dois diferentes tipos

de câmera, uma *webcam* e a *Pi Camera*. Nesse experimento, foram selecionados quatro indivíduos; o teste considerou o momento em que a face foi detectada até o alerta de fadiga.

### 2.6.1 Teste de capacidade com variação da distância

Os testes foram realizados em ambiente com luminosidade parcialmente controlada, variando o posicionamento da pessoa. A figura 6 mostra a distância dos testes em relação à câmera.

Figura 6 – Distância entre a pessoa e a câmera



Fonte: Os autores (2022).

O indivíduo, a ser detectado, variou sua distância em relação ao sensor, essa variação foi de 20 a 125cm com algumas distâncias pré-definidas ao longo dos testes.

#### 2.6.2 Teste de detecção com variação de luminosidade

Os testes de variação da iluminação foram realizados com auxílio de uma luz de LED, na qual foi possível fazer alteração da intensidade luminosa e um luxímetro, para a medição da expressividade de luz emitida na face do indivíduo. O teste variou a distância da pessoa acompanhada do luxímetro e, também, a quantidade de luz incidida na face a ser detectada.

#### 2.6.3 Teste de variação do tamanho da janela de detecção (*frame width*) e análise contínua de quadros (*frame check*)

Para realizar o mapeamento do rosto e a sua detecção, o sistema utiliza a imagem do *frame*, capturada pela câmera. Quanto maior a designação do tamanho do *frame*, maior a janela aparece na tela, facilitando a detecção facial e, conseqüentemente, fazendo com

que o sistema seja capaz de mapear os olhos, mesmo com o condutor mais afastado da câmera. Em contrapartida, a imagem aumentada requer um maior processamento, causando a defasagem no tempo de detecção da fadiga pelo sistema, visto que a taxa de QPS (quadros por segundo) da janela é diminuída. Esse ponto tem efeito negativo no processo de análise e reconhecimento do cansaço; o sistema realiza o *Frame check*, analisando continuamente os quadros da imagem e, caso identifique um número de quadros igual e/ou maior ao indicado e em cuja exaustão seja verificada, reconheça-a e emita o sinal, ou seja, quanto maior a lentidão para a amostragem de quadros, maior o atraso no funcionamento do sistema.

#### 2.7 Ligação dos componentes eletrônicos

No sistema embarcado, foram implementados alertas sonoros e visuais, para chamar a atenção do condutor, caso a exaustão seja constatada. Nos alertas sonoros, foi usado um *Buzzer*, conectado nos pinos do GPIO do *Raspberry Pi*, que são portas programáveis

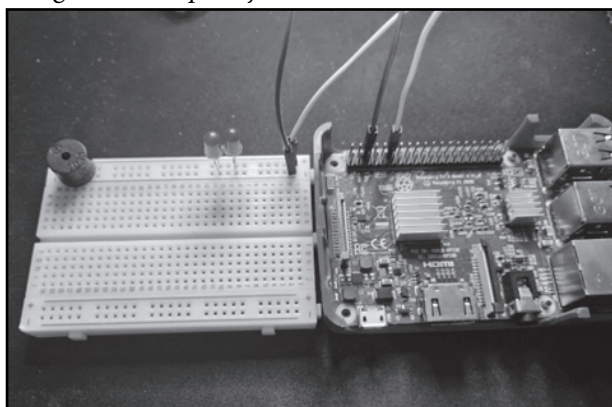
de entradas e saídas de dados. Já nos alertas visuais, foram usados LED.

### 3 Resultados e discussões

#### 3.1 Implementação

O sistema embarcado foi implementado em um *Raspberry Pi 3*, com sinais de alerta sonoros e visuais, para sinalizar a fadiga ao condutor, conforme figura 7.

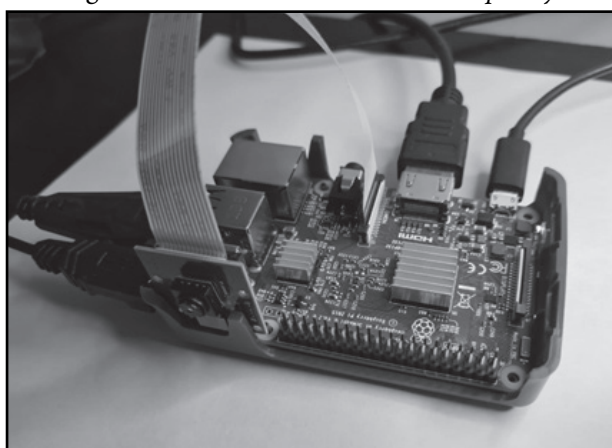
Figura 7 – *Raspberry Pi 3* com os sinais dos alertas



Fonte: Os autores (2022).

Para a identificação das características faciais de fadiga, utilizamos uma câmera com sensor Sony IMX219, resolução de 1080p 30, 720p 60, 640/480p 60/90, conectada ao *Raspberry*, conforme figura 8.

Figura 8 – Conexão da câmera no *Raspberry*



Fonte: Os autores (2022).

Para o alerta sonoro, foi usado um *Buzzer* conectado aos pinos do GPIO do *Raspberry Pi 3* que são portas programáveis de entradas e saídas de dados. Já nos alertas visuais, foram usados LED, ilustrados nas figuras 7 e 8, supra-apresentadas. A soma dos custos dos elementos utilizados ficou

entre 200 e 300 dólares, tornando-o com um sistema com preço competitivo.

#### 3.2 Alerta

Em seguida, foi feita a verificação do tempo de processamento em diferentes tipos de *hardware*, em um computador *desktop*, com configuração mediana, o *Raspberry Pi Model B+* e, em diferentes tipos de câmeras, como uma *webcam* HD 1080, INTELBRAS e a Pi Câmera. A figura 9 mostra a representação do *frame* da câmera do sistema para a detecção da fadiga.

Figura 9 – Alerta exibido no monitor



Fonte: Os autores (2022).

Previamente programada, a mensagem será exibida no *frame*, após a detecção da fadiga; essa janela é utilizada pelo sistema para detectar a fadiga. A partir da imagem da câmera, o *frame* identificou a fadiga do rosto.

#### 3.3 Comparativo de tempo de detecção de fadiga entre *hardwares*

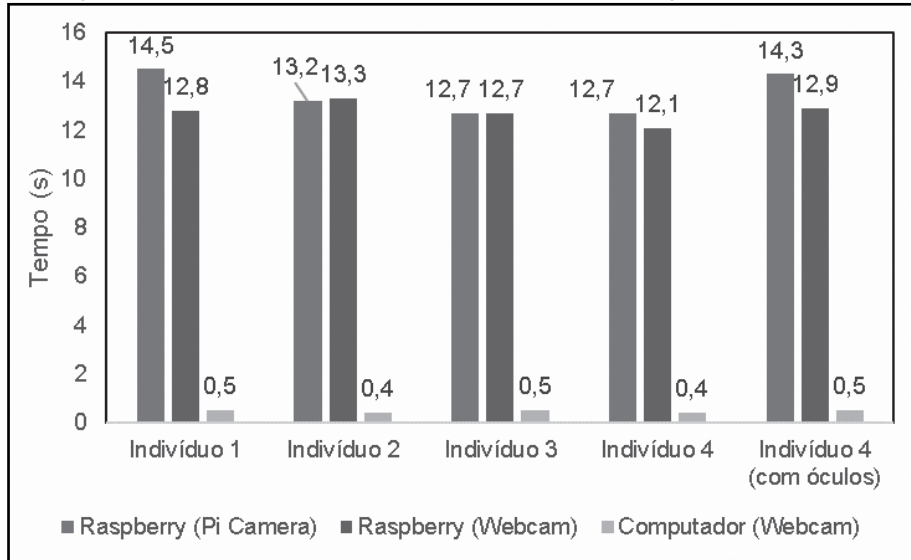
Para o experimento, foram selecionados quatro indivíduos. A distância entre o rosto e a câmera foi de 50cm, e o tempo foi cronometrado, desde o momento em que a face foi detectada até o alerta de fadiga, conforme figura 10.

O tempo de detecção da fadiga mostrou-se semelhante entre os indivíduos, até mesmo, ao utilizar algum acessório facial, como por exemplo, óculos de grau. O tempo de detecção do *Raspberry* ficou entre 12 e 13 segundos. Com a simulação da programação no computador, os resultados foram próximos a um (1) segundo. Os desvios de

padrão do *Raspberry*, utilizando-se a *Pi Camera*, o *Raspberry* com a *Webcam* e um computador de mesa que utiliza a *Webcam*, foram, respectivamente,

de: 0,868407, 0,451088 e 0,06442. Isso demonstra que os testes tiveram grande interferência, devido à capacidade do *hardware* utilizado.

Figura 10 – Comparativo de tempo de detecção de fadiga entre *hardwares*



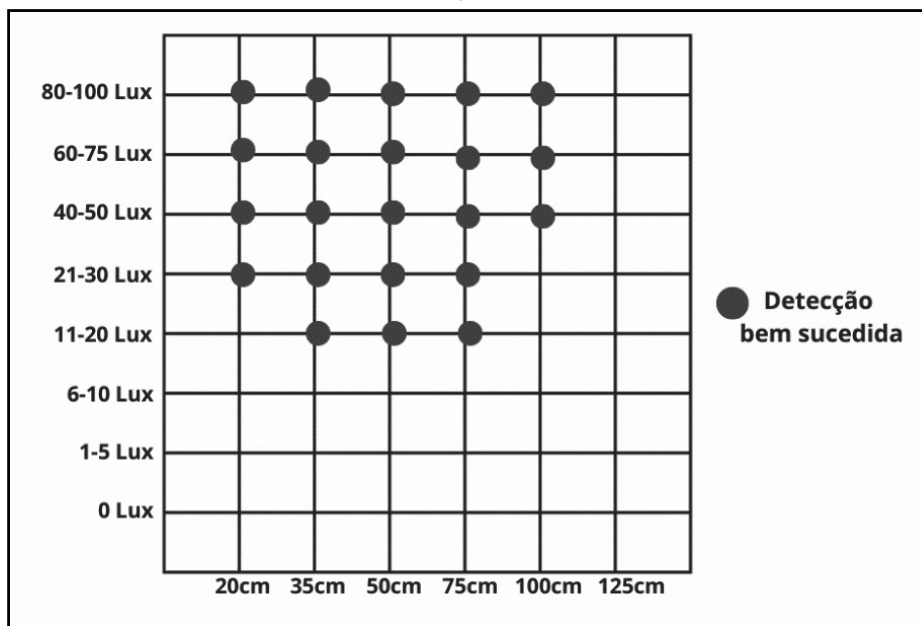
Fonte: Os autores (2022).

### 3.4 Comparativo de tempo com variação de luminosidade

Após os testes de tempo de detecção, foram feitas análises de comportamento do sensor em diferentes níveis de luminosidades e utilizou-se o *Raspberry* com a *Pi Camera*. O teste foi realizado em distâncias que

variaram de 20 a 125 centímetros entre a câmera do protótipo e o rosto. E, em luminosidades de 0 a 100 lux, utilizando uma luz de LED com níveis variáveis de intensidade luminosa e um luxímetro, para medir a intensidade de luz emitida na face do indivíduo, como mostra a figura 11, abaixo.

Figura 11 – Capacidade do sistema de detectar fadiga, em relação a diferentes distâncias e luminosidade



Fonte: Os autores (2022).



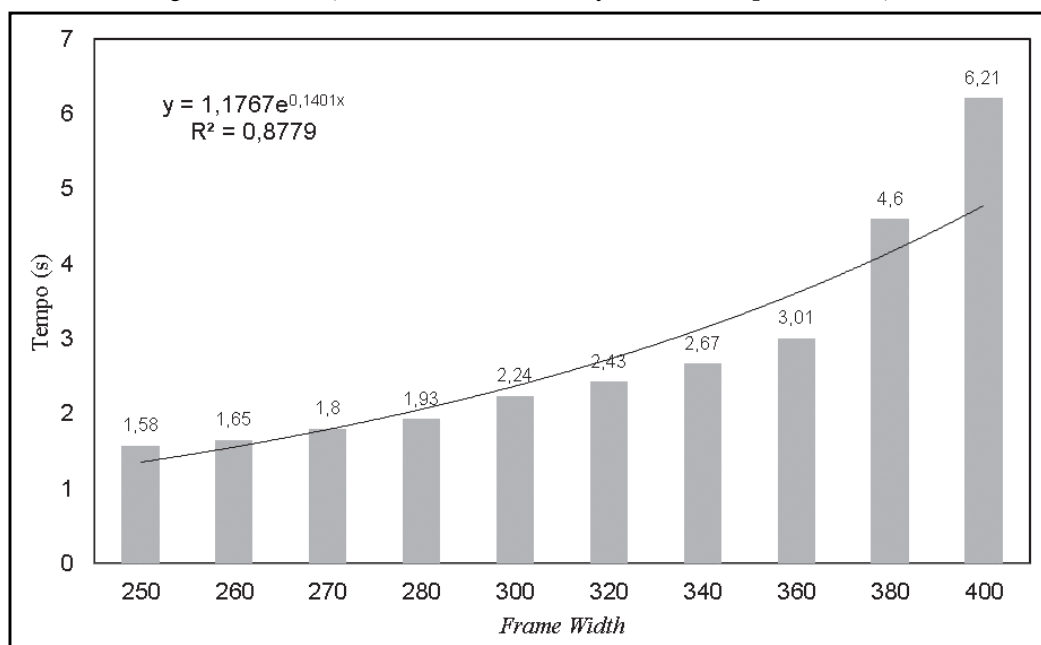
Não foi possível detectar o rosto em uma intensidade luminosa inferior a 10 lux, ao utilizar a câmera atual, por não possuir capacidade para detecção em baixo índice de luminosidade. Também foi constatado que, em distâncias acima de 125 cm, o rosto não foi identificado. Com a menor luminosidade, de 11 a 20 lux, os testes de funcionamento do sensor em distâncias de 35cm a 75cm mostraram-se satisfatórios para a detecção da fadiga.

### 3.5 Testes com variação do tamanho do *frame*

Logo após os testes de luminosidade, foram elaborados ensaios, para alterar o tamanho da janela

(*frame*), local onde a imagem da câmera e, posteriormente, do alerta apareceriam. A detecção da fadiga tem como base a imagem apresentada nessa janela, os ensaios constataram que, quando alterado o tamanho do *frame*, a distância e o tempo de detecção sofrem variações, podendo assim diminuir o tempo e, por consequência, diminuir a distância de detecção. A figura 12 mostra a relação entre o tamanho do *frame* e o tempo que o sensor leva para realizar a detecção da fadiga, quando utilizar o *Raspberry* com a *Pi Camera*, com distância de 50 cm e iluminação de 40-50 lux.

Figura 12 – Relação entre o tamanho do *frame* e o tempo de detecção



Fonte: Os autores (2022).

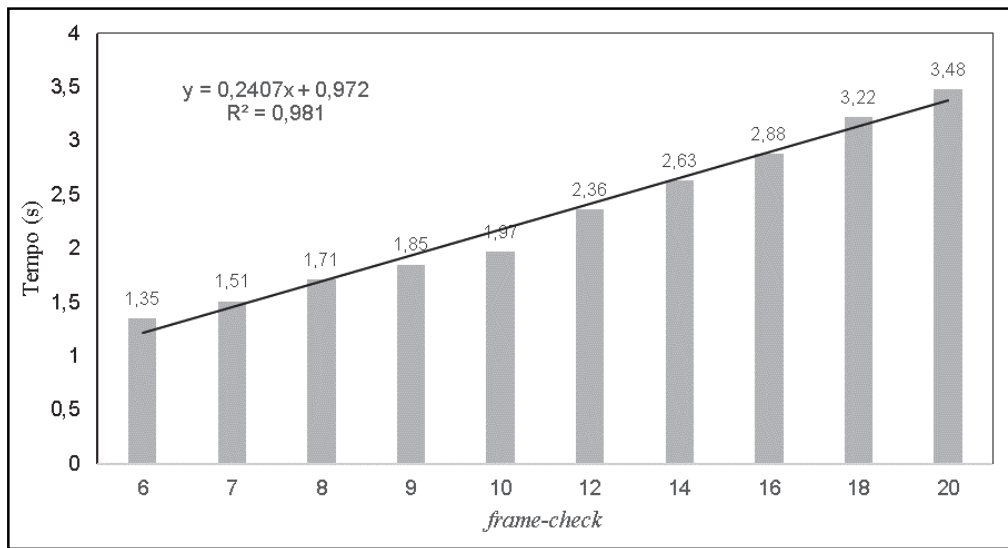
Os resultados mostram que o tempo de detecção da fadiga ficou abaixo de 2 s, para condições abaixo de 280 pixels. Acima dessa resolução, os testes provaram maior atraso na detecção da fadiga. Os testes demonstraram um comportamento exponencial, com a métrica  $R^2$  de 0,87, o que aponta para uma boa relação estatística de proporcionalidade entre a variação do *Frame width* e do tempo de resposta. Partindo do princípio dos testes de alteração do *frame*, para a melhora no tempo de detecção de fadiga, foram realizados testes de variação do *Frame check*, um comando que indica, a partir de quantos quadros, após o fechamento dos olhos, a fadiga é detectada. A figura 13 mostra a relação entre o *Frame check* e o tempo de detecção da fadiga.

O melhor resultado ocorreu depois de seis (6)

*frames* detectados. Essa detecção ocorreu com o tempo próximo de um (1) segundo. Os testes demonstraram um comportamento linear, com  $R^2$  de 0,98 é possível notar uma ótima relação estatística de proporcionalidade entre a variação do *Frame check* e do tempo de resposta. Esses resultados validam que o sensor de fadiga pode se tornar um grande aliado na prevenção de acidentes. O desvio de atenção do motorista, por dois (2) segundos ou mais, é responsável por cerca de 72% dos tombamentos e outros tipos de acidentes de trânsito, sendo o uso de aplicativos de celular e cansaço as principais causas (TRIMBLE, 2022).

O tempo de detecção do *Raspberry* inicialmente se deu entre 12 e 13 segundos, figura dez (10).

Figura 13 – Relação entre o tempo de detecção e o *frame check*



Fonte: Os autores (2022).

Com a simulação da programação no computador, os resultados fora próximos a um (1) segundo. Não foi possível a detecção do rosto em uma intensidade luminosa inferior a dez (10) lux, utilizando a câmera atual, por não possui capacidade para detecção em baixo índice de luminosidade; além disso, foi constatado que, em distâncias acima de 125 cm, o rosto não foi identificado. Com a menor luminosidade, de 11 a 20 lux, os testes de funcionamento do sensor em distâncias de 35cm a 75cm se mostraram satisfatórios para a detecção da fadiga.

A referência para a distância entre o rosto do motorista e o para-brisa de um veículo varia, dependendo da fonte consultada e do modelo do veículo. No entanto, em geral, essa distância costuma ser de cerca de 50 a 70 centímetros. Essa informação pode ser encontrada em manuais de instrução de caminhões, artigos, publicações sobre segurança no trânsito e em sites de

fabricantes de veículos. É importante lembrar que a posição do banco do motorista e a altura do motorista podem afetar essa distância, portanto, é recomendável que o motorista ajuste o banco corretamente para garantir uma boa visibilidade e segurança ao dirigir.

Considerando que o protótipo poderia ter a câmera fixada no para-brisa é possível basear-se nesses parâmetros para otimizar os resultados e relacioná-los ao tempo de detecção de fadiga. Sendo assim, o quadro 1 mostra as áreas de detecção, com base na distância entre o motorista e a câmera, assimilando-se o funcionamento do equipamento com a faixa de distância em que o motorista estaria. Com o intuito de selecionar as melhores condições de operação do protótipo, as áreas de operação do sistema foram classificadas em: fadiga não detectada, fadiga detectada fora da área de interesse e fadiga detectada dentro da área de interesse.

Quadro 1 – Capacidade do protótipo

Luminosidade	Distância					
	20 cm	35 cm	50 cm	75 cm	100 cm	125 cm
0 lux	Fadiga não detectada					
1-5 lux	Fadiga não detectada					
6-10 lux	Fadiga não detectada					
11-20 lux	Fadiga detectada fora da área de interesse					
21-30 lux	Fadiga detectada fora da área de interesse					
40-50 lux	Fadiga detectada dentro da área de interesse					
60-75 lux	Fadiga detectada dentro da área de interesse					
80-100 lux	Fadiga detectada dentro da área de interesse					

Fonte: Os autores (2023).

Como mostrado no quadro 1, foi possível determinar as regiões de operação do protótipo, com base em sua eficácia. As melhores condições de funcionamento se evidenciaram em distâncias entre 50 e 75 cm e iluminação de 11-100 lux. Atrelado a isso, como pode ser visto na figura 12, o tempo de detecção da fadiga obtido, quando variado o Frame width, ficou abaixo de dois (2) segundos para condições abaixo de 280 pixels. Com a variação do Frame check, figura 13, o melhor resultado ocorreu com o tempo próximo de um (1) segundo, com o Frame check setado em 6 (seis) quadros.

A melhor região de operação, com tempo de detecção próximo a um (1) segundo, quadro 1, mostra que a capacidade do protótipo proposto é similar aos equipamentos comercializados e homologados por empresas do ramo de segurança no trânsito, o que torna os resultados obtidos mais consistentes. Destaca-se que a detecção da fadiga, fora da área de interesse, poderia garantir maior confiabilidade ao sistema proposto, isso porque há variáveis como: a altura do motorista e a regulagem dos bancos que podem interferir na posição das pessoas em um veículo.

O desvio padrão dos resultados obtidos com o Raspberry, ao utilizar a Pi Camera, o Raspberry com a Webcam e um computador de mesa, utilizando a Webcam foram, respectivamente, de: 0,868407, 0,451088 e 0,06442. Isso demonstra que os resultados tiveram grande influência, devido à capacidade de processamento do hardware utilizado. Os testes, com variação do Frame width, demonstraram um comportamento exponencial, com um  $R^2$  de 0,87, o que aponta para uma boa relação estatística de proporcionalidade entre a variação do Frame e do tempo de resposta. Os testes com variação de Frame check demonstraram um comportamento linear, com um  $R^2$  de 0,98, estabelecendo uma ótima relação estatística de proporcionalidade entre a variação do Frame check e o tempo de resposta.

Algumas empresas do ramo, como a Trimble, relatam que o uso de tecnologia semelhante reduz em até 86% dos potenciais acidentes com sensores de fadiga (TRIMBLE, 2022). Existem empresas com sensores de fadiga implementados em 100% da frota de caminhões, havendo uma diminuição drástica no número de acidentes resultante de fadiga.

#### 4 Considerações finais

O protótipo que utilizou *Raspberry* e a *Pi Camera* se mostrou funcional, e os objetivos propostos neste trabalho foram alcançados. Foi possível realizar a

detecção da fadiga e o acionamento dos alertas, com uma capacidade semelhante a ferramentas homologadas e comercializadas no mercado, que trabalham com sistemas de fadiga com tempo de detecção inferiores a dois (2) segundos.

Os melhores resultados para a detecção da fadiga, obtidos com o protótipo, ficaram abaixo de dois (2) segundos, com variação do tamanho e da quantidade de *frames*. Os testes que utilizaram *Raspberry* e a *Pi Camera*, com variação do *Frame width* e *Frame check*, apresentaram um comportamento de proporcionalidade em relação ao tempo de resposta. O  $R^2$  dos resultados desses testes variou de 0,87 a 0,98, o que validou a relação de proporcionalidade entre a variação dos *frames* e o tempo de resposta obtido.

O desvio de padrão dos resultados dos testes de tempo de resposta dos componentes, em relação aos indivíduos, teve uma variação considerável. O desvio padrão do tempo de resposta, obtido com o *Raspberry* e a *Pi Camera*, o *Raspberry* e a *Webcam* e um computador de mesa com a *Webcam*, foram de: 0,868407, 0,451088 e 0,06442, respectivamente. Isso demonstra que o tempo de detecção e sua variação dependem principalmente da capacidade do *hardware* utilizado.

É importante destacar que os resultados obtidos com este trabalho foram positivos, em relação à efetividade do sistema. Além disso, pode-se afirmar que o objetivo de se obter um custo competitivo com uso de tecnologia alternativa se mostrou efetivo.

Como sugestão para futuros trabalhos, um possível *upgrade* para o sistema, seria a implementação de uma câmera com sensor infravermelho, visto que capacitaria o protótipo a detectar faces, mesmo com baixa luminosidade. Outra proposta seria a alteração do *hardware*, a partir da produção de uma placa com maior capacidade de processamento e configurada especificamente para o funcionamento do sistema, podendo reduzir exponencialmente o valor do produto, ao otimizar e melhorar o desempenho do protótipo.

#### Referências

- CONFEDERAÇÃO NACIONAL DO TRANSPORTE. W. 2022. Disponível em: <https://www.cnt.org.br/painel-acidente>. Acesso em: 12 jun. 2022.
- FERREIRA, J. M. S.; MARQUES, R. C. S.; BRAZ JUNIOR, G. **Detecção de sonolência ao volante utilizando Facial Landmarks e Eye Aspect Ratio**. 2018. Disponível em: <https://docplayer.com.br/71582082-Deteccao-de-sonolencia-ao-volante-utilizando-facial-landmarks-e-eye-aspect-ratio.html>. Acesso em: 31 maio 2022.

KUWAHARA, A. *et al.* Eye fatigue estimation using blink detection based on *Eye Aspect Ratio Mapping* (EARM). **Cognitive Robotics**, v. 2, p. 50-59, 2022. DOI: 10.1016/j.cogr.2022.01.003.

PAIXÃO, E. P. **Aplicação do algoritmo de Viola-Jones na detecção de objeto**. 49 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica) – Universidade Regional do Noroeste do Estado do Rio Grande do Sul, Ijuí, 2018. Disponível em: <https://bibliodigital.unijui.edu.br:8443/xmlui/bitstream/handle/123456789/5832/Elisiane%20Pelke%20Paixão.pdf?sequence=1&isAllowed=y#:~:text=Por%20meio%20do%20algoritmo%20Viola,vetores%20de%20identificaç~ao>. Acesso em: 30 maio 2022.

PARANHOS, V. K. **Sistema de detecção de sonolência**. 2019. 67 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica) – Faculdade de Engenharia e Arquitetura, Universidade de Passo Fundo, Passo Fundo 2019. Disponível em: <http://repositorio.upf.br/bitstream/riupf/1709/1/PF2019Vinicius%20Kerber%20Paranhos.pdf>. Acesso em: 11 maio 2022.

ROSEBROCK, A. **Facial landmarks with DLIB, OpenCV, and Python**. 2017. Disponível em: <https://pyimagesearch.com/2017/04/03/facial-landmarks-DLIB-OpenCV-Python/>. Acesso em: 31 maio 2022.

SAGONAS, C. *et al.* 300 Faces in-the-wild challenge: the first facial landmark localization challenge. *In: IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION*, 2013, Sidney. **Proceedings** [...]. Piscataway: IEEE, 2013. p. 397-403. Disponível em: <https://nottingham-repository.worktribe.com/preview/1000750/tzimiroICCVW13.pdf>. Acesso em: 15 jul. 2022.

SOUKUPOVÁ, T.; CECH, J. Real time eye blink detection using facial landmarks. *In: COMPUTER VISION WINTER WORKSHOP*, 21., 2016, Slovenia. **Proceedings** [...]. Slovenia: The Slovenian Pattern Recognition Society, 2016. p. 1-8.

SUAREZ, A. J. **Real-time drowsiness detection OpenCV Python with source code**. 2021. Disponível em: <https://itsourcecode.com/free-projects/Python-projects/real-time-drowsiness-detection-OpenCV-Python-with-source-code/>. Acesso em: 20 abr. 2022.

TIEMI, A. **Sensor de fadiga: como funciona?** 2021. Disponível em: <https://tl.trimble.com/blog/sensor-de-fadiga/#:~:text=O%20detector%20ou%20sensor%20de,nível%20de%20atenção%20do%20condutor>. Acesso em: 25 jun. 2022.

TRIMBLE. **Empresa global que desenvolve tecnologias disruptivas no setor de logística e transporte há mais de 40 anos, conectam embarcadores e transportadores a soluções tecnológicas para maximizar a segurança, produtividade e rentabilidade**. 2022. Disponível em: <https://tl.trimble.com/sobre-nos/>. Acesso em: 25 jun. 2022

WORLD HEALTH ORGANIZATION. **Global status report on road safety 2018**. Geneva, 2018. 403 p. Disponível em: <https://www.unifor.br/documentos/929808/930330/9789241565684-eng.pdf/36966e12-d0f4-725a-2f4a-a7ed90a54855>. Acesso em: 20 maio 2022.

WU, Y.; JI, Q. Facial landmark detection: a literature survey. **International Journal of Computer Vision**, v. 127, n. 2, p. 115-142. 2018. DOI: 10.1007/s11263-018-1097-z. Disponível em: <https://link.springer.com/article/10.1007/s11263-018-1097-z>. Acesso em: 31 maio 2022.