



**Centro Federal de Educação Tecnológica de Minas Gerais  
Unidade Araxá**

Tiago Bruno Dias

**Análise de um protótipo de um sistema embarcado para monitoramento do  
nível de atenção dos motoristas**

Araxá-MG

2020

Tiago Bruno Dias

**Análise de um protótipo de um sistema embarcado para monitoramento do  
nível de atenção dos motoristas**

Monografia apresentada ao Curso de Engenharia de Automação Industrial, do Centro Federal de Educação Tecnológica de Minas Gerais - Unidade Araxá, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Automação Industrial.

Orientador: Prof. Dr. Kléber Lopes Fontoura  
Coorientador: Prof. Me. Sérgio Luiz da Silva Pithan

Araxá-MG

2020



ATA Nº 18/2020 - DFGAX (11.57.03)

Nº do Protocolo: 23062.033912/2020-91

Araxá-MG, 10 de dezembro de 2020.



**SERVIÇO PÚBLICO FEDERAL**  
**MINISTÉRIO DA EDUCAÇÃO**  
**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS**  
**COORDENAÇÃO DO CURSO DE ENGENHARIA DE AUTOMAÇÃO INDUSTRIAL / ARAXÁ**  
**TRABALHO DE CONCLUSÃO DE CURSO - TCC - ATA DE DEFESA**

ATA DA DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO DE ENGENHARIA DE AUTOMAÇÃO INDUSTRIAL  
do aluno **TIAGO BRUNO DIAS**

Às quatorze horas do oitavo de dezembro de dois mil e vinte, reuniu-se, pelo Microsoft Teams, na turma "Trabalho de Conclusão de Curso", do Centro Federal de Educação Tecnológica de Minas Gerais - CEFET-MG/ Unidade Araxá, a Comissão Examinadora de Trabalho de Conclusão de Curso para julgar, em exame final, o trabalho intitulado "Análise de um protótipo de um sistema embarcado para monitoramento do nível de atenção dos motoristas", como requisito parcial para a obtenção do grau de Bacharel em Engenharia de Automação Industrial. Abrindo a sessão, o Presidente da Comissão, Prof. Dr. Kléber Lopes Fontoura, após dar a conhecer aos presentes o teor das Normas Regulamentares do Trabalho Final, concedeu a palavra ao candidato, Tiago Bruno Dias, para a exposição de seu trabalho. Após a apresentação, seguiu-se a arguição pelos examinadores, com a respectiva defesa do candidato. Ultimada a arguição, a Comissão se reuniu, sem a presença do candidato e do público, para julgamento e expedição do resultado final. Após a reunião da Comissão Examinadora, o candidato foi considerado: **APROVADO**, obtendo nota final de: 90/100 (noventa pontos). O resultado final foi comunicado publicamente ao candidato pelo Presidente da Comissão. O aluno, abaixo assinado, declara que o trabalho ora identificado é da sua autoria material e intelectual, excetuando-se eventuais elementos, tais como passagens de texto, citações, figuras e datas, desde que devidamente identificadas a fonte original. Declara ainda, neste âmbito, não violar direitos de terceiros. Nada mais havendo a tratar, o Presidente

encerrou os trabalhos. A Profa. Dra. Érica Araújo, responsável pela disciplina "Trabalho de Conclusão de Curso II", lavrou a presente ATA, que, após lida e aprovada, será assinada por todos os membros participantes da Comissão Examinadora. Araxá, oito de dezembro de dois mil e vinte.

*(Assinado digitalmente em 11/12/2020 08:46)*

ERICA DANIELA DE ARAUJO  
PROFESSOR ENS BASICO TECN TECNOLOGICO  
DFGAX (11.57.03)  
Matrícula: 1019759

*(Assinado digitalmente em 10/12/2020 22:58)*

KLEBER LOPES FONTOURA  
PROFESSOR ENS BASICO TECN TECNOLOGICO  
DELMAX (11.57.05)  
Matrícula: 2330904

*(Assinado digitalmente em 10/12/2020 10:15)*

MARCO ANTONIO DURCO  
PROFESSOR ENS BASICO TECN TECNOLOGICO  
DELMAX (11.57.05)  
Matrícula: 1045745

*(Assinado digitalmente em 10/12/2020 09:07)*

SERGIO LUIZ DA SILVA PITHAN  
COORDENADOR - TITULAR  
CHEFE DE UNIDADE  
CEAIX (11.51.16)  
Matrícula: 2121424

*(Não Assinado)*

Tiago Bruno Dias  
DISCENTE  
Matrícula: 201514550458

Para verificar a autenticidade deste documento entre em <https://sig.cefetmg.br/public/documentos/index.jsp> informando seu número: **18**, ano: **2020**, tipo: **ATA**, data de emissão: **10/12/2020** e o código de verificação: **fedd9a4694**



SERVIÇO PÚBLICO FEDERAL  
MINISTÉRIO DA EDUCAÇÃO  
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS  
COORDENAÇÃO DO CURSO DE ENGENHARIA DE AUTOMAÇÃO INDUSTRIAL / ARAXÁ

TRABALHO DE CONCLUSÃO DE CURSO – TCC – ATA DE DEFESA

ATA DA DEFESA DE TRABALHO DE CONCLUSÃO DE CURSO DE ENGENHARIA DE  
AUTOMAÇÃO INDUSTRIAL do aluno **TIAGO BRUNO DIAS**

Às quatorze horas do oito de dezembro de dois mil e vinte, reuniu-se, pelo Microsoft Teams, na turma "Trabalho de Conclusão de Curso", do Centro Federal de Educação Tecnológica de Minas Gerais - CEFET-MG/ Unidade Araxá, a Comissão Examinadora de Trabalho de Conclusão de Curso para julgar, em exame final, o trabalho intitulado "Análise de um protótipo de um sistema embarcado para monitoramento do nível de atenção dos motoristas", como requisito parcial para a obtenção do grau de Bacharel em Engenharia de Automação Industrial. Abrindo a sessão, o Presidente da Comissão, Prof. Dr. Kléber Lopes Fontoura, após dar a conhecer aos presentes o teor das Normas Regulamentares do Trabalho Final, concedeu a palavra ao candidato, Tiago Bruno Dias, para a exposição de seu trabalho. Após a apresentação, seguiu-se a arguição pelos examinadores, com a respectiva defesa do candidato. Ultimada a arguição, a Comissão se reuniu, sem a presença do candidato e do público, para julgamento e expedição do resultado final. Após a reunião da Comissão Examinadora, o candidato foi considerado: **APROVADO**, obtendo nota final de: 90/100 (noventa pontos). O resultado final foi comunicado publicamente ao candidato pelo Presidente da Comissão. O aluno, abaixo assinado, declara que o trabalho ora identificado é da sua autoria material e intelectual, excetuando-se eventuais elementos, tais como passagens de texto, citações, figuras e datas, desde que devidamente identificada a fonte original. Declara ainda, neste âmbito, não violar direitos de terceiros. Nada mais havendo a tratar, o Presidente encerrou os trabalhos. A Profa. Dra. Érica Araújo, responsável pela disciplina "Trabalho de Conclusão de Curso II", lavrou a presente ATA, que, após lida e aprovada, será assinada por todos os membros participantes da Comissão Examinadora. Araxá, oito de dezembro de dois mil e vinte.

\_\_\_\_\_  
Presidente e Orientador: Prof. Dr. Kléber Lopes Fontoura

\_\_\_\_\_  
Coorientador: Prof. Dr. Sérgio Luiz da Silva Pithan

\_\_\_\_\_  
Membro Titular Interno: Prof. Dr. Marco Antônio Durço

Maicon Fernando Dias Torres  
Membro Titular Interno: Prof. Me. Maicon Fernando Dias Torres

\_\_\_\_\_  
Professora da Disciplina TCCII: Profa. Dra. Érica Daniela de Araújo

Tiago Bruno Dias  
Aluno: Tiago Bruno Dias

Aos meus pais e irmãos que sempre estiveram  
ao meu lado durante toda a minha vida

## **AGRADECIMENTOS**

À Deus pelo dom da vida e saúde. Aos meus pais, Carlos Dias da Silva e Dulcelena de Fátima Rosa Dias, aos meus irmãos, Pablo Joanes Dias, Gabriel Thalís Dias e Carlos Dias da Silva Júnior, pelo incentivo e ajuda no qual contribuiu para a realização desse trabalho. Os professores Kléber Lopes Fontoura, Sérgio Luiz da Silva Pithan, Marco Antônio Durço e Maicon Fernando Dias Torres por aceitarem o convite para participar da banca de defesa desta monografia.

“Ama e faz o que quiseres. Se calares, calarás com amor; se gritares, gritarás com amor; se corrigires, corrigirás com amor; se perdoares, perdoarás com amor. Se tiveres o amor enraizado em ti, nenhuma coisa senão o amor serão os teus frutos. “

Santo Agostinho

## RESUMO

Uma parcela considerável dos acidentes de trânsito no Brasil está relacionada à sonolência e ao cansaço ao volante de veículos. Desse modo, surgem sensores de fadiga instalados, para alertar o motorista dessas ocorrências. Utilizou-se a placa Raspberry Pi 3 como computador embarcado para o processamento da imagem e uma câmera com LEDs infravermelho para o protótipo funcionar durante a noite. Testou-se algumas metodologias utilizado por diversos autores para a detecção do rosto e olhos como o de Viola e Jones (2001) e Khunpisuth *et al.* (2016). Para implementar o algoritmo foi utilizado principalmente a biblioteca OpenCV de processamento de imagem. E por fim foi aplicado um algoritmo que analisa o comportamento dos olhos, se estão fechados ou não, com isso o protótipo alerta em caso de detecção de sonolência.

**Palavras-chave: Sensor de fadiga. Detecção de sonolência.**

## **ABSTRACT**

A considerable portion of traffic accidents in Brazil are related to drowsiness and tiredness behind the wheel of vehicles. In this way, fatigue sensors are installed to alert the driver of these occurrences. The Raspberry Pi 3 board was used as an embedded computer for image processing and a camera with infrared LEDs for the prototype to work at night. Some methodologies used by several authors for the detection of the face and eyes were tested, such as that by Viola and Jones (2001) and Khunpisuth et al. (2016). To implement the algorithm, the OpenCV image processing library was used. Finally, an algorithm was applied to analyze the behavior of the eyes, whether they are closed or not, with this the prototype alerts in case of detection of drowsiness.

**Keywords: Fatigue sensor. Detection of drowsiness.**

## LISTA DE ILUSTRAÇÕES

Figura 1 – Sensor de fadiga i-VUE D10.....	13
Figura 2 – Sensor de fadiga Vsafe.....	14
Figura 3 – Sensor de fadiga Marpress.....	15
Figura 4 - Posicionamento dos componentes do sistema de monitoramento embarcado no painel do veículo.....	17
Figura 5 - Placa de desenvolvimento SBC2440-I.....	18
Figura 6 - Placa Raspberry Pi 3 com câmera já conectada.....	20
Figura 7 – Característica de linha aplicada ao rosto.....	22
Figura 8 - Características analisadas de um objeto.....	22
Figura 9 - Resultados obtido pelo algoritmo de alinhamento de face.....	23
Figura 10 - Testes do algoritmo de Viola e Jones (2001) com rostos na posição frontal.....	25
Figura 11 - Testes com rostos na posição frontal.....	26
Figura 12 - Utilização do sistema operacional usando o VNC.....	29
Figura 13 - Configuração da câmera no Raspberry Pi.....	30
Figura 14 - Teste aquisição de imagem.....	31
Figura 15 - Teste aquisição de imagem com Thread.....	31
Figura 16 - Teste com classificador frontal face.....	32
Figura 17 - Teste com classificador frontal face + profile face.....	33
Figura 18 - Resultado do teste com o algoritmo de Kazemi e Sullivan (2014).....	34
Figura 19 - Resultado do teste com o algoritmo de Kazemi e Sullivan (2014).....	34
Figura 20 – Regulador de tensão LM2596.....	35
Figura 21 - O protótipo.....	36
Figura 22 - Protótipo instalado.....	37

## LISTA DE QUADROS

Quadro 1 - Comparação entre o algoritmo de Viola e Jones (2001) com o método aplicado por Khunpisuth <i>et al.</i> (2016).....	26
Quadro 2 – Características dos conversores AMS1117-5.0 e LM2596.....	35
Quadro 3 – Custos do protótipo.....	38
Quadro 4 – Resultado do teste de piscada diurno.....	38
Quadro 5 – Resultado do teste de piscada noturno.....	39
Quadro 6 - Taxa de precisão da face e detecção do olho.....	39
Quadro 7 – Resultado do teste de sonolência diurno.....	40
Quadro 8 – Resultado do teste de sonolência noturno.....	40

## LISTA DE ABREVIATURAS E SIGLAS

LED	Light-emitting diode
USB	Universal Serial Bus
GPS	Global positioning system
ARM	Advanced RISC Machine
RAM	Random-access memory
OPENCV	Open Source Computer Vision Library
VNC	Virtual Network Computing
FPS	Frames Per Second
EAR	Eye Aspect Ratio
A	Ampere
V	Volt
ABS	Acrylonitrilo butadieno estireno

## Sumário

<b>1 INTRODUÇÃO.....</b>	<b>13</b>
<b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>17</b>
2.1 Processamento embarcado de um sensor de fadiga.....	17
2.2 Processamento de imagem.....	20
<b>3 METODOLOGIA.....</b>	<b>26</b>
3.1 Instalação das ferramentas de desenvolvimento.....	28
3.2 Aquisição da imagem.....	30
3.3 Detecções de sonolência e de fadiga.....	32
3.4 Montagem do equipamento.....	35
3.5 Implementação do sistema de monitoramento embarcado em um veículo de teste.....	36
<b>4 ANÁLISE DOS DADOS E DISCUSSÕES.....</b>	<b>37</b>
4.1 Custos de hardware do protótipo.....	37
4.2 O protótipo.....	38
<b>5 CONSIDERAÇÕES FINAIS.....</b>	<b>41</b>
<b>REFERÊNCIAS.....</b>	<b>42</b>

## 1 INTRODUÇÃO

Novas tecnologias surgem a cada dia com o propósito de melhorar a qualidade de vida dos seres humanos. Dentre essas, o presente trabalho destaca as que envolvem transformação do trânsito de veículos e têm se tornadas aliadas na prevenção e redução do número de mortes envolvendo acidentes automotivos no mundo todo. No Brasil, segundo Batista (2019),

**[...] cerca de 42% dos acidentes de trânsito estão relacionados à sonolência. Essa é ainda uma das principais causas de mortes nas rodovias.** Os dados são da Associação Brasileira de Medicina de Tráfego (ABRAMET), com o alerta de que a falta de descanso compromete a direção em nível semelhante ao provocado pela ingestão de bebida alcoólica. Pesquisa realizada pela Academia Brasileira de Neurologia, em conjunto com o Conselho Regional de Medicina e a ABRAMET, para a campanha “Não dê carona ao sono”, entrevistou quase 500 motoristas e concluiu que **mais de 20% deles costumam dirigir com sono.** Além disso, segundo o diretor de comunicação da ABRAMET, Dr. Dirceu Rodrigues Alves Júnior, **18% dos acidentes são ocasionados pela fadiga dos motoristas. Juntos, sono e cansaço representam alarmantes 60% dos acidentes de trânsito do país** (BATISTA, 2019, grifo negrito nosso).

Nesse contexto, empresas de tecnologia têm investido, cada vez mais, em dispositivos capazes de alertar aos motoristas diante de uma situação de perigo visando reduzir o número de acidentes e, conseqüentemente, mortes. Entre esses dispositivos, há aqueles que cumprem a função de alertar ao motorista em caso de desatenção de tal modo a protegê-lo de um possível acidente. Esses dispositivos utilizam diferentes sensores como câmeras, utilizando sistemas de processamento de imagem, ou de batimento cardíaco e são denominados sensores de fadiga.

Um dos modelos de sensor de fadiga disponíveis no mercado é o i-VUE D10, da fabricante TreeEarth, ilustrado na Figura 1.

Figura 1 – Sensor de fadiga i-VUE D10.



Fonte: TreeEarth, 2018

Esse modelo apresenta diodos emissores de luz (*light-emitting diode, ou LED*) infravermelho para utilização à noite, alto-falante embutido, para alertar ao motorista da ocorrência, e também remove o reflexo irregular da imagem ocasionado pelo sol usando filtros óticos especiais. Suas funções básicas envolvem o rastreamento dos olhos, sonolência, distração e desvio de face. Com a função de desvio de face, o sensor de fadiga emite um alerta caso não encontre nenhum rosto. Segundo o fabricante, a porcentagem do desempenho de reconhecimento destas funções é de 97.5%. Os eventos são registrados contendo a localização de latitude e longitude fornecidas pelo GPS. O sensor fornece também, saídas de vídeos das imagens da câmera por USB e Tv-out. Esse sensor tem um custo estimado de R\$2500.

A empresa Veltec, apresenta como solução para detecção de fadiga o Vsafe, representado Figura 2. Esse dispositivo, como afirmado pelo fabricante, é capaz de detectar fadiga do motorista mesmo se ele estiver utilizando óculos de grau ou de sol.

Figura 2 – Sensor de fadiga Vsafe.



Fonte: Veltec, 2018

Um de seus diferenciais é contar com uma câmera que analisa o motorista em seu veículo e outra voltada para a pista. Com a utilização da câmera voltada para a pista, é emitido um alerta caso haja algum risco de colisão com o veículo da frente. Além disso, também conta com recurso de registro das ocorrências em formato de vídeo e efetua a transferência das ocorrências por meio de rede 3G. O custo desse sensor não foi informado pelo fabricante, nos contatos realizados pelo autor.

Outro sensor de fadiga é o da fabricante Marpress, representado na Figura 3. Esse sensor também conta com uma câmera com um conjunto de LEDs infravermelhos, juntamente com o *software* que efetua a análise e o monitoramento do grau de fadiga dos motoristas. Em

relação ao custo desse sensor, foi entrado em contato com o fabricante, porém o mesmo não disponibilizou seu custo.

Figura 3 – Sensor de fadiga Marpress.



Fonte: Marpress, 2018

Diante dos modelos apresentados, suas especificidades, este trabalho tem como objetivo desenvolver um sistema embarcado, com foco na confiabilidade, para monitoramento da sonolência e da fadiga dos condutores de veículos. Para tanto, estabeleceu-se como objetivos específicos desta pesquisa:

- Determinar a câmera e o computador embarcados mais adequados aos propósitos desta pesquisa;
- Comparar os dados obtidos nas rodagens-teste do sistema de monitoramento embarcado proposto com outros métodos elaborados por outros autores, como Khunpisuth *et al.* (2016);
- Analisar o desempenho e a confiabilidade de todo o sistema;

O desenvolvimento desta investigação se justifica na medida em que um sensor com baixo custo, alta confiabilidade e independente de empresas, irá contribuir com o desenvolvimento de novos produtos e poderá, por consequência, contribuir na redução da quantidade de acidentes no trânsito das cidades e rodovias. Com os fundamentos e testes abordados nesta pesquisa, pode-se tornar um avanço em uma tecnologia mais acessível em questões financeiras, porém sem perder a confiabilidade da detecção da sonolência.

Em aspectos de funcionalidades, o sensor irá monitorar o comportamento dos olhos, disparando um alarme sonoro que alerta ao motorista no caso de apresentar sinais de sonolência ao volante.

Com o propósito de realizar a investigação aqui apresentada, este trabalho está estruturado em três capítulos, além desta “Introdução” e de “Considerações Finais”. No segundo capítulo, apresenta-se uma revisão bibliográfica sobre os principais conceitos implicados nessa proposta de pesquisa, entre eles, cita-se os presentes trabalhos de

Khunpisuth *et al.* (2016) e Kazemi e Sullivan (2014). No terceiro capítulo, detalha-se a metodologia utilizada neste estudo de caso, a qual utiliza o algoritmo de Kazemi e Sullivan (2014) para reconhecimento do olho e posteriormente a aplicação do algoritmo de detecção de sonolência. No quarto capítulo, analisam-se os dados coletados durante o evento de teste com o protótipo instalado dentro de um veículo e discute-se os resultados obtidos.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, retomando trabalhos pertinentes da literatura, detalham-se os principais conceitos mobilizados nesta pesquisa, em especial, sobre: (i) a constituição e o funcionamento dos sistemas embarcados; (ii) o processamento de imagens nesses sistemas e (iii) o sistema embarcado e o processamento de imagens proposto.

### 2.1 Processamento embarcado de um sensor de fadiga

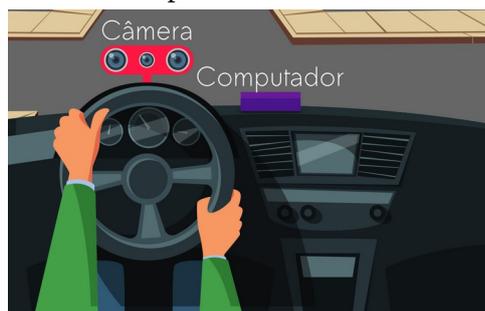
Um sistema é considerado embarcado, segundo Chase (2007),

[...] quando este é dedicado a uma única tarefa e interage continuamente com o ambiente a sua volta por meio de sensores e atuadores. [...] A denominação “embarcado” (do inglês *Embedded Systems*) vem do fato de que estes sistemas são projetados geralmente para serem independentes de uma fonte de energia fixa como uma tomada ou gerador. As principais características de classificação deste sistema são a sua capacidade computacional e a sua independência de operação. [...] Todo sistema embarcado é composto por uma unidade de processamento, que é um circuito integrado, fixado a uma placa de circuito impresso (CHASE, 2007, p. 3).

Ante a essa definição, o desenvolvimento de um sensor de fadiga utilizando esse sistema possui como vantagens a independência de operação e a interação contínua com o ambiente, já que o mesmo não necessita de outros recursos para conseguir efetuar a detecção de sonolência. E por este motivo o sensor de fadiga é um sistema embarcado.

No caso do processamento embarcado de um sensor de fadiga, tendo por base os modelos presentes no mercado, em geral esse sistema é composto por uma câmera instalada atrás do volante e um computador embarcado para processamento, conforme ilustrado na Figura 4.

Figura 4 - Posicionamento dos componentes do sistema de monitoramento embarcado no painel do veículo.



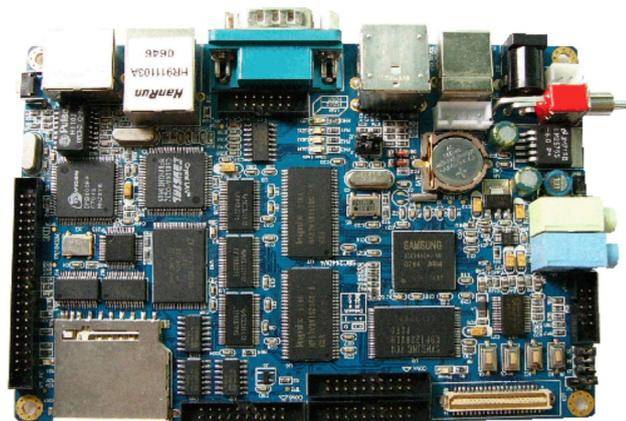
Fonte: Elaborada pelo autor.

A câmera tem por objetivo capturar a imagem do rosto do condutor do veículo e enviar essa imagem para o computador embarcado, responsável por aplicar algoritmos interpretativos sobre a imagem, a fim de identificar se o condutor do veículo está com sonolência. Quando os algoritmos interpretam e identificam a imagem como indicativa de sonolência, o sistema emite um alerta sonoro ao motorista e registra essa ocorrência em memória interna que pode, posteriormente, ter seus dados avaliados por um supervisor.

Especificadas as características desse sensor, a escolha do computador embarcado para a montagem do sistema de monitoramento deve levar em consideração os seguintes aspectos: (i) a capacidade de processamento compatível com as necessidades do sistema e (ii) a conexão para utilização de câmeras;

No que concerne ao primeiro aspecto, capacidade de processamento compatível com a necessidade do sistema, retoma-se aqui a pesquisa publicada por Rozental (2009). Em seu trabalho é desenvolvido um dispositivo portátil e robusto que, de forma não invasiva, monitora o estado de alerta do condutor. Em seu dispositivo, utilizou a placa de desenvolvimento SBC2440-I, ilustrada na Figura 5, a qual faz uso do microprocessador Samsung S3C2440A, que opera na arquitetura ARM, e de outro microprocessador AMD Geode LX800, com arquitetura x86. O desempenho dessa placa, que já não está mais disponível para compra, para a aplicação proposta por Rozental (2009), no entanto, não foi satisfatório, conforme resultados e conclusões apresentados pelo autor.

Figura 5 - Placa de desenvolvimento SBC2440-I.



Fonte: Embest Info&Tech Co., LTD.

O segundo aspecto, conexão para utilização de câmeras, no caso do sensor desenvolvido, é fundamental, visto que, por meio de câmeras e das imagens do rosto do condutor, o sistema processa sua tarefa – identificar a fadiga do condutor – e interage

continuamente com o ambiente, em tempo real. Queiroz (2011) desenvolveu um dispositivo não invasivo que, rastreando o rosto e os olhos do motorista, monitora e detecta fadiga. Em sua pesquisa, a câmera escolhida para compor seu sistema fornece uma resolução de 288x352 pixels, com uma taxa de 30 frames por segundo. Porém, a conexão do computador com a câmera selecionada é por USB; conexão essa não recomendada para a utilização dentro de carros e de caminhões, devido às vibrações que podem gerar mau contato e, conseqüentemente, falha no desempenho do sistema.

Por fim, o terceiro aspecto, custo, deve ser considerado no desenvolvimento do sistema, haja vista o fato de sistemas semelhantes serem vendidos no mercado a valores expressivos. Sobre isso, retoma-se novamente o estudo elaborado por Queiroz (2011). Em sua pesquisa, o autor usou a placa FITPC2, que é composta por um Intel Atom 1.6 Ghz Z510, com 1GB de memória RAM. Essas especificações garantiram que o computador utilizado pelo autor supriria as necessidades de sua investigação, visto que a placa tinha um baixo consumo de energia aliado a um bom desempenho. Porém, o custo desse equipamento é muito elevado. Uma placa com *hardware* semelhante custa, em torno de \$79,00 (setenta e nove dólares), o que corresponde a R\$327,06 no câmbio atual, sem impostos e frete.

Ante ao exposto, verifica-se o quão importante são os três aspectos antes mencionados na escolha do computador embarcado. Frente a isso e em vista dos requisitos deste estudo, a pesquisa publicada por Khunpisuth *et al.* (2016) tornou-se fundamental no encaminhamento deste trabalho. Isso porque, esses autores, com o objetivo de criar um sensor para alertar os motoristas sonolentos no ato de dirigir, desenvolveram um experimento capaz de calcular o nível de sonolência do condutor, a partir da frequência com que esse inclina a cabeça e pisca os olhos. Para realizar esse experimento, Khunpisuth *et al.* (2016) utilizaram a placa Raspberry Pi 3, que é composta por um processador Broadcom BCM2837B0 ARM Cortex-A53 Quad-Core, com Clock de 1.4 GHz, e 1GB de memória RAM. Ademais, contém uma entrada para câmera, bastando apenas conectar a câmera à placa, através de um cabo flat. Essa placa oferece um alto poder de processamento aliado a um tamanho reduzido, o que permite seu posicionamento junto à câmera no painel do veículo. No mercado brasileiro, é possível adquiri-la pelo preço de R\$271,90.

O estudo desenvolvido por Khunpisuth *et al.* (2016), embora focalizando a mesma temática que esta pesquisa, difere-se do propósito aqui intentado, uma vez que para reconhecer a sonolência Khunpisuth *et al.* (2016) utilizou o algoritmo proposto por Viola e

Jones (2001) e o tema desta pesquisa utiliza como base o algoritmo proposto pelo Kazemi e Sullivan (2014).

Dito isso, comparando a placa FITPC2 empregada por Queiroz (2011) em sua pesquisa e a placa Raspberry Pi 3 utilizada por Khunpisuth *et al.* (2016), essa última oferece como vantagens: melhor processamento, custo reduzido e disponibilidade de compra no Brasil. Em virtude disso, optou-se por desenvolver o sistema embarcado para monitoramento da sonolência e da fadiga utilizando a placa Raspberry Pi 3, dado que é a que melhor atende aos requisitos deste estudo. Na Figura 6 apresenta-se a câmera já conectada à placa em estudo.

Figura 6 - Placa Raspberry Pi 3 com câmera já conectada.



Fonte: Elaborada pelo autor.

Nas pesquisas presentes na literatura, para além daquelas aqui citadas, observa-se que a identificação do nível de sonolência de motoristas, feita por sistemas embarcados de monitoramento, é realizada por meio do emprego de técnicas e de algoritmos para processamento da imagem capturada pela câmera. Sendo assim, na sequência, detalha-se de que modo essas imagens são processadas.

## 2.2 Processamento de imagem

O desenvolvimento de softwares, de maneira geral, pode ser extenso em termos de tempo de programação, testes e validação escalando de acordo com as necessidades da aplicação. Por exemplo, quando se trata de sistemas baseados em redes neurais e

aprendizagem de máquina faz-se necessário que sejam feitas várias operações antes e depois do processamento, que por si só já é uma etapa complexa e com vários subsistemas, ocorrer.

Com o objetivo de acelerar o processo de desenvolvimento e utilizar ferramentas já validadas, foi feito o uso de bibliotecas. Alguns exemplos de bibliotecas de código aberto disponível é a OpenCV (*Open Source Computer Vision*) e Dlib.

A OpenCV é uma biblioteca escrita em C++. Essa tem grande relevância para a área de processamento de imagens e aprendizado de máquina e vem sendo utilizada por grandes companhias como Google, Microsoft e Intel. Conforme Bradski e Kaehler (2000), a biblioteca apresenta mais de 500 algoritmos para as mais diferentes aplicações e conforme sua documentação, tem uma comunidade ativa com mais de 47 mil desenvolvedores.

Já a DLib também é uma biblioteca, porém mais recente que a OpenCV. Essa, conforme Davis (2009) tem foco em oferecer um conjunto de ferramentas que sejam de fácil acesso a profissionais de diversas áreas, com uma documentação completa e precisa, com várias funções otimizadas e escritas em linguagem C++.

No presente trabalho, essas bibliotecas foram utilizadas para realizar testes dos diversos algoritmos aplicáveis e, ao final do projeto, o desenvolvimento do sistema completo. Diversos são os autores que propuseram algoritmos para processamento de imagens com foco em detecção de faces e características do rosto. Sendo assim, neste trabalho, detalham-se os algoritmos desenvolvidos por: (i) Viola e Jones (2001); (ii) Kazemi e Sullivan (2014); (iii) Malla *et al.* (2010) e (iv) Khunpisuth *et al.* (2016). Analisando cada um desses algoritmos, busca-se evidenciar suas vantagens e suas limitações, a fim de escolher aquele que melhor se adequa à aplicação aqui proposta.

O algoritmo, desenvolvido por Viola e Jones (2001), foi implementado como uma extensão do OpenCV. Esse algoritmo consiste na utilização de máscaras para encontrar um objeto em uma imagem, que, no caso, é o rosto e os olhos do condutor. Cada imagem analisada é subdividida em janelas menores de 24 por 24 pixels. Por meio da matriz de convolução, é possível percorrer a imagem, utilizando essas janelas, em busca de todas as características esperadas que tornam possível determinar um objeto. Essas por sua vez, envolvem isolar uma região da imagem, a fim de encontrar algo que caracterize uma parte do rosto da pessoa. Por exemplo, a região do nariz é mais clara do que a dos olhos e, com isso, pode-se analisar uma característica de linha. Confira, na Figura 7, a ilustração dessa propriedade.

Figura 7 – Característica de linha aplicada ao rosto.

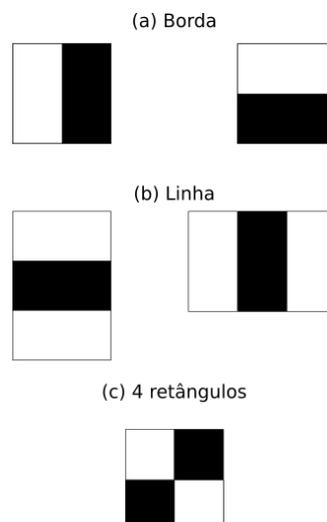


Fonte: Elaborada pelo autor.

Ainda sobre o algoritmo de Viola e Jones (2001), na figura 8, todas as características retangulares analisadas são ilustradas. Em (a), é possível analisar a característica de borda, em que a matriz de convolução é dividida em dois retângulos na vertical ou na horizontal. O valor calculado dessa característica é obtido subtraindo a quantidade de pixels branco da quantidade de pixels preto para cada região.

Em (b), é possível analisar a característica de linha, em que a matriz de convolução é dividida em três retângulos. Nesse caso, o valor calculado é obtido através da soma da quantidade de pixels branco com a quantidade de pixels preto para cada região. Em (c), é possível analisar a característica dos quatro retângulos, e o valor calculado é obtido através da diferença entre pares diagonais do retângulo para cada região.

Figura 8 - Características analisadas de um objeto.



Fonte: Elaborada pelo autor.

Para análise e maior desempenho de todos os recursos envolvidos na detecção das características de um objeto, Viola e Jones (2001) utilizaram uma abordagem de aprendizado de máquina, o Adaboost. Essa abordagem é um tipo de classificador que “aprende”, a partir de exemplos, a classificar os objetos que foram previamente rotulados. Ou seja, o aprendizado de máquina consegue classificar um objeto desconhecido em uma determinada classe, desde que o classificador já tenha sido apreendido.

O algoritmo desenvolvido por Viola e Jones (2001) apresenta, como vantagens, um bom nível de detecção do rosto. Entretanto, possui como limitações o baixo a demora no processamento da imagem, trazendo como consequência o baixo desempenho do sistema quando aplicado a um processamento em tempo real .

Uma alternativa de algoritmo de detecção da face é a proposta por Kazemi e Sullivan (2014). Os autores tiveram como objetivo propor um novo método para tratar de alinhamento da face, identificando a estrutura geométrica. Assim, esse método não apenas detecta a localização e tamanho do rosto como também é capaz de identificar pontos das estruturas faciais como nariz e olhos.

Figura 9 - Resultados obtido pelo algoritmo de alinhamento de face



Fonte: Kazemi e Sullivan (2014).

Para isto, os autores desenvolveram um algoritmo com um conjunto de árvores de regressão, que marca na imagem pontos chaves para a identificação do rosto. As árvores de regressão são um tipo de aprendizagem de máquina supervisionada, ou seja, em que durante o treinamento são fornecidos as entradas e saídas esperadas, voltada para classificação e regressão de dados.

No caso do trabalho de Kazemi e Sullivan (2014), o autor utiliza de várias árvores de regressão, para identificar várias características do rosto. Este algoritmo ajuda a reconhecer o padrão de forma de rosto através da distribuição dos pontos de marcação, utilizando como fonte de treinamento imagens de rostos já com os pontos identificados marcados de forma manual.

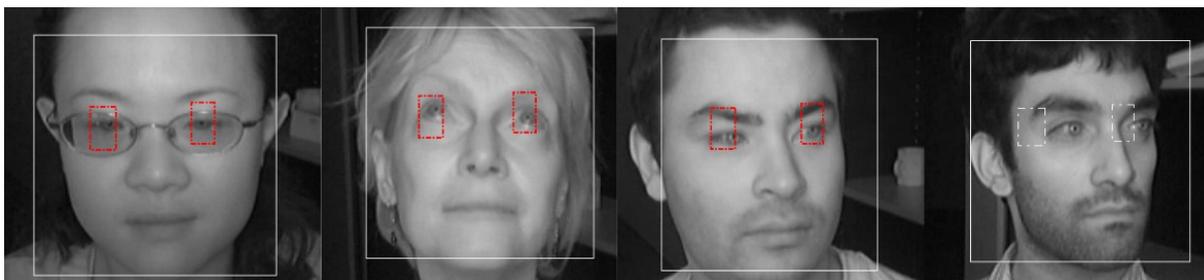
Basicamente, o treinamento se inicia com uma pose média em que os pontos são distribuídos de uma forma padrão no local central da imagem sem se preocupar com a localização exata do rosto ou de suas características. Depois, através de um algoritmo de otimização que estima o erro de cada ponto, os pontos são modificados até se encontrar um erro mínimo. Conforme os autores explicam, há outro algoritmo que otimiza quais recursos (como cor de pixels ou padrões em volta do ponto) devem ser analisados para se encontrar cada ponto.

Após treinado, o sistema torna-se especialista em posicionar os pontos em uma imagem, de acordo com os padrões aprendidos, localizando as marcas de face. No caso de vídeos, essa análise é realizada em cada um dos quadros e ainda garante desempenho que, conforme o título do artigo de Kazemi e Sullivan (2014), tem atraso de milissegundos.

Malla *et al.* (2010), por sua vez, para encontrar o rosto e os olhos em uma imagem, decidiram utilizar o algoritmo proposto por Viola e Jones (2001). Após isolar o rosto da imagem, Malla *et al.* (2010) observaram que, mesmo se o rosto ficasse estático, havia uma variação no tamanho do quadro da área isolada. Por essa razão, esses autores optaram pela aplicação do filtro Kalman no algoritmo. Esse filtro, segundo Clemente (2008, p. 2), também denominado KF, “é uma ferramenta que possibilita lidar com informações ruidosas do sistema dinâmico, utilizando informações matemáticas para estimar o estado atual do ambiente no qual está inserido”. O melhoramento aplicado por Malla *et al.* (2010) ao algoritmo proposto por Viola e Jones (2001) produziu, com a utilização do KF, como vantagens a melhoria na detecção do olho caso o algoritmo de detecção do rosto falhar.

No entanto, o algoritmo de Viola e Jones (2001), mesmo com o melhoramento aplicado por Malla *et al.* (2010), continuava a apresentar limitações, pois, em situações em que o rosto do condutor não estava direcionado ou posicionado para a câmera, o sistema não gerava confiabilidade na leitura. Na Figura 10, é possível analisar que o rosto, capturado pela câmera, não está totalmente direcionado e posicionado, o que produz erros de leitura na detecção dos olhos.

Figura 10 - Testes do algoritmo de Viola e Jones (2001) com rostos na posição frontal.



Fonte: Malla *et al.* (2010, p. 6742).

Além dessa limitação não tratada por Malla *et al.* (2010), esses autores também não abordaram aspectos falhos do algoritmo, como o tempo de processamento que o método consome em cada quadro do vídeo. Esse tempo deve ser considerado e é de grande importância em um sistema embarcado em tempo real, visto que influencia no desempenho do sensor de fadiga.

O último algoritmo aqui analisado é o proposto por Khunpisuth *et al.* (2016). Esse, assim como fez Malla *et al.* (2010), também utilizou o algoritmo proposto por Viola e Jones (2001) em sua composição, agregando a ele aprimoramentos, a fim de alcançar melhores resultados e precisão na leitura. Khunpisuth *et al.* (2016) aperfeiçoaram o algoritmo de Viola e Jones (2001) por meio do isolamento da região de interesse, cujo objetivo foi reduzir o tempo de processamento do sistema e diminuir o número de falso positivo. Além disso, sobre a dificuldade do método em reconhecer a posição dos olhos quando o rosto não está direcionado à câmera, Khunpisuth *et al.* (2016) observaram que o eixo y da região dos olhos poderia identificar a inclinação do rosto e, conseqüentemente, corrigir a leitura da imagem.

Logo, foi implementado, ao algoritmo de Viola e Jones (2001), um algoritmo capaz de girar a imagem até que os dois olhos ficassem alinhados em 180 graus. O estudo de Khunpisuth *et al.* (2016) obteve 83,09% de detecção de rostos de 10 voluntários, sendo 5 do sexo masculino e 5 do sexo feminino.

Conforme figura 11, essa correção realizada por Khunpisuth *et al.* (2016) funciona somente se a cabeça estiver inclinada para um dos lados e com o rosto posicionado à frente da câmera.

Figura 11 - Testes com rostos na posição frontal.



Fonte: Khunpisuth *et al.* (2016, p. 664).

Conforme o Quadro 1, no qual se compara o algoritmo de Viola e Jones (2001) e o algoritmo melhorado por Khunpisuth *et al.* (2016), verifica-se que os melhoramentos feitos por Khunpisuth *et al.* (2016) resultaram em um aumento da precisão da detecção de rosto em cerca de 16%.

Quadro 1 - Comparação entre o algoritmo de Viola e Jones (2001) com o método aplicado por Khunpisuth *et al.* (2016).

Gênero	Comparação entre <i>Haar Cascade</i> e esse papel			
	Total de quadros	Haar Cascade	Trabalho proposto	Diferença
Homem 1	5,236	4,660	5,236	576
Homem 2	3,139	2,509	3,139	630
Homem 3	4,063	3,270	4,015	745
Homem 4	2,330	1,992	2,330	338
Homem 5	2,647	2,216	2,644	428
Mulher 1	3,289	1,862	3,289	1427
Mulher 2	3,317	2,733	3,317	584
Mulher 3	3,512	3,149	3,512	363
Mulher 4	2,853	2,567	2,853	286
Mulher 5	3,570	3,255	3,570	315
<b>Total</b>	<b>33,956</b>	<b>28,213</b>	<b>33,905</b>	<b>5,692</b>

Fonte: Traduzido de Khunpisuth *et al.* (2016).

### 3 METODOLOGIA

Comparando as opções de placa, de câmera e de algoritmo descritas no referencial teórico deste trabalho, neste capítulo, detalha-se o passo a passo que foi executado para a concretização do objetivo geral antes expresso, qual seja: desenvolver um sistema embarcado,

de alta confiabilidade, para monitoramento da sonolência e da fadiga dos condutores de veículos.

Antecipa-se que, para realizar o intento, a escolha pelo *hardware* que oferece o melhor custo-benefício, atendendo às exigências e às necessidades do projeto, foi fundamental. Levando isso em consideração, o computador embarcado escolhido foi a placa Raspberry Pi 3, haja vista as vantagens que esse equipamento oferece, quais sejam: custo baixo, alto poder de processamento e compatibilidade para uso de câmeras nativas.

Dito disso, o presente trabalho foi desenvolvido em seis etapas, descritas a seguir, a saber: (i) instalação das ferramentas de desenvolvimento; (ii) aquisição da imagem; (iii) detecção de sonolência e de fadiga; (iv) montagem do equipamento; (v) implementação do sistema de monitoramento embarcado em um veículo e rodagens-teste; e (vi) análise de dados.

### 3.1 Instalação das ferramentas de desenvolvimento

Após a escolha do *hardware* para o desenvolvimento do sistema embarcado pretendido, iniciou-se a instalação das ferramentas necessárias ao seu funcionamento, quais sejam: a) instalação do sistema operacional Linux no computador embarcado; b) instalação e configuração do VNC e c) instalação da biblioteca de visão computacional.

Em concordância com a placa de desenvolvimento escolhida, pode-se escolher entre sistemas operacionais com kernel linux ou versão específica do Microsoft Windows 10. Os sistemas Linux, têm como principal vantagem o custo inexistente, visto que o kernel linux é open-source e gratuito, em relação ao Microsoft Windows 10.

No site da Fundação Raspberry Pi há disponível diversas opções de sistemas linux, cada um apresentando características específicas como o ambiente gráfico ou sistema de gerenciamento de pacotes. Atualmente, a principal distribuição Linux para o Raspberry Pi 3 é a Raspbian OS que oferece interface gráfica habilitada por padrão e ser fortemente otimizada para a placa, visto que é mantida pela própria fundação.

Após o download do sistema, a instalação consiste em gravar a imagem de disco em um cartão de memória o que pode ser feito em softwares específicos para tal. O No caso deste trabalho, foi utilizado o Gnome Disk Utility, ferramenta disponível por padrão no Ubuntu Linux.

A classe do cartão de memória, isto é a sua velocidade de leitura e escrita, irá influenciar tanto nesta etapa quanto na utilização do sistema: quanto menor a classe, mais lento serão as operações que envolvem arquivos no disco o que pode comprometer o desempenho geral do sistema. Por exemplo, o atraso de escrita no cartão, durante uma operação de salvar um evento, pode reduzir a precisão do sistema durante o processamento de imagens. Sendo assim, neste trabalho foi utilizado um cartão de memória classe 10 (maior classe existente), com velocidade mínima de 10MB/s.

Após gravar no cartão de memória a imagem do Raspbian OS, o mesmo foi inserido no local apropriado no Raspberry Pi 3 que, por sua vez, foi ligado a uma fonte de energia para iniciar a instalação. Com o sistema instalado, este pode ser acessado via SSH o que permitiu acesso a terminal de comandos.

Porém, com o objetivo de facilitar o desenvolvimento do sistema optou-se por habilitar a interface gráfica no Raspbian OS. Porém para acessar de forma remota, isto é, sem conectar o hardware diretamente a um monitor de vídeo e sim através da rede, e com modo

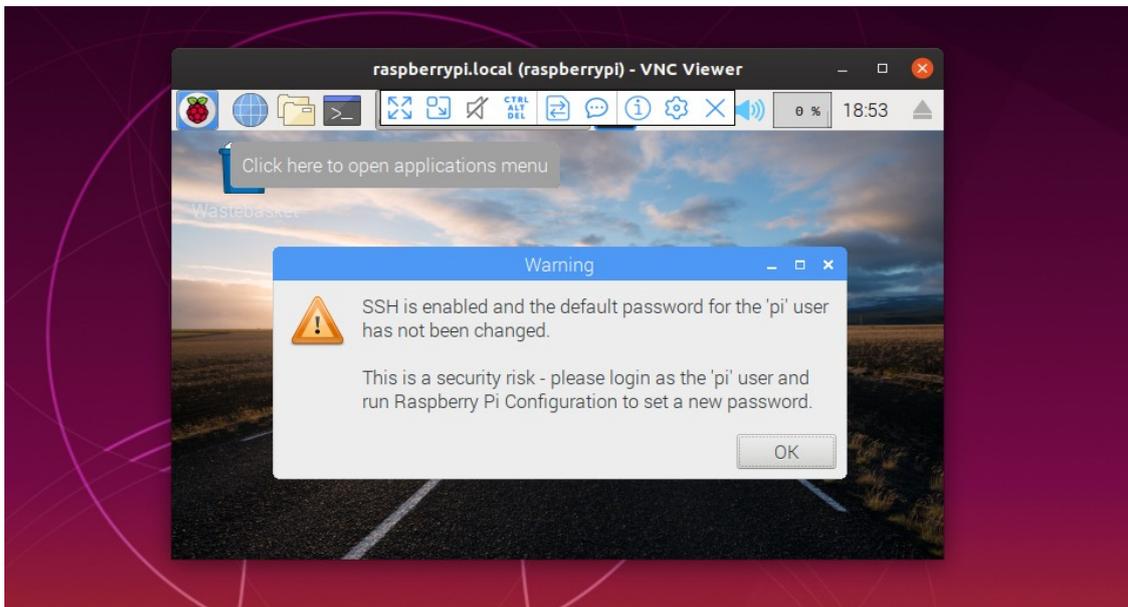
gráfico habilitado foi utilizado o sistema Virtual Network Computing (VNC) ou em tradução livre computação de rede virtual.

Basicamente, o VNC permite que um computador seja acessado remotamente através de uma rede, com exibição em tempo real da interface gráfica e interações do usuário. Neste projeto, este sistema permite acompanhar em tempo real o vídeo capturado pela câmera e os pontos analisados durante o processamento de imagens.

Já logado no sistema, foi executado no terminal: “sudo apt update && sudo apt install tightvncserver”. Este comando, atualizou a lista de pacotes que podem ser instalados e depois instalou o software “Tight VNC Server”.

Com o software instalado, o servidor VNC foi inicializado com o comando “tightvncserver -geometry 800x600” com a resolução desejada para o sistema, nesse caso 800px por 600px. Para acessar o Raspberry Pi através do VNC foi utilizado o software Remmina já presente no sistema Ubuntu Linux.

Figura 12 - Utilização do sistema operacional usando o VNC



Fonte: Elaborada pelo autor.

Por fim, para realizar testes do algoritmo proposto por Viola e Jones (2001) na placa Raspberry Pi 3, foi necessário instalar a biblioteca de visão computacional OpenCV. Essa biblioteca dispõe de diferentes ferramentas para processamento de imagem. É possível efetuar sua instalação através dos repositórios do sistema operacional da placa Raspberry Pi 3, utilizando o comando: sudo apt install python3-opencv. Para verificar se a instalação foi bem-

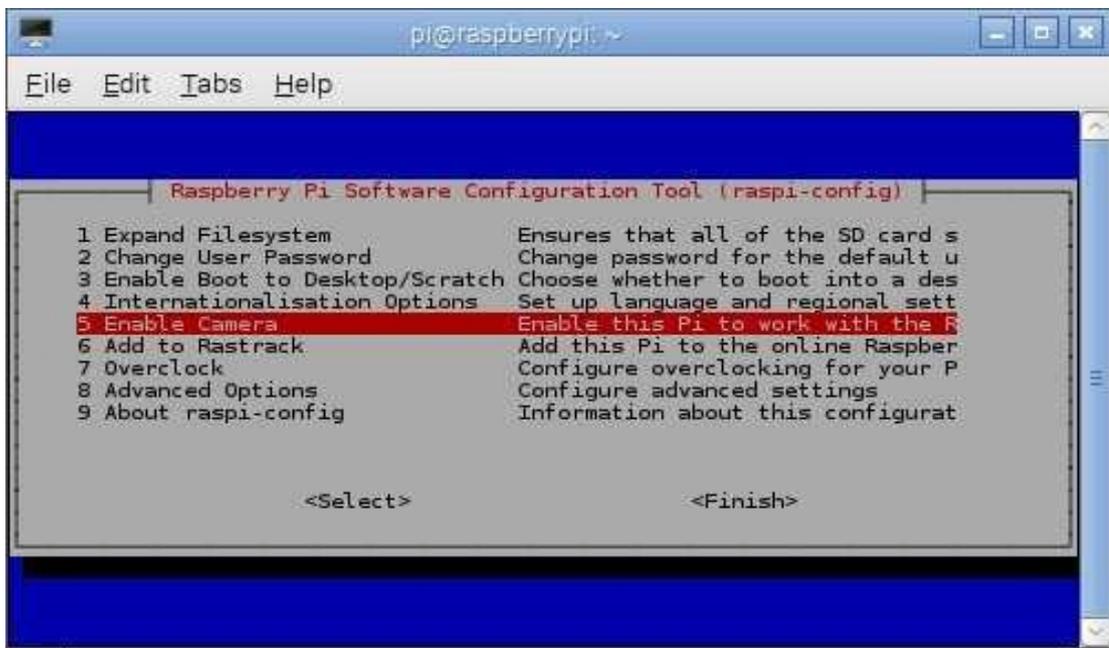
sucedida, emprega-se o comando `python3 -c "import cv2; print(cv2.__version__)"`. A saída mostra a versão que foi instalada.

### 3.2 Aquisição da imagem

Com base nas câmeras utilizadas por Rozental (2009) e Queiroz (2011) em suas pesquisas, escolheu-se desenvolver este trabalho com a câmera do Raspberry Pi, a qual utiliza o sensor OV5647. Esse sensor dispõe de 5 megapixels de resolução e, ao redor da placa, contém LEDs infravermelhos que facilitam a obtenção da imagem mesmo com pouca luminosidade. Além disso, essa câmera se conecta diretamente ao Raspberry Pi 3, permitindo a obtenção das imagens com alta taxa de amostragem e qualidade.

Inicialmente, para efetuar a aquisição da imagem, é necessário ativar a função de câmera no sistema operacional do Raspberry Pi 3. Para isso, digitou-se no terminal `sudo raspi-config` e selecionou-se a opção 5 Enable Camera, conforme ilustrado na Figura 13.

Figura 13 - Configuração da câmera no Raspberry Pi.



Fonte: Elaborada pelo autor.

Feito isso, testou-se a quantidade de frames por segundo, utilizando a função VideoCapture do OpenCV. Para isso, por meio de algumas linhas de programação, foi escrito um programa para medir a quantidade de frames por segundo em uma amostragem de 100 quadros.

Figura 14 - Teste aquisição de imagem.

```

vs = cv2.VideoCapture(0)
fps = FPS().start()

while fps._numFrames < 100:
    (grabbed, frame) = vs.read()
    frame = imutils.resize(frame, width=400)

    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF
    fps.update()

fps.stop()
print("[INFO] Normal FPS: {:.2f}".format(fps.fps()))

```

Fonte: Elaborada pelo autor.

Realizado o teste, o resultado obtido foi de 28.98 FPS. A função `read()` do `VideoCapture` da biblioteca do `OpenCV`, que foi utilizada no programa anterior, aguarda o quadro a ser lido pela câmera para, depois, prosseguir com o restante da programação. Como a placa `Raspberry Pi` contém 4 núcleos de processamento, foi possível mover a função de leitura dos quadros da câmera para uma *Thread* separada. Sobre a *Thread*, diz Scheffer (2007), que se trata de um mecanismo que “permite que processos sequenciais sejam executados paralelamente, apesar de poderem fazer chamadas ao sistema que bloqueiam processos”. Utilizando esse mecanismo de threads, é possível aumentar a velocidade de processamento na leitura da câmera. A função `WebcamVideoStream` da biblioteca `imutils` já contém isso implementado. Com isso, o código de programação passou a ser:

Figura 15 - Teste aquisição de imagem com Thread.

```

vs = WebcamVideoStream(src=0).start()
fps = FPS().start()

while fps._numFrames < 100:
    frame = vs.read()
    frame = imutils.resize(frame, width=400)

    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF
    fps.update()

fps.stop()
print("[INFO] Thread FPS: {:.2f}".format(fps.fps()))

```

Fonte: Elaborada pelo autor.

O resultado obtido testando com o uso de *Thread* foi de 39.05 FPS, o que representa um aumento de 35% em relação ao resultado anterior. Esse resultado pode ser aumentado ainda mais em ambas as situações, caso não seja exibida a imagem na tela, com o uso da função *imshow* da biblioteca *OpenCV*.

### 3.3 Detecções de sonolência e de fadiga

Inicialmente, empregou-se o algoritmo de Viola e Jones (2001) para encontrar o rosto e os olhos dentro de uma imagem capturada pela câmera. O *OpenCV* já disponibiliza classificadores pré-treinados, que são arquivos no formato XML, para essa detecção. Dentre esses classificados, os que se aplica nesta pesquisa são: *frontalFace.xml*, *profileFace.xml* e *eye.xml*.

Com o objetivo de testar a eficiência desses classificadores, foi desenvolvido um algoritmo que utiliza o classificador pré-treinado de face frontal (*frontalFace.xml*) por um período de amostragem de 500 quadros. Na Figura 16 ilustra-se o código de programação para efetuar esse teste.

Figura 16 - Teste com classificador frontal face

```
faceCascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
faceCascadeProfile = cv2.CascadeClassifier("haarcascade_profileface.xml")

vs = WebcamVideoStream(src=0).start()
fps = FPS().start()

while fps._numFrames < 500:
    frame = vs.read()
    frame = imutils.resize(frame, width=400)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    facesFrontal = faceCascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=3, minSize=(30, 30) )
    for (x, y, w, h) in facesFrontal:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF
    fps.update()

fps.stop()
print("[INFO] FPS: {:.2f}".format(fps.fps()))
```

Fonte: Elaborada pelo autor.

Foram realizados movimentos leves com a cabeça. A eficiência obtida com o uso do classificador pré-treinado de rostos frontal foi de 88% com 29 FPS. Em uma situação normal do sensor, é necessário que ele consiga detectar o rosto mesmo com a face um pouco inclinada para qualquer lado. Para isso, desenvolveu-se um algoritmo que, caso o rosto não

seja detectado com a utilização do classificador de face frontal (frontalFace.xml), tente encontrar o rosto utilizando o classificador de face de perfil, o profileFace.xml. Na Figura 17 apresenta-se o código de programação atualizado.

Figura 17 - Teste com classificador frontal face + profile face

```

faceCascade = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
faceCascadeProfile = cv2.CascadeClassifier("haarcascade_profileface.xml")

vs = WebcamVideoStream(src=0).start()
fps = FPS().start()

while fps._numFrames < 500:
    frame = vs.read()
    frame = imutils.resize(frame, width=400)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    facesFrontal = faceCascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=3, minSize=(30, 30) )
    for (x, y, w, h) in facesFrontal:
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

    if(len(facesFrontal) == 0):
        facesProfile = faceCascadeProfile.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=3, minSize=(30, 30) )
        for (x, y, w, h) in facesProfile:
            cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

    cv2.imshow("Frame", frame)
    key = cv2.waitKey(1) & 0xFF
    fps.update()

fps.stop()
print("[INFO] FPS: {:.2f}".format(fps.fps()))

```

Fonte: Elaborada pelo autor.

Movimentando a cabeça como no primeiro teste, a eficiência na detecção da imagem aumentou para 99%, porém o desempenho diminuiu para 17 FPS. Em virtude do baixo FPS e por ter que implementar ainda outras técnicas de processamento de imagem para saber se o olho está aberto ou fechado, testou-se o algoritmo de Kazemi e Sullivan (2014).

O algoritmo de Kazemi e Sullivan (2014) foi implementado na biblioteca do Dlib. Para otimizar mais ainda o algoritmo, e visto que neste trabalho não se fazia necessário conhecer todas as posições de marcações do rosto, o treinamento do algoritmo proposto por Kazemi e Sullivan (2014) foi realizado apenas com os 6 pontos em volta de cada um dos olhos.

A abordagem de utilizar somente a identificação dos olhos ajuda a aumentar o desempenho geral do sistema pelo simples fato de haver menos dados a serem analisados resultando em menor tempo de análise. Na Figura 18 ilustra-se o código de programação para efetuar esse teste.

Figura 18 - Resultado do teste com o algoritmo de Kazemi e Sullivan (2014)

```

for rect in rects:
    (x, y, w, h) = face_utils.rect_to_bb(rect)
    cv2.rectangle(image, (x, y), (x + w, y + h), (0, 255, 0), 2)

    shape = predictor(gray, rect)
    shape = face_utils.shape_to_np(shape)

    leftEye = shape[0:6]
    rightEye = shape[6:12]
    leftEAR = eye_aspect_ratio(leftEye)
    rightEAR = eye_aspect_ratio(rightEye)

    ear = (leftEAR + rightEAR) / 2.0

    for (sX, sY) in shape:
        cv2.circle(image, (sX, sY), 1, (0, 255, 0), -1)

    leftEyeHull = cv2.convexHull(leftEye)
    rightEyeHull = cv2.convexHull(rightEye)
    cv2.drawContours(image, [leftEyeHull], -1, (0, 255, 0), 1)
    cv2.drawContours(image, [rightEyeHull], -1, (0, 255, 0), 1)

```

Fonte: Elaborada pelo autor.

Na figura 19 é possível visualizar os 6 pontos que foi desenhado no rosto. Com esses 6 pontos conseguimos calcular a distância entre a parte superior do olho com a parte inferior, e encontrar uma proporção para essas dimensões denominado em inglês *Eye Aspect Ratio* (EAR). Quando o olho está aberto, experimentalmente observou-se que é possível encontrar valores por volta de 0.32 e fechado de 0.16 para essa proporção.

Figura 19 - Resultado do teste com o algoritmo de Kazemi e Sullivan (2014)



Fonte: Elaborada pelo autor.

O algoritmo da implementação da sonolência analisa se o valor de EAR está abaixo de 0.18 e conta quantos milissegundos ele permaneceu nesse estado. Se o motorista permanecer com os olhos fechados durante um valor considerável, o protótipo acionará um alarme sonoro a fim de sinalizar o mesmo.

### 3.4 Montagem do equipamento

Para alimentar a placa Raspberry Pi 3 e a câmera, empregou-se um conversor DC-DC no modo *Step Down*, para converter os 12V ou 24V do automóvel em 5V. Dos conversores mais conhecidos no mercado, destacam-se: AMS1117-5.0 e LM2596. O quadro 2 evidencia algumas características desses conversores.

Quadro 2 – Características dos conversores AMS1117-5.0 e LM2596.

Característica	AMS1117-5.0	LM2596
Corrente de saída (A)	1	3
Tensão de entrada máximo (V)	20	40
Tensão de entrada mínimo (V)	6.5	6.5

Fonte: Elaborado pelo autor

Segundo Souza, Oliveira e Pilla (2018), o Raspberry Pi 3 exigiu da fonte de alimentação cerca de 1.34A. Em consequência disso, no protótipo proposto, decidiu-se empregar o conversor LM2596, representado na Figura 20.

Figura 20 – Regulador de tensão LM2596.



Fonte: Elaborada pelo autor.

### 3.5 Implementação do sistema de monitoramento embarcado em um veículo de teste

Após validar os componentes de forma independente, fez-se necessário juntar tudo como um único protótipo, com um encapsulamento para proteger o hardware e também permitir que todo o sistema seja fixado no veículo.

Para o desenvolvimento do encapsulamento do protótipo foi utilizado o *software FreeCad* por ser de código aberto, multiplataforma, com capacidade de desenho assistido por computador em três dimensões. As características desejadas durante o projeto desse encapsulamento foram de permitir utilização de sistema de refrigeração externo, e fácil montagem utilizando de parafusos.

Como forma de fabricação do protótipo, optou-se pela impressão 3D, por oferecer grande facilidade de prototipagem rápida a custos muito baixos. O material escolhido foi o plástico Acrilonitrila butadieno estireno (ABS) por ter um boa resistência mecânica, necessária para garantir a proteção dos componentes internos, e térmica, visto que dentro de um veículo estacionado exposto ao Sol poderia desgastar o material.

A impressão foi realizada em impressora do próprio autor, e alguns ajustes tiveram que ser feitos no projeto original, no software CAD. Estes ajustes visaram corrigir o posicionamento dos furos para encaixe da câmera que ficaram ligeiramente errados.

Visando a facilidade de instalação do protótipo, foi utilizado um suporte ajustável, com ventosa. Esse suporte permite que todo o sistema seja posicionado da melhor forma possível permitindo enquadramento do rosto do motorista pela câmera.

A figura 21 mostra o protótipo finalizado, após impressão e montagem com todos os componentes.

Figura 21 - O protótipo



Fonte: Elaborada pelo autor.

O protótipo foi acoplado ao vidro do para-brisa utilizando o suporte de ventosa à vácuo, sendo energizado usando a saída de 12V do veículo no período da tarde de um dia da semana. Na figura 22 é possível visualizar o protótipo no vidro do carro.

Figura 22 - Protótipo instalado



Fonte: Elaborada pelo autor.

## 4 ANÁLISE DOS DADOS E DISCUSSÕES

Após a etapa anterior de rodagens-teste ter sido concluída, efetuou-se um levantamento de dados estatísticos de falsos-positivos oriundos do reconhecimento de face. Esse levantamento foi efetuado analisando os arquivos gerados no cartão de memória do Raspberry Pi 3. Com isso, foi possível verificar a confiabilidade de leitura do sistema através das estatísticas de falsos-positivos e da quantidade de quadros por segundos que o sistema consegue analisar. Essa parte deste trabalho apresenta os resultados obtidos a partir da análise desses testes.

### 4.1 Custos de hardware do protótipo

Para a realização desse protótipo foi necessário comprar componentes no Brasil e China. O único item importado da China foi a câmera devido à falta de disponibilidade a pronta entrega no Brasil. A câmera teve um custo de \$16.04, no quadro é convertido para reais utilizando a cotação do dólar na época em que foi comprado. O protótipo foi impresso em plástico ABS em uma impressora 3D no qual foi gasto somente 200 gramas, porém a quantidade mínima de compra é de 1KG.

Quadro 3 – Custos do protótipo

<b>Item</b>	<b>Preço</b>
Placa Raspberry Pi 3	R\$233,90
Câmera Raspberry Pi Infravermelho	R\$72,01
Suporte Ventosa Á Vácuo	R\$34,89
Cartão de memória 16GB	R\$45,00
1KG filamento ABS	R\$50,00
Cooler 12V	R\$10,00
Regulador de tensão LM2596	R\$15,00
Buzzer Piezoelétrico	R\$16,90
<b>TOTAL</b>	<b>R\$477,70</b>

Fonte: Elaborado pelo autor

#### 4.2 O protótipo

Após rodagens-testes foi possível constatar que o nível de robustez e estabilidade empregados na construção, foram suficientes para que o protótipo desempenhasse sua função de processamento de imagem mesmo com alguns desníveis do asfalto. O uso do suporte de ventosa é bastante prático e versátil possibilitando obter o melhor posicionamento do protótipo no vidro do carro.

#### 4.2 Teste de desempenho

Para determinar o nível de confiabilidade do sistema, o protótipo foi submetido a vários testes de bancada. O primeiro teste é o teste de piscada. Esse teste foi realizado diurno e noturno com ajuda de 2 pessoas no qual foi solicitado que as mesmas se comportasse normalmente em frente ao protótipo para que pudesse ser avaliado o número de piscadas, positivo verdadeiro, falso positivo e falso negativo. Os resultados são mostrados no quadro abaixo.

Quadro 4 – Resultado do teste de piscada diurno

<b>Gênero</b>	<b>Total de piscadas</b>	<b>Positivo verdadeiro (%)</b>	<b>Falso positivo (%)</b>	<b>Falso negativo (%)</b>
Homem 1	22	84.00	16.00	0.00
Homem 2	25	96.00	0.00	4.00
<b>Total</b>	<b>47</b>	<b>90.00</b>	<b>8.00</b>	<b>2.00</b>

Fonte: Elaborado pelo autor

Quadro 5 – Resultado do teste de piscada noturno

<b>Gênero</b>	<b>Total de piscadas</b>	<b>Positivo verdadeiro (%)</b>	<b>Falso positivo (%)</b>	<b>Falso negativo (%)</b>
Homem 1	21	100.00	0.00	0.00
Homem 2	25	84.00	0.00	16.00
Mulher 1	30	96.66	3.34	0.00
<b>Total</b>	<b>76</b>	<b>93.55</b>	<b>1.11</b>	<b>5.33</b>

Fonte: Elaborado pelo autor

Comparando-se os resultados obtidos deste projeto no qual utilizou a metodologia proposta por Kazemi e Sullivan (2014), com os resultados de Khunpisuth et al. (2016) (Quadro 6), podemos verificar que a taxa de confiabilidade de Khunpisuth et al. (2016) é maior.

Quadro 6 - Taxa de precisão da face e detecção do olho

<b>Gênero</b>	<b>Total de piscadas</b>	<b>Positivo verdadeiro (%)</b>	<b>Falso positivo (%)</b>	<b>Falso negativo (%)</b>
Homem 1	95	100.00	0.00	0.00
Homem 2	103	99.03	0.00	0.97
Homem 3	100	99.00	1.00	1.00
Homem 4	83	98.80	1.20	1.20
Homem 5	88	100.00	11.36	0.00
Mulher 1	99	98.99	0.00	1.01
Mulher 2	100	100.00	3.00	0.00
Mulher 3	100	100.00	1.00	0.00
Mulher 4	100	100.00	0.00	0.00
Mulher 5	98	100.00	5.10	0.00
<b>Total</b>	<b>966</b>	<b>99.59</b>	<b>2.17</b>	<b>0.41</b>

Fonte: Traduzido de Khunpisuth et al. (2016).

Outro teste realizado foi o de sonolência diurno e noturno. Esse teste foi realizado com 3 pessoas no qual foi solicitado que atue como se estivesse em um estado de sonolência em momentos aleatórios. Em uma amostragem de 10 sonolências para cada pessoa, foi anotado se o protótipo detectou como sonolência, falso positivo ou falso negativo. Os resultados são mostrados no quadro abaixo.

Quadro 7 – Resultado do teste de sonolência diurno

<b>Gênero</b>	<b>Total de sonolências</b>	<b>Positivo verdadeiro (%)</b>	<b>Falso positivo (%)</b>	<b>Falso negativo (%)</b>
Homem 1	11	100.00	0.00	0.00
Homem 2	11	81.81	0.00	18.19
Mulher 1	6	100.00	0.00	0.00
<b>Total</b>	28	93.93	0.00	6.06

Fonte: Elaborado pelo autor

Quadro 8 – Resultado do teste de sonolência noturno

<b>Gênero</b>	<b>Total de sonolências</b>	<b>Positivo verdadeiro (%)</b>	<b>Falso positivo (%)</b>	<b>Falso negativo (%)</b>
Homem 1	12	91.66	0.00	8.34
Homem 2	8	75.00	0.00	25.00
Mulher 1	6	100.00	0.00	0.00
<b>Total</b>	26	88.88	0.00	11.11

Fonte: Elaborado pelo autor

## 5 CONSIDERAÇÕES FINAIS

A partir dos resultados obtidos nesse trabalho apresentados na seção análise de dados e discussões pode se verificar que o desempenho do sistema proposto ficou abaixo do esperado. Vários fatores podem ter contribuído para esses resultados, como por exemplo a temperatura de operação da placa Raspberry Pi 3, condições de iluminação do ambiente de testes, imprecisão do classificador pré treinado.

No caso da temperatura o Raspberry Pi 3 se aquece durante o período de funcionamento. A medida que a temperatura aumenta o próprio sistema, para evitar danos permanentes, diminui o clock do processador e consequentemente diminui a velocidade de processamento. Por isso deve ser estudado melhores formas de dissipador de calor ou sistemas de refrigeração alternativas.

Ainda assim, a placa Raspberry Pi 3 se mostrou uma boa escolha para o desenvolvimento desse projeto, devido a flexibilidade com a utilização de sistemas Linux e com bastante documentação disponível. O custo do desenvolvimento do protótipo com esta placa, também demonstrou baixo quando comparado a produtos comercialmente disponíveis.

Já para as condições de iluminação, podemos destacar a importância de uma boa iluminação bem como um posicionamento bem próximo da câmera pelo usuário. No caso de Khunpisuth et al. (2016) não foi citado com precisão como foi efetuado os testes de desempenho. No caso desse trabalho, todos os testes foram baseados em condições mais próximas do real, com o protótipo instalado no veículo e com imagens capturadas em uma real condução pelo motorista.

A câmera escolhida apresentou bons resultados com qualidade de imagem satisfatória e boa nitidez mesmo em ambientes com baixa iluminação. Apesar disso, futuramente pode ser estudado diferentes lentes para a câmera de forma a aumentar o campo de visão ou tentar aproximar mais o foco ao motorista.

Em relação à imprecisão do classificador pré treinado, observou se que muitas vezes a posição encontrada por este para os olhos do motorista, estava imprecisa, resultando na classificação incorreta do estado de sonolência do motorista. Como esse trabalho foi focado em desenvolver a aplicação, não foi dedicado tempo hábil para treinamento de rede neural específica para o classificador. Por este motivo, em trabalhos futuros pode-se efetuar esse treinamento a fim de verificar o nível de confiabilidade do sistema.

## REFERÊNCIAS

BATISTA, P. Dormir ao volante está entre as principais causas de mortes no trânsito: sono e o cansaço estão relacionados à 60% dos acidentes no país. **Portal do Trânsito**, 6 de setembro de 2019. Disponível em: <https://portaldotransito.com.br/noticias/dormir-ao-volante-esta-entre-as-principais-causas-de-mortes-no-transito/>. Acesso em: 8 nov. 2019.

CHASE, O. Sistemas embarcados. **SBAjovem**, dez. 2007. Disponível em: [http://www.maxpezzin.com.br/aulas/6\\_EAC\\_Sistemas\\_Embarcados/1\\_SE\\_Introducao.pdf](http://www.maxpezzin.com.br/aulas/6_EAC_Sistemas_Embarcados/1_SE_Introducao.pdf). Acesso em: 7 dez. 2019.

DE SOUZA, Rodrigo Clemente Thom. Previsão de séries temporais utilizando rede neural treinada por filtro de Kalman e evolução diferencial. 2008.

KAZEMI, V.; SULLIVAN, J. One millisecond face alignment with an ensemble of regression trees. *In: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*. IEEE Computer Society, USA, 2014. p. 1867-1874.

KHUNPISUTH, O. *et al.* Driver drowsiness detection using eye-closeness detection. *In: 2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. IEEE, 2016. p. 661-668.

MALLA, A. M. *et al.* Automated video-based measurement of eye closure for detecting behavioral microsleep. *In: 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. Buenos Aires, 2010, p. 6741-6744.

QUEIROZ, K. L. **Sistema baseado em vídeo para detecção de sonolência em motoristas**. 2011. 55 f. Orientador: Ícaro dos Santos. Dissertação (Mestrado em Engenharia Elétrica) – Faculdade de Tecnologia, Universidade de Brasília, Brasília, 2009.

ROZENTAL, V. N. **Detecção de fadiga de condutores através de processamento de imagem em tempo real em sistemas embarcados**. Orientador: Ícaro dos Santos. 2009. 48 f. Trabalho de Conclusão de Curso (Bacharelado em Engenharia Elétrica) - Faculdade de Tecnologia, Curso de Graduação em Engenharia Elétrica, Universidade de Brasília, Brasília, 2009.

SOUZA, G. S.; OLIVEIRA, J. V. V. T; PILLA, M. L. Comparação de Desempenho do Workload YCSB em Raspberry PI B+ e 3. *In: SCOLA REGIONAL DE ALTO DESEMPENHO DA REGIÃO SUL (ERAD-RS)*, 2018, Porto Alegre. **Anais da XVIII Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul**. Porto Alegre: Sociedade Brasileira de Computação, 2018. Disponível em: <https://sol.sbc.org.br/index.php/eradr/article/view/2792>. Acesso em: 15 nov. 2019.

SCHEFFER, Rosely. Uma visao geral sobre threads. *Campo Digital*, v. 2, n. 1, 2007.

SBC2440-I. **SBC2440-I**. Página inicial . Disponível em: <https://doc.xdevs.com/docs/Embtest/SBC2440I.pdf>. Acesso em: 8 de dezembro de 2020.

TREEEARTH. **TreeEarth**. Página inicial . Disponível em: <<http://www.treearth.com.br/I-VUE.php>>. Acesso em: 8 de dezembro de 2020.

VELTEC. **Sensor de fadiga - Veltec**. Página inicial . Disponível em: <<https://veltec.com.br/marketplace/sensor-de-fadiga/>>. Acesso em: 8 de dezembro de 2020.

VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. *In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. CVPR, v. 1, Kauai, HI, USA, 2001.