

UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”

Faculdade de Ciências - Campus Bauru

Bacharelado em Sistemas de Informação

**GERAÇÃO DE DISPOSITIVO SINALIZADOR DE SONOLÊNCIA PARA
CONDUTOR DE VEÍCULOS A FIM DE CONTER ACIDENTES DE TRÂNSITO**

Lucas Cegielkowski Guimarães

Bauru-SP

2021

UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”

Faculdade de Ciências - Campus Bauru

Bacharelado em Sistemas de Informação

**GERAÇÃO DE DISPOSITIVO SINALIZADOR DE SONOLÊNCIA PARA
CONDUTOR DE VEÍCULOS A FIM CONTER ACIDENTES DE TRÂNSITO**

Lucas Cegielkowski Guimarães

Orientador: Prof. Dr. João Paulo Papa

Trabalho de Conclusão de Curso apresentado à
Faculdade de Ciências da Universidade Estadual
Paulista “Júlio de Mesquita Filho” - Câmpus de
Bauru, para obtenção do grau de Bacharel em
Sistemas de Informação.

Bauru-SP

2021

G963g Guimarães, Lucas Cegielkowski
Geração de dispositivo sinalizador de sonolência para conduta de
veículos a fim de conter acidentes de trânsito / Lucas Cegielkowski
Guimarães. -- Bauru, 2021
38 f. : il.

Trabalho de conclusão de curso (Bacharelado - Sistemas de
Informação) - Universidade Estadual Paulista (Unesp), Faculdade de
Ciências, Bauru
Orientador: João Paulo Papa

1. Visão computacional. 2. Inteligência artificial. 3.
Reconhecimento ocular. 4. Python. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Faculdade de
Ciências, Bauru. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

AGRADECIMENTOS

Começo agradecendo a Deus por, ao longo deste processo complicado e desgastante, ter me mostrado o caminho, nos momentos em que pensei em desistir. Não posso deixar de agradecer a UNESP por ser um espaço que privilegia o conhecimento e onde todas as ideias são bem recebidas.

Deixo também um agradecimento especial a minha companheira Fernanda Furlani, por toda força e apoio durante esse processo, sem ela esta pesquisa não teria sido possível.

Aos meus pais, eu devo a vida e todas as oportunidades que nela tive e que espero um dia poder lhes retribuir. Agradeço também ao meu irmão que me ajudou em vários testes da pesquisa.

Deixo o agradecimento ao meu orientador que sempre me inspirou muito, desde o primeiro dia de aula na universidade. Quero deixar também agradecimento ao professor Remo, que sempre ajudou a iluminar e traçar os caminhos acadêmicos.

Agradeço ainda aos meus amigos e familiares que ao longo desta etapa me encorajaram e me apoiaram, fazendo com que esta fosse uma das melhores fases da minha vida.

RESUMO

A sonolência é uma das principais causas de acidentes nas rodovias. Portanto, um dispositivo que reconheça se o motorista de um veículo está sonolento pode prevenir muitos acidentes em todo o mundo. Estudos baseados na fisiologia do sono provaram que existe uma relação entre o tempo de piscar e a sonolência humana, e tecnologias poderosas, como as existentes no campo da visão computacional, podem ajudar no desenvolvimento de dispositivos com esse fim. Sendo assim, esta pesquisa desenvolveu, utilizando um Raspberry PI 4 Model B, junto de um algoritmo desenvolvido na linguagem de programação Python, com a ajuda da biblioteca CV2, um dispositivo que torna possível o reconhecimento ocular do condutor, possibilitando distinguir se os olhos estão abertos ou fechados, tanto de dia quanto de noite, para, dessa forma, poder classificar o estado do motorista, entre acordado, sonolento ou dormindo.

Palavras chave: Visão computacional. Raspberry PI. OpenCV. Python. Reconhecimento ocular. Inteligência artificial.

ABSTRACT

Drowsiness is one of the main causes of road accidents. Therefore, a device that recognizes whether the driver of a vehicle is drowsy can prevent many accidents around the world. Studies based on sleep physiology have proven that there is a relationship between blink time and human drowsiness, and powerful technologies, such as those in the field of computer vision, can help in the development of devices for this purpose. Therefore, this research developed, using a Raspberry PI 4 Model B, together with an algorithm developed in the Python programming language, with the help of the CV2 library, a device that makes the driver's eye recognition possible, making it possible to distinguish whether the eyes are open or closed, both during the day and at night, in order to be able to classify the driver's state as awake, drowsy or sleeping.

Keywords: Computer vision. Raspberry PI. OpenCV. Python. Eye recognition. Artificial intelligence.

LISTA DE FIGURAS

FIGURA 1 - Diagrama de dados de pixel.....	13
FIGURA 2 -Uma visão geral da pipeline de reconhecimento facial OpenCV.	15
FIGURA 3 - Características semelhantes a Haar	16
FIGURA 4 - Imagem integral.....	17
FIGURA 5 - Algoritmo AdaBoost.....	18
FIGURA 6 - Classificador Cascade.....	19
FIGURA 7 - Anti Sleep Pilot	20
FIGURA 8 - Dispositivo sempre alerta.....	21
FIGURA 9 - Dispositivo gerado pelo autor.....	25
FIGURA 10 - Raspberry PI Modelo 4 B.....	27
FIGURA 11 - Câmera infravermelho.....	28
FIGURA 12 - Interface do Raspberry PI OS.....	28
FIGURA 13 - Buzzer ativo.	29
FIGURA 14 - Placa de ensaio.....	30
FIGURA 15 - Placa de resfriamento.....	30
FIGURA 16 - Usuário 1 em estado sonolento.....	31
FIGURA 17 - Usuário 2 classificado com olhos abertos.....	32
FIGURA 18 - Usuário 4 com atraso na classificação.....	33
FIGURA 19 - Usuário 5 classificado como dormindo.....	33

LISTA DE TABELAS

TABELA 1 - Custo da pesquisa.....	24
TABELA 2 - Tempo de atraso e ocorrências.....	34
TABELA 3 - Acertos e erros do software.....	34

LISTA DE DIAGRAMAS

DIAGRAMA 1 - Fluxo do software.....	25
-------------------------------------	----

SUMÁRIO

1. INTRODUÇÃO.....	10
2. RECORTE TEÓRICO METODOLÓGICO PARA CRIAÇÃO DA PESQUISA.....	12
2.1 Visão Computacional.....	12
2.2 OpenCV.....	13
2.3 Algoritmo Viola e Jones.....	15
2.4 Python.....	19
3. SOLUÇÕES JÁ EXISTENTES.....	20
4. ABORDAGEM METODOLÓGICA: PERCURSO DA PESQUISA.....	22
4.1 Desenvolvimento do dispositivo.....	23
4.2 Interface de uso.....	24
4.3 Custo da pesquisa.....	24
4.4 Diagrama de funcionamento do software.....	25
5. TECNOLOGIAS PESQUISADAS E UTILIZADAS.....	26
5.1 Raspberry PI.....	26
5.2 Câmera.....	27
5.3 Sistema operacional.....	28
5.4 Buzzer ativo.....	28
5.5 Placa de ensaio.....	29
5.6 Capa de resfriamento para Raspberry PI.....	30
6. RESULTADOS.....	31
7. PROJETOS FUTUROS.....	36
8. CONSIDERAÇÕES FINAIS.....	37
REFERÊNCIAS.....	38

1. INTRODUÇÃO

A Organização Mundial da Saúde (OMS) relata que aproximadamente 1,35 milhões de pessoas morrem em acidentes de carro em todo o mundo. De um modo geral, os acidentes rodoviários devem-se principalmente a métodos de condução inadequados. Essas condições podem ocorrer se o motorista estiver drogado, sonolento ou alcoolizado.

Segundo o NSC (National Safety Council), a velocidade de reação, o reconhecimento dos perigos e o foco pioram quando mais sonolento o condutor fica. Além disso, o motorista também tem três vezes mais chances de se acidentar se estiver cansado.

Levando em consideração esse problema, pensou-se que uma tecnologia avançada poderia ajudar os motoristas a serem sinalizados de seu estado de sonolência e, conseqüentemente, não sofreram nenhum acidente por conta desse fato.

Os computadores de hoje em dia estão cada vez mais potentes, permitindo trazer novas tecnologias e técnicas computacionais cada vez mais avançadas, inclusive graças a esta evolução, foi criado os microcomputadores como o Raspberry PI, o qual possui um poder computacional incrível do tamanho de um cartão de crédito.

A Visão Computacional é uma área da inteligência artificial que é responsável pelo treinamento de computadores em busca da compreensão e interpretação do mundo visual. Isto acontece através do deep learning com o uso de imagens que possibilitam às máquinas reagirem e tomarem decisão de acordo com o que elas veem.

Dessa forma, pretende-se com essa pesquisa, elaborar um dispositivo tecnológico para auxiliar no monitoramento de condutores de veículos trazendo por meio desse sinais sonoros que detectam estado de sonolência, tendo a linguagem de programação como instrumento auxiliar. Para atender a esse objetivo geral,

foram estabelecidos como específicos: a) selecionar linguagens de programação que facilitem a construção e qualidade do dispositivo; b) comprar equipamentos que funcionem de acordo com o objetivo geral e tenha um custo-benefício de qualidade; c) elaborar um código pertinente para que reconheça a sonolência do condutor e, assim, ativar o sinal sonoro e, d) superar os déficits dos dispositivos já existentes.

Sendo assim, esta investigação justifica-se por trazer à tona um assunto-problema dos dias atuais e, através disso, desenvolver um mecanismo com custo acessível para amenizar a problemática utilizando as tecnologias existentes. Nesse caso, viabiliza a possibilidade de aplicação de um algoritmo complexo de visão computacional em um microcomputador.

2. RECORTE TEÓRICO METODOLÓGICO PARA CRIAÇÃO DA PESQUISA

Este capítulo tende apresentar os pressupostos teóricos-metodológicos que fundamentam a pesquisa. Desse modo, apresenta-se a visão computacional sendo utilizada através do OpenCV com a implementação do Viola e Jones.

2.1 Visão Computacional

Segundo Brownlee, a visão computacional, muitas vezes abreviada como CV, é definida como um campo de pesquisa que visa desenvolver tecnologia para ajudar os computadores a "ver" e compreender o conteúdo de imagens digitais (como fotos e vídeos).

Visão computacional é o processo de usar máquinas para entender e analisar imagens (fotos e vídeos). Embora esses tipos de algoritmos tenham aparecido de várias formas desde a década de 1960, os avanços mais recentes no aprendizado de máquina, bem como saltos no armazenamento de dados, capacidade de computação e dispositivos de entrada baratos de alta qualidade, levaram a grandes melhorias no desempenho de nosso software.

Visão computacional é um termo geral para qualquer cálculo que envolva conteúdo visual, o que significa imagens, vídeos, ícones e qualquer coisa que envolva pixels, como exemplo tem-se a figura 1.

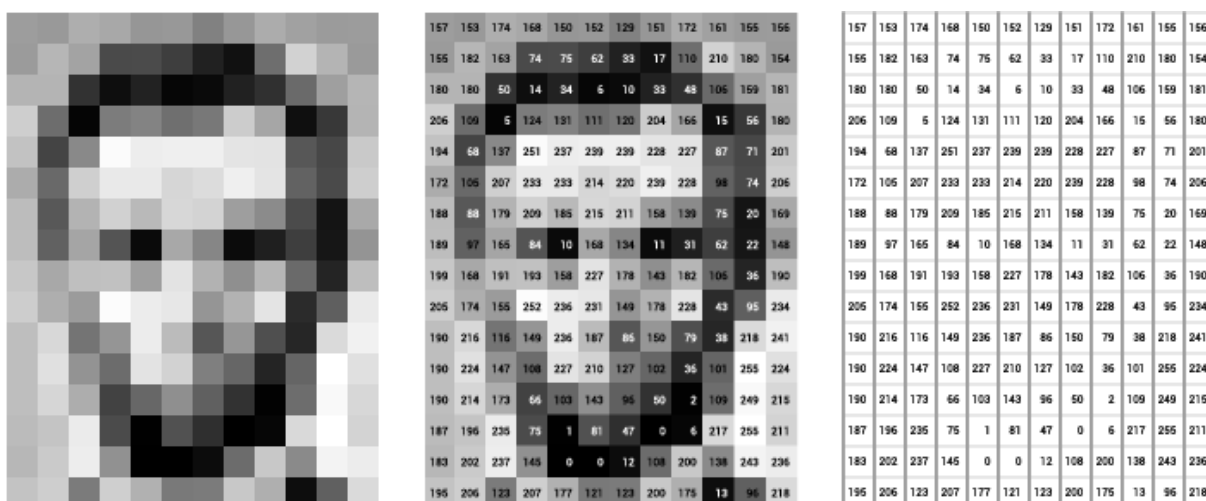


Figura 1 - Diagrama de dados de pixel. À esquerda, a imagem de Lincoln; no centro, os pixels marcados com números de 0 a 255, representando seu brilho; e à direita, esses números sozinhos. Disponível: <https://openframeworks.cc/ofBook/chapters/image_processing_computer_vision.html>. Acesso em: 20 de julho de 2021.

2.2 OpenCV

Segundo o site oficial da OpenCV (Open Source Computer Vision Library), ela é uma biblioteca de software de visão computacional e aprendizado de máquina de código aberto. A biblioteca foi construída para prover uma infraestrutura comum para aplicativos de visão computacional e para acelerar o uso da percepção da máquina em produtos comerciais.

A biblioteca possui mais de 2.500 algoritmos de otimização, incluindo um conjunto completo de algoritmos de visão computacional e aprendizado de máquina clássicos e de última geração. Esses algoritmos podem ser usados para detectar e reconhecer rostos, reconhecer objetos, classificar o comportamento humano em vídeos, rastrear o movimento da câmera, rastrear objetos em movimento, extrair modelos 3D de objetos, gerar nuvens de pontos 3D de câmeras estéreo, encontrar imagens semelhantes no banco de dados, remover olhos vermelhos de imagens tiradas com flash, rastrear os movimentos dos olhos.

O OpenCV conta com mais de 47 mil usuários em sua comunidade e com downloads estimados a mais de 18 milhões. É amplamente utilizada em empresas, grupos de pesquisa e por órgãos governamentais.

Empresas bem estabelecidas como Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota que empregam a biblioteca, existem muitas startups, como Applied Minds, VideoSurf e Zeitera, que fazem uso extensivo do OpenCV. Os usos implantados do OpenCV abrangem desde juntar imagens de street view, detectar intrusões em vídeo de vigilância em Israel, monitorar equipamentos de mina na China, ajudar robôs a navegar e pegar objetos na Willow Garage, detecção de acidentes de afogamento em piscinas na Europa, exibição de arte interativa em Espanha e Nova York, verificando as pistas em busca de destroços na Turquia, inspecionando rótulos em produtos em fábricas ao redor do mundo até a detecção rápida de rostos no Japão.

Contém interfaces C ++, Python, Java e MATLAB e conta com suporte Linux, Windows, Android e Mac OS. O OpenCV se destina principalmente para aplicações de visão em tempo real. OpenCV foi desenvolvido em C ++ e contém uma interface modelada que trabalha perfeitamente com contêineres STL.

A figura 2 demonstra o funcionamento do reconhecimento facial ao colocar uma imagem no fluxo de reconhecimento.

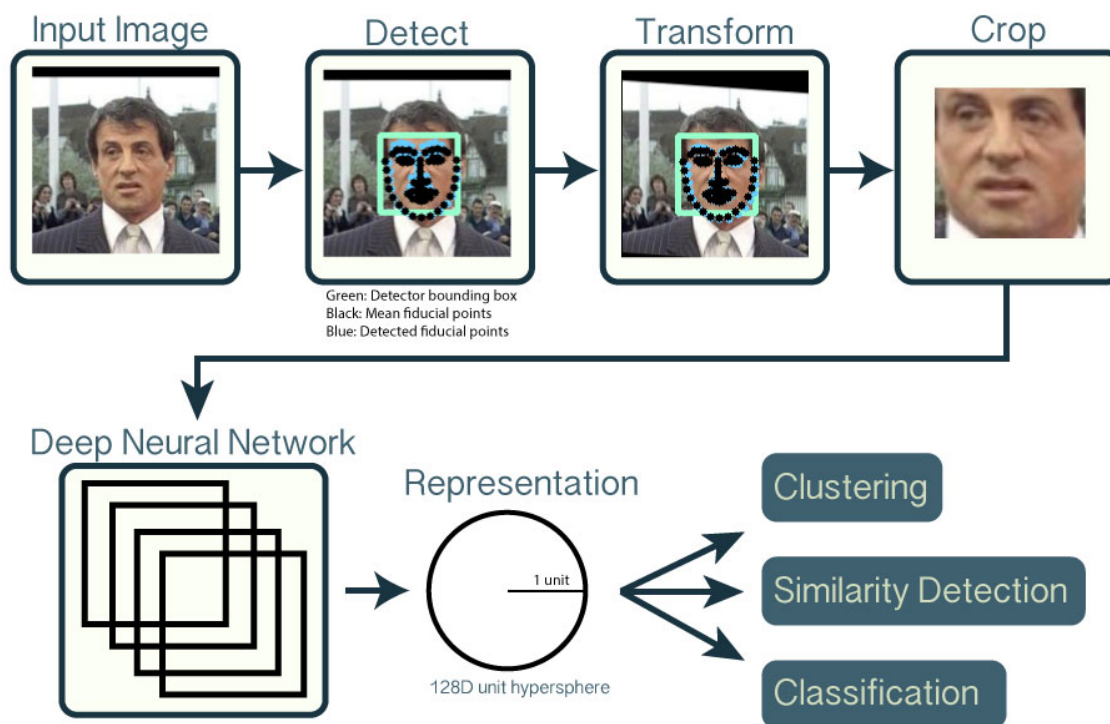


Figura 2 - Uma visão geral da pipeline de reconhecimento facial OpenCV. Que gera embeddings faciais de 128-d. Disponível: <<https://cmusatyalab.github.io/openface/>>. Acesso em: 18 de julho de 2021.

2.3 Algoritmo Viola e Jones

O algoritmo de Viola Jones tem o nome de dois pesquisadores da visão computacional, que propuseram o método em 2001, Paul Viola e Michael Jones no artigo "Rapid Object Detection using a Boosted Cascade of Simple Features".

Embora seja uma estrutura desatualizada, Viola-Jones é muito poderoso e sua aplicação tem se mostrado muito significativa na detecção de rosto em tempo real. A velocidade de treinamento desse algoritmo é muito lenta, mas ele pode detectar rostos em tempo real a uma velocidade incrível. Dada uma imagem (o algoritmo é adequado para imagens em tons de cinza), o algoritmo verifica muitas sub-regiões menores e tenta encontrar um rosto procurando por características específicas em cada sub-região. É necessário verificar muitas posições e

proporções diferentes, porque uma imagem pode conter muitas faces de tamanhos diferentes. Viola e Jones usam recursos semelhantes a Haar neste algoritmo para detectar rostos humanos.

O algoritmo Viola-Jones é um mecanismo muito usado para detecção de objetos. A principal propriedade deste algoritmo é que o treinamento é lento, mas a detecção é rápida. Este algoritmo usa filtros de recurso de base Haar, portanto, não usa multiplicações.

O algoritmo tem quatro etapas principais: 1) seleção de recursos semelhantes a Haar; 2) criação de uma imagem integral; 3) execução do treinamento AdaBoost; e 4) criação de cascatas classificadoras.

A figura 3 demonstra na parte superior características semelhantes a Haar e na parte inferior demonstra como calculá-las.

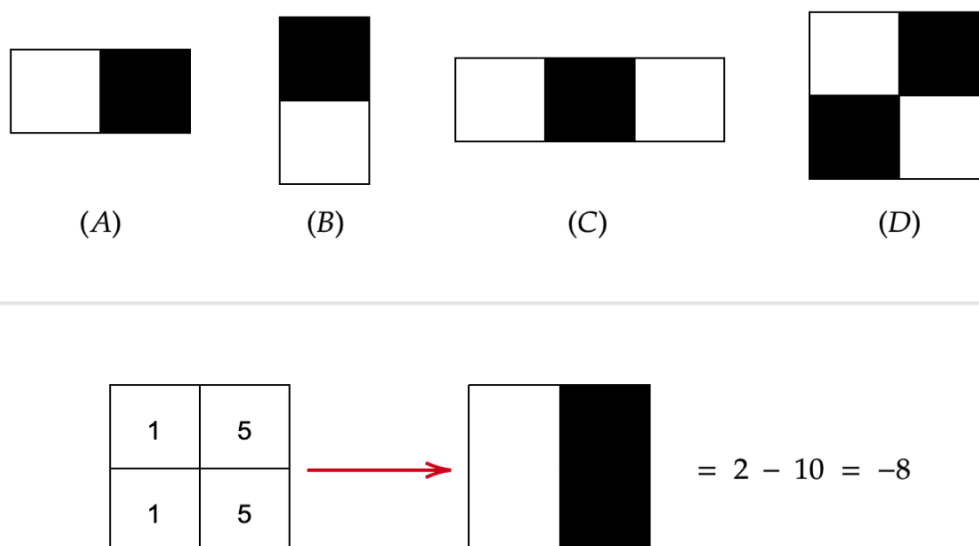
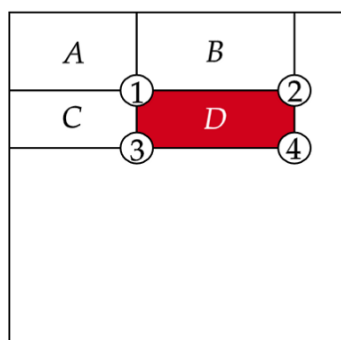
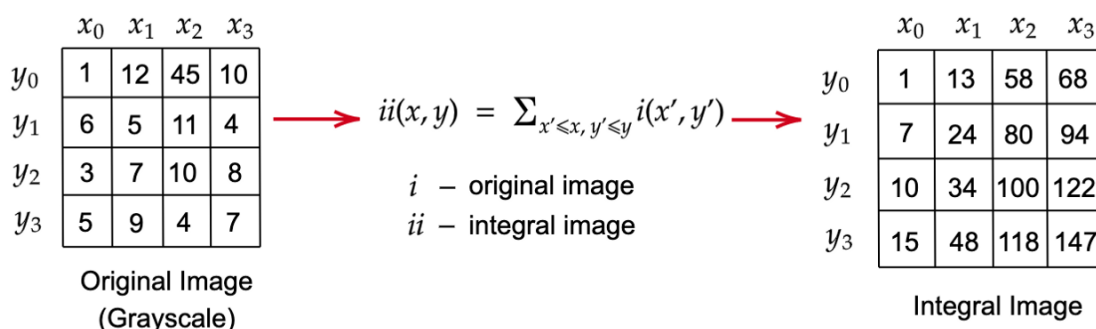


Figura 3 - Características semelhantes a Haar (parte superior) e como calculá-las (parte inferior). Disponível: <<https://towardsdatascience.com/understanding-face-detection-with-the-viola-jones-object-detection-framework-c55cc2a9da14>>. Acesso em: 18 de julho de 2021.

Já na figura 4 tem-se a demonstração de como ocorre a conversão de uma imagem original em uma imagem integral na parte superior, onde a imagem da esquerda superior é a imagem original e a imagem da direita superior é a imagem integral, já na parte inferior da imagem pode ser visto como calcular uma região retangular usando uma imagem integral.

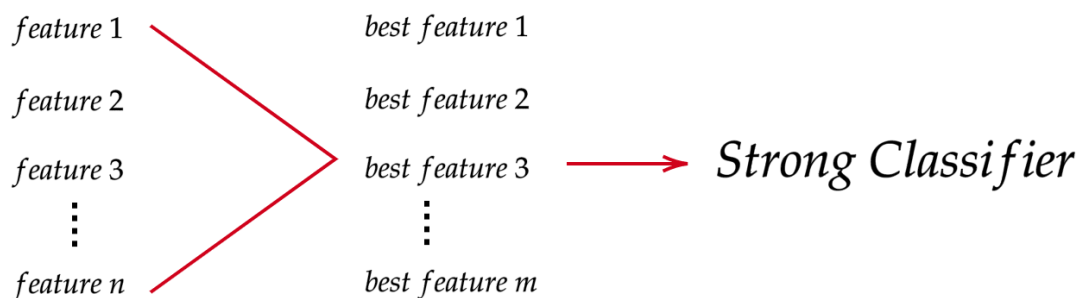


$$D = \textcircled{4} - \textcircled{2} - \textcircled{3} + \textcircled{1}$$

$$= \textcircled{4} + \textcircled{1} - (\textcircled{2} + \textcircled{3})$$

Figura 4 - Conversão da imagem original em imagem integral (parte superior) e como calcular uma região retangular usando uma imagem integral (parte inferior). Disponível: <<https://towardsdatascience.com/understanding-face-detection-with-the-viola-jones-object-detection-framework-c55cc2a9da14>>. Acesso em: 18 de julho de 2021.

De acordo com Somet Lee, o algoritmo AdaBoost (Adaptive Boosting) é um algoritmo para selecionar o melhor subconjunto de recursos entre todos os recursos disponíveis. O retorno do algoritmo é um classificador denominado “Classificador Forte” ou em inglês “Strong Classifier”, a Figura 5 demonstra o algoritmo.



where $n > m$

Figura 5 - O objetivo de usar o algoritmo AdaBoost é extrair os melhores recursos de n recursos. Observação: os melhores recursos também são conhecidos como classificadores fracos. Disponível: <<https://towardsdatascience.com/understanding-face-detection-with-the-viola-jones-object-detection-framework-c55cc2a9da14>>. Acesso em: 18 de julho de 2021.

Também de acordo com Somet Lee o classificador em cascata é um classificador de vários estágios que pode realizar a detecção com rapidez e precisão. Cada estágio consiste em um classificador forte produzido pelo algoritmo AdaBoost. De um estágio para outro, o número de classificadores fracos em um classificador forte aumenta, sendo que uma entrada é avaliada em uma base sequencial. Se um classificador para um estágio específico produz um resultado negativo, a entrada é descartada imediatamente, mas, caso a saída seja positiva, a entrada é encaminhada para o próximo estágio.

De acordo com Viola P. e Jones (2001), esta abordagem de múltiplos estágios permite a construção de classificadores mais simples que podem então ser usados para rejeitar a maioria das entradas negativas rapidamente, enquanto se gasta mais tempo nas entradas positivas, como demonstra a figura 6.

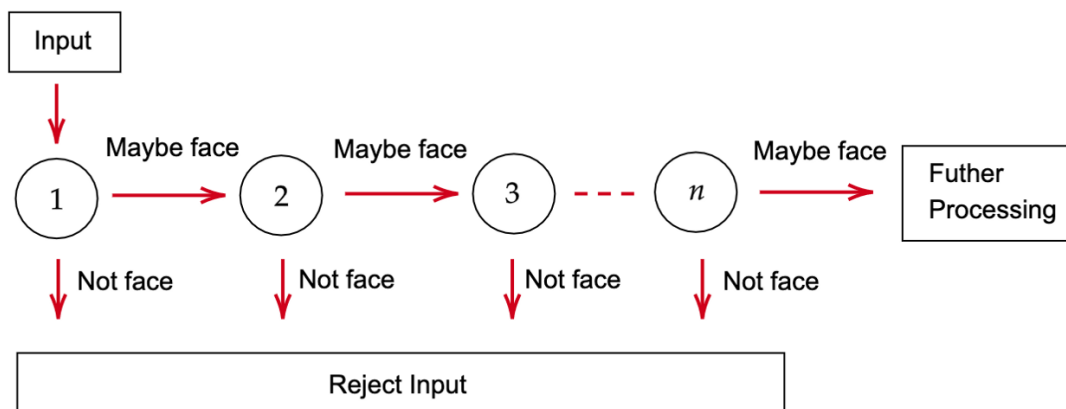


Figura 6- O Classificador Cascade. Disponível: <<https://towardsdatascience.com/understanding-face-detection-with-the-viola-jones-object-detection-framework-c55cc2a9da14>>. Acesso em: 18 de julho de 2021.

2.4 Python

Segundo o site oficial do Python, a própria linguagem é descrita como uma linguagem de programação de alto nível de uso geral que pode ser aplicada a muitos tipos diferentes de problemas.

O Python vem com uma vasta biblioteca padrão que contém áreas como processamento de strings (expressões regulares, Unicode, cálculo de diferenças entre arquivos), protocolos de Internet (HTTP, FTP, SMTP, XML-RPC, POP, IMAP, programação CGI), engenharia de software (teste de unidade, registro, criação de perfil, análise de código Python) e interfaces de sistema operacional (chamadas de sistema, sistemas de arquivos, soquetes TCP / IP).

Além disso, a linguagem traz um benefício muito grande quando se trata da área da inteligência artificial, pois é muito fácil de utilizar quando o assunto é computação quantitativa e analítica, visto que ela contém uma variedade de bibliotecas para manipulação de dados.

3. SOLUÇÕES JÁ EXISTENTES

A pesquisa proposta já foi abordada em outros produtos com funcionalidades semelhantes, porém com alguns pontos importantes a se destacar que mostram a ineficácia das aplicações encontradas para solucionar tal problema. Devido a isso, esta pesquisa passa a ter como objetivo também, superar esse déficit de outras plataformas, como explanado na introdução deste documento.

Algumas soluções existentes podem ser encontradas como a Anti Sleep Pilot, a qual o motorista tem que ficar apertando o botão dele a cada X tempo, que varia dependendo da sonolência do condutor, e baseado no tempo de resposta consegue fazer uma detecção do nível de cansaço do motorista e sugerir uma parada (mostrado na figura 7).

Também pode ser encontrado o Sempre Alerta, um dispositivo colocado atrás da orelha e quando inclina a cabeça dispara um alarme para acordar o usuário, caso da figura 8.



Figura 7 - Dispositivo Anti Sleep Pilot fixado no painel do do carro. Disponível: <<https://www.nytimes.com/2011/05/22/automobiles/22DROWSY.html>>. Acesso em: 18 de julho de 2021.



Figura 8- Dispositivo sempre alerta. Disponível: <<https://www.semprealertavida.com.br/sempre-alerta-vida/>>. Acesso em: 18 de julho de 2021.

Portanto, é possível ver nas figuras 7 e 8, esses produtos que já existem com a mesma finalidade proposta nesta pesquisa, mas não contam com uma tecnologia tão adequada para a facilidade e proteção dos motoristas, uma vez que o Anti Sleep Pilot, que apesar de ter uma boa eficácia, traz um grande incômodo para os motoristas de ter que ficar apertando ele durante toda viagem, onde ele fica sempre emitindo sinais sonoros para você apertar ele.

Já o dispositivo Sempre Alerta, segundo o (Alarme Anti-Sono: riscos de usar o aparelho para não dormir), o movimento da cabeça é um dos últimos efeitos da sonolência segundo o fisioterapeuta e especialista da CPAPS, Clêdisson Souza, trazendo muitos riscos ao utilizar o dispositivo, já que como o movimento da cabeça é um dos últimos estágios da sonolência, até ele sinalizar, o motorista já pode ter causado um acidente.

4. ABORDAGEM METODOLÓGICA: PERCURSO DA PESQUISA

A pesquisa foi desenvolvida em cinco etapas: a primeira foi destinada à escolha do hardware a ser utilizado que, levando em consideração, custo benefício, poder de processamento e facilidade de acoplação a periféricos, foi escolhido o Raspberry PI. A segunda etapa foi o momento de escolher os acessórios que colaboraram na eficácia do dispositivo, sendo assim, escolheu-se uma câmera infravermelha, já que uma das necessidades era poder interpretar as imagens mesmo sem luz no ambiente e, devido a tecnologia desta câmera, é possível que as imagens sejam interpretadas até mesmo no escuro. Outro acessório escolhido foi o buzzer ativo para emissão sonora do alarme, já que ele além de fácil utilização via software é também de fácil instalação no Raspberry PI.

Já a terceira etapa da pesquisa foi selecionar a linguagem de programação que auxiliaria na viabilidade do dispositivo, e isso não poderia ser diferente, o Python com sua grande comunidade e seu grande suporte de funcionalidades que facilitam a implementação da visão computacional, também contava com o suporte no Raspberry PI, se tornando uma ótima escolha para se integrar ao projeto. Além dele, foi escolhido o OpenCV como biblioteca auxiliar da linguagem e o método Viola e Jones para o reconhecimento facial.

Na quarta etapa, o software foi desenvolvido através de uma lógica que fez o monitoramento ocular detectar se a pessoa estava com os olhos abertos ou fechados. Dessa forma, possibilitou desenvolver o algoritmo para classificar o condutor como acordado, sonolento ou dormindo.

Na quinta e última etapa, o software foi testado em seis pessoas para gerar um conjunto de dados de teste e, conseqüentemente, detectar a eficácia do dispositivo.

Dessa forma, tem-se com esta pesquisa um Raspberry PI com uma câmera infravermelha que é responsável por tornar possível a utilização da visão computacional, através do algoritmo em Python.

Levando isso em consideração, os principais requisitos do dispositivo levantados para o desenvolvimento foram:

- Detecção de sonolência no condutor;
- Alarme sonoro para acordar o motorista;
- Funcionar durante o dia e durante a noite;
- Fácil instalação;
- Custo acessível.

4.1 Desenvolvimento do dispositivo

O dispositivo precisa conectar um cabo USB C que ficará responsável por prover energia elétrica, este cabo pode ser conectado em uma tomada, em um acendedor de cigarro, em um usb ou em uma bateria externa, visto que o USB C é removível, então pode ser substituído facilmente para outra interface de preferência do motorista. Nesse caso, o dispositivo pode ser implantado de maneira simples, necessitando apenas de uma fonte de energia.

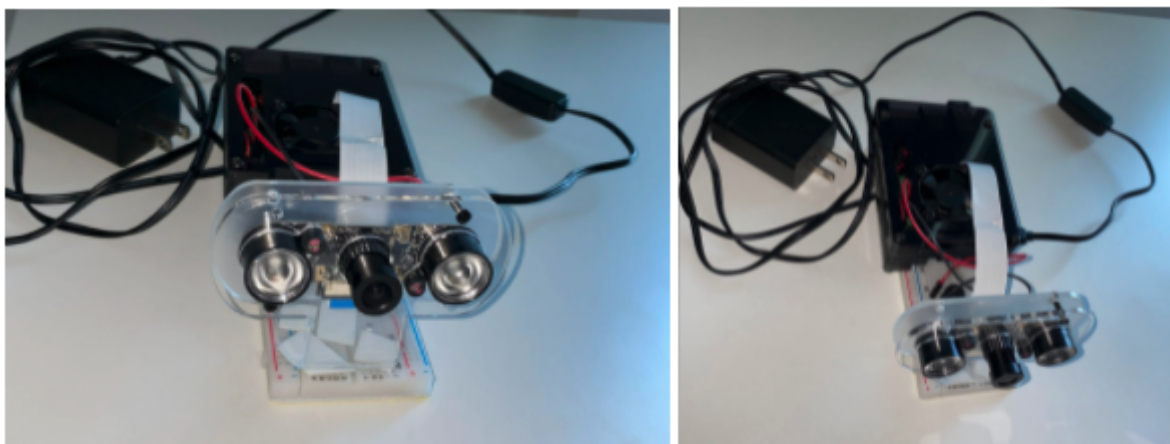


Figura 9- Dispositivo gerado pelo autor. Fonte: elaborado pelo autor.

4.2 Interface de uso

Após adquirir um o dispositivo o motorista só precisa ligá-lo a energia elétrica e fixá-lo no painel do carro com a câmera apontada para seu rosto, a partir disso, o dispositivo já começa a funcionar e monitorar a sonolência do condutor disparando, assim, o alarme quando houver estado de sonolência.

4.3 Custo da pesquisa

Abaixo contém a tabela com o custo de todos os equipamentos e acessórios utilizados para o desenvolvimento da pesquisa.

Descrição	Valor em reais
Raspberry PI Model 4B 4G Ram	436,31
Breadboard + buzzer + fios	30,68
Câmera Infravermelho	73,73
Capa de resfriamento	57,62
TOTAL	598,34

Tabela 1: Custo da pesquisa. Fonte: elaborado pelo autor.

Pela tabela é possível concluir que o gasto total da pesquisa foi de 598,34 reais, um valor considerado acessível devido ao benefício oferecido pelo dispositivo.

4.4 Diagrama de funcionamento do software

O diagrama abaixo demonstra como funciona a lógica do software para detectar se o motorista está sonolento, também, a definição do estado de sonolência e de sono, onde foi possível encontrar os dados da duração do piscar dos olhos, sendo que se a duração do piscar dos olhos estiver abaixo de 400 ms é porquê o condutor está em estado normal, e acima de 400 ms ele está em estado de sonolência, caso esse tempo ultrapasse 800 ms já pode considerar o estado da pessoa como dormindo, esses dados foram retirados do Drowsy Driver Detection System Using Eye Blink Patterns.

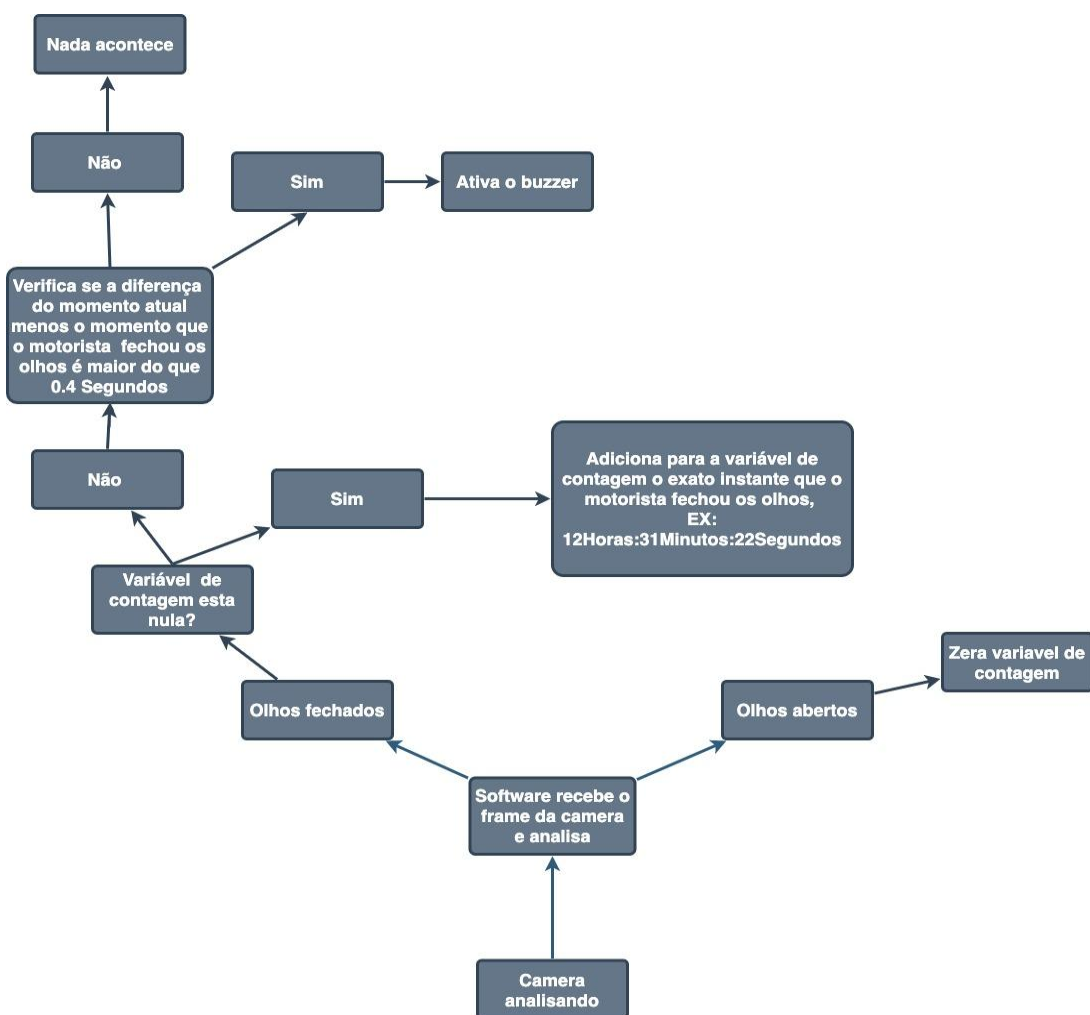


Diagrama 1: Fluxo do software. Fonte: elaborado pelo autor.

5. TECNOLOGIAS PESQUISADAS E UTILIZADAS

Neste capítulo são tratadas as tecnologias escolhidas e as motivações para tal escolha, além das especificações necessárias para o funcionamento do produto.

5.1 Raspberry PI

Devido a necessidade de um processamento considerável, foi escolhido o RASPBERRY PI, um minicomputador, semelhante a um computador ou notebook. A diferença é que este dispositivo é compacto e contém os principais componentes de um computador numa placa do tamanho de um cartão de crédito. Para mais, ele possui um custo benefício ótimo e traz facilidade de utilização com outros acessórios.

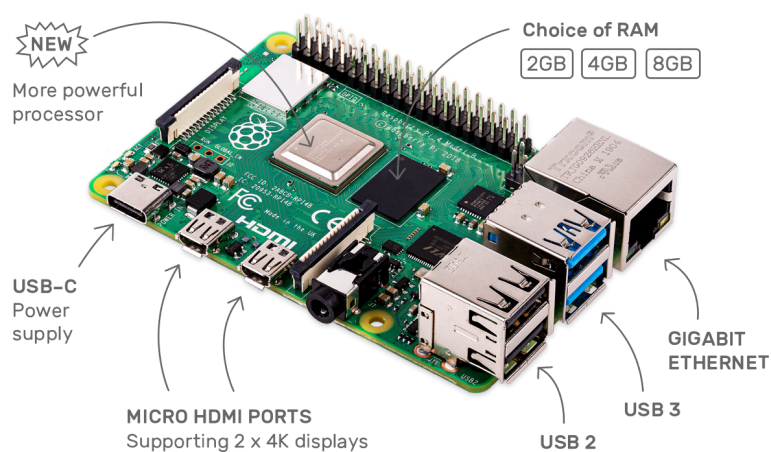


Figura 10 - Raspberry PI Modelo 4 B. Disponível: <<https://www.raspberrypi.org/products/raspberry-pi-4-model-b/?variant=raspberry-pi-4-model-b-4gb>>. Acesso em: 18 de julho de 2021.

Configurações: Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) SoC de 64 bits a 1,5 GHz, 4GB LPDDR4-3200 SDRAM, 2,4 GHz e 5,0 GHz IEE, 802.11ac sem fio, Bluetooth 5.0, BLE, Gigabit Ethernet, 2 portas USB 3.0; 2 portas USB 2.0, Cabeçalho GPIO de 40 pinos padrão Raspberry Pi, 2 × portas micro-HDMI, Porta de exibição MIPI DSI de 2 vias, Porta de câmera MIPI CSI de 2 vias, Áudio, estéreo de 4 pólos e porta de vídeo composto, H.265, H264, OpenGL ES 3.1, Vulkan 1.0, Slot para cartão Micro-SD para carregar sistema operacional e armazenamento de dados, 5 V DC via conector USB-C, 5 V DC via cabeçalho GPIO, Power over Ethernet (PoE) habilitado (requer PoE HAT separado).

5.2 Câmera

Foi utilizado a câmera de visão noturna para raspberry pi, com 5 megapixel ov5647 sensor e lente ajustável manualmente focal. O campo de visão é normal 60 graus (diagonal) com efeito de visão noturna através da luz do sensor infravermelho.



Figura 11 - Câmera infravermelho. Disponível: <<https://pt.aliexpress.com/item/32421714972.html?spm=a2g0s.9042311.0.0.4a90b90atzXktm>>. Acesso em: 18 de julho de 2021.

5.3 Sistema operacional

O sistema operacional escolhido foi o Raspberry Pi OS, devido a sua afinidade com o RASPBERRY PI. O sistema operacional foi desenvolvido pelos mesmos criadores do RASPBERRY PI, trazendo uma maior confiabilidade e suporte.

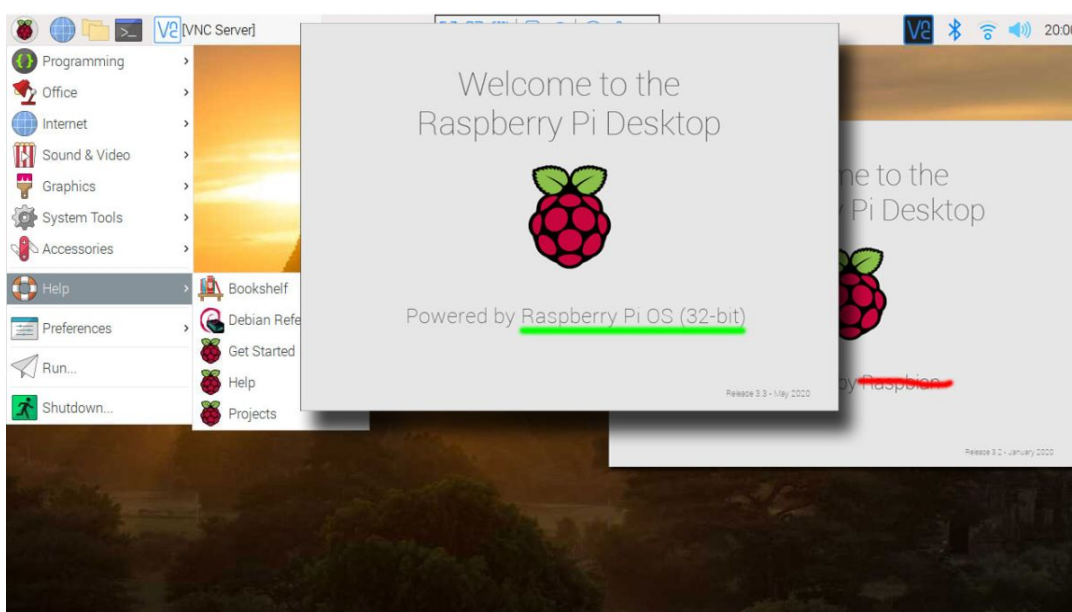


Figura 12 - Printscreen da interface do Raspberry Pi OS. Disponível: <<https://www.tomshardware.com/news/raspberry-pi-os-no-longer-raspbian>>. Acesso em: 18 de julho de 2021.

5.4 Buzzer Ativo

O buzzer ativo é um componente indicado para adicionar efeitos sonoros em projetos eletrônicos como alarmes, sistemas de sinalização, jogos, brinquedos, entre outros.

O buzzer do tipo ativo contém um circuito oscilador embutido, assim basta energizar o componente para que o mesmo comece a emitir um beep contínuo.



Figura 13 - Imagem de um buzzer ativo. Disponível: <<https://www.arduinoportugal.pt/qual-diferenca-buzzer-ativo-vs-buzzer-passivo/>>. Acesso em: 19 de julho de 2021.

5.5 Placa de ensaio

A placa de ensaio é uma placa com furos e conexões condutoras utilizada para a montagem de protótipos e projetos em estado inicial. A grande vantagem da placa de ensaio na montagem de circuitos eletrônicos é a facilidade de inserção de componentes, uma vez que não necessita de soldagem. As placas vão de 800 furos até 6000 furos, contendo conexões verticais e horizontais.

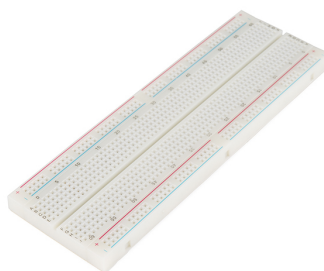


Figura 14 - Imagem de uma placa de ensaio. Disponível: <https://www.sparkfun.com/products/12615?_ga=2.254330376.1803911031.1626715704-1469670085.1626715704>. Acesso em: 19 de julho de 2021.

5.6 Capa de resfriamento para Raspberry PI

A capa de resfriamento é um ponto muito importante para a pesquisa, já que ela auxilia a manter a temperatura do RASPBERRY PI controlada. Nesta pesquisa, devido ao uso excessivo de recursos aplicados ao raspberry pi, o microcomputador esquenta demais travando-o. Portanto, sem ela a temperatura sobe muito e trava o dispositivo.

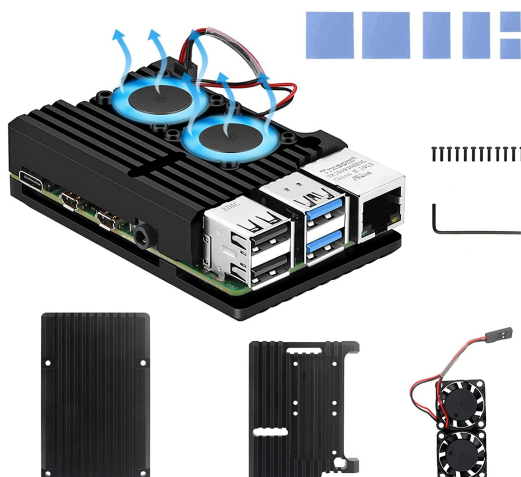


Figura 15 - Imagem de uma placa de resfriamento para raspberry pi. Disponível: <<https://www.amazon.com/Raspberry-Aluminum-Intelligent-Temperature-Protection/dp/B092QVXP6Z>>. Acesso em: 19 de julho de 2021.

6. RESULTADOS

Para verificar a eficácia do dispositivo desenvolvido, foi feito o teste do mecanismo em laboratório, devido às limitações da pesquisa, com cinco pessoas diferentes. A testagem ocorreu da seguinte forma, o indivíduo que participou sentou-se em uma cadeira, a qual estava em frente do software e simulou estado de sonolência através do fechamento de seus olhos, os testes ocorreram tanto com luz, quanto sem luz.

A figura 16 demonstra um dos usuários simulando o estado de sonolência. Pode ser observado na imagem que ele foi classificado como sonolento pois o tempo com os olhos fechados era de 0.72 segundos.

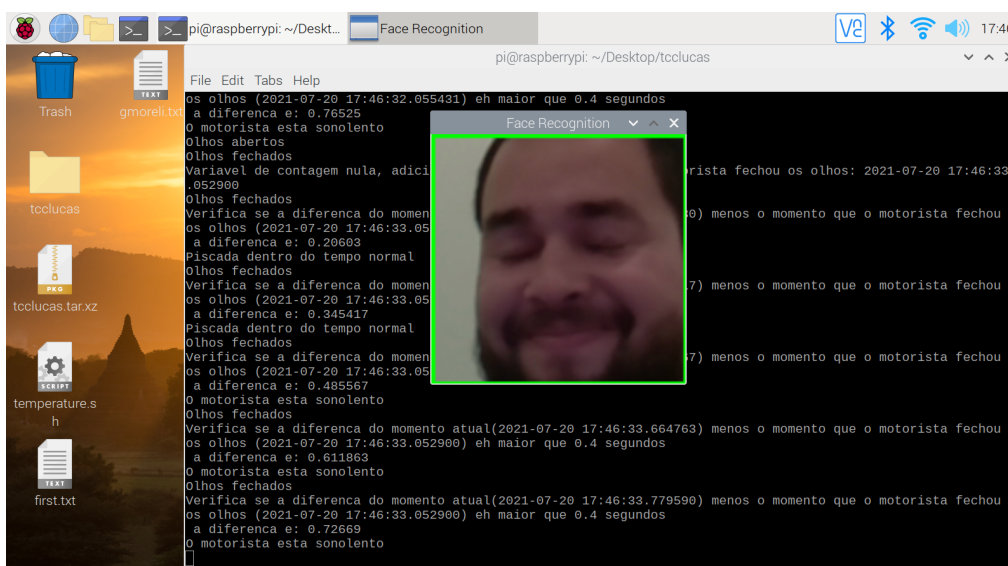


Figura 16 - Usuário 1 em estado sonolento.

Na figura 17, podemos observar o usuário sendo classificado corretamente com os olhos abertos.

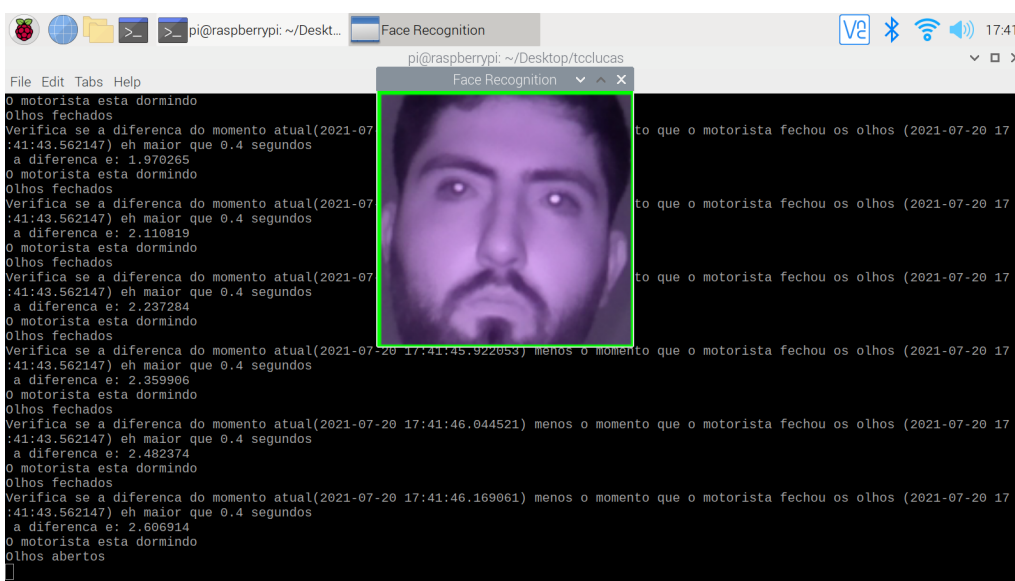


Figura 17 - Usuário 2 classificado com olhos abertos.

Na figura 18, podemos observar que mesmo usuário utilizando óculos com reflexo ainda foi possível classificá-lo corretamente.

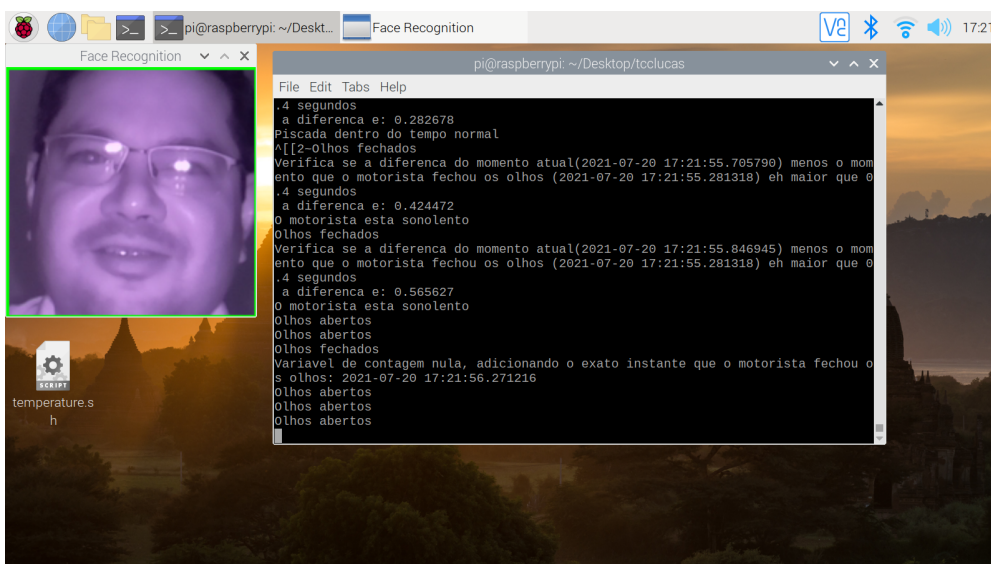


Figura 18 - Usuário 3 com reflexo nos óculos sendo classificado corretamente.

Na figura 19 podemos observar o usuário 4 com os olhos abertos, mas com a classificação como dormindo, isso se deve não a uma classificação errada, mas sim um atraso na classificação, o qual é demonstrado na tabela.

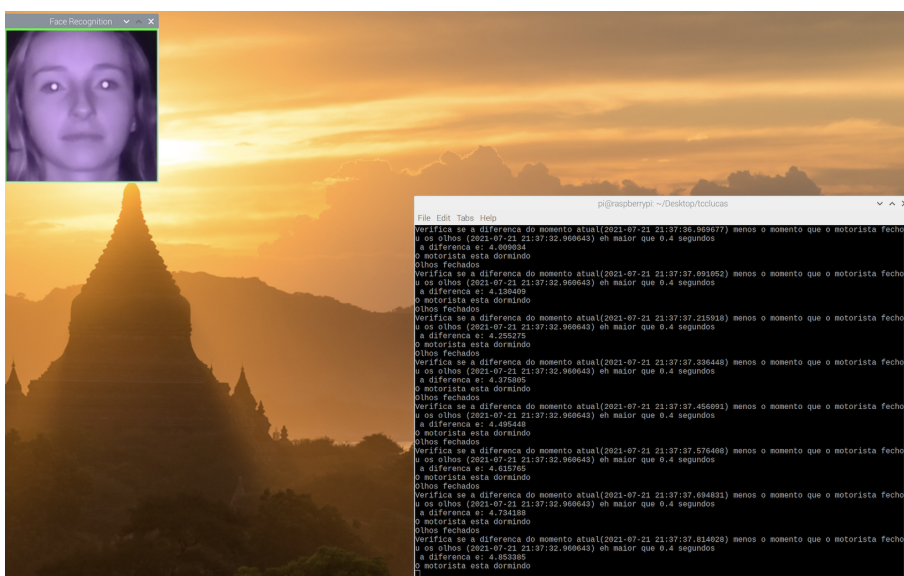


Figura 19 - Usuário 4 com atraso na classificação.

Na figura 20 podemos observar o usuário 5 simulando estar dormindo e o dispositivo classificando-o corretamente.

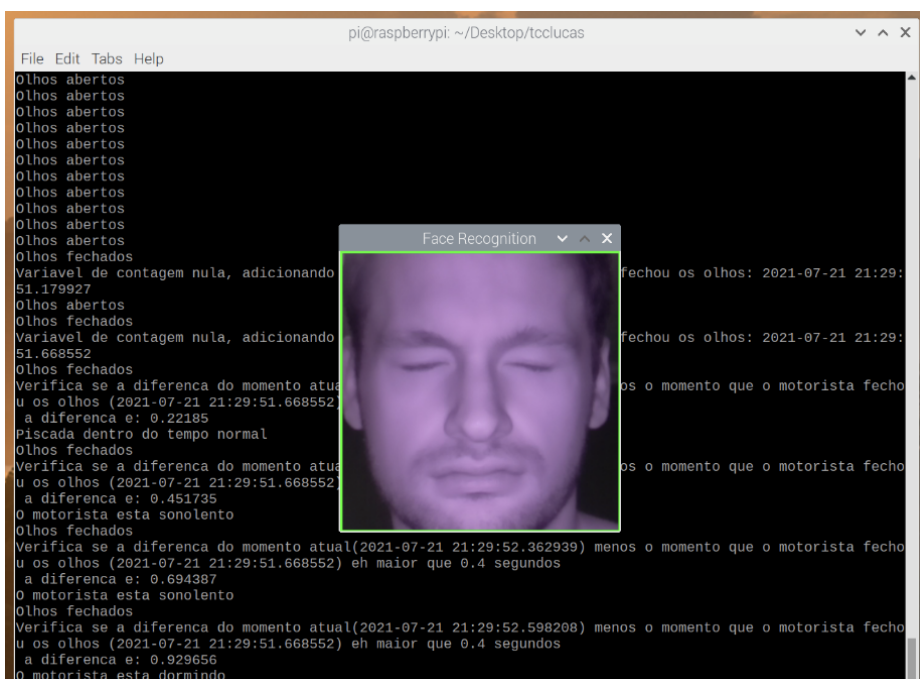


Figura 20 - Usuário 5 classificado como dormindo.

Um ponto muito importante observado nos testes foi um atraso no processamento das imagens pelo software, então com isso em mãos foi feito uma massa de testes para estipular a média de atraso desse processamento.

Na tabela abaixo conseguimos visualizar o atraso de processamento do dispositivo e a ocorrência destes atrasos.

Tempo	Quantidade
0.76	23
0.81	12
0.84	18
0.63	12

Tabela 2 - Tempo de atraso e ocorrências

Foi medido em 65 amostras o tempo que os resultados demoravam para ser processados, obtendo uma média (somatória da multiplicação do tempo pela quantidade, dividido pela quantidade total) de atraso na classificação em 0.77 segundos.

Também foi estipulado a porcentagem de confiabilidade do software através dos dados de classificações corretas e erradas, conforme a tabela abaixo.

Usuário	Classificações corretas	Classificações erradas
1	132	6
2	83	3
3	117	7
4	65	3
5	92	4

Tabela 3 - Acertos e erros do software

Segundo os testes, foram 704 acertos e 23 erros, do total de 727 amostras, utilizando a fórmula da taxa de acerto ($1 - (\text{erro total} / \text{total de amostras})$) isso resulta em uma precisão de 96.83% do dispositivo, o que podemos concluir em uma ótima taxa de precisão para a pesquisa.

7. PROJETOS FUTUROS

Um problema que existe na pesquisa atual, é que se o motorista não estiver com sono, mas por algum motivo, como ressecamento ocular por conta do ar condicionado, ele precisar piscar por um intervalo de tempo mais longo, o software irá reconhecer como estado de sonolência/dormindo, o que, teoricamente, seria uma categorização errada. Então, nesse caso, pode-se ver como um projeto futuro, poder ir armazenando o tempo da piscada do condutor, criando assim uma sequência. Já que o sono não é um estado que aparece repentinamente e sim é um estado progressivo, o piscar de olhos também e, dessa forma, ele vai progredindo o tempo até se encaixar em sonolência.

À vista disso, é sabido que criando um conjunto de dados contendo o tempo da piscada do condutor, seria possível categorizar se aquela piscada que levou um pouco mais de tempo foi causada por um ato de sonolência ou não, criando assim apenas alarmes que façam sentido, e não causando disparos de alarmes em um momento em que o motorista se encontre em estado de alerta e por algum motivo demorou um pouco mais na piscada.

8. CONSIDERAÇÕES FINAIS

A pesquisa conseguiu responder a pergunta que norteou e também todo o trabalho feito (É possível gerar um dispositivo sinalizador de sonolência para condutor de veículos?). A gestação de dispositivo sinalizador de sonolência para condutor de veículos é possível através de tecnologias e equipamentos atuais, utilizando linguagem de programação adequada e técnicas apropriadas.

Além disso, o trabalho apresentado atende os objetivos propostos visto o sucesso em seu reconhecimento tanto com luz solar quanto sem nenhuma luz, então o uso do Raspberry PI trouxe uma ótima eficiência. Para mais, o custo do projeto também foi um ponto de sucesso, visto que teve um custo acessível, trazendo assim uma chance de implementação mais viável.

Para finalizar, é interessante lembrar que esta pesquisa possibilitou a aplicação de conceitos aprendidos durante o curso de graduação, pois aquilo que era visto como teoria na universidade pode aqui ser colocado em prática.

REFERÊNCIAS

BROWNLEE, Jason. **A Gentle Introduction to Computer Vision**. Machine Learning Mastery. Disponível em:

<<https://machinelearningmastery.com/what-is-computer-vision/>>.

DANISMAN, Taner; BILASCO, Ian M; DJERABA, Chabane; *et al.* **Drowsy driver detection system using eye blink patterns**. 2010 International Conference on Machine and Web Intelligence, 2010.

Drivers are Falling Asleep Behind the Wheel. National Safety Council. Disponível em: <<https://www.nsc.org/road-safety/safety-topics/fatigued-driving>>.

Image Processing and Computer Vision. ofBook - Image Processing and Computer Vision. Disponível em:

<https://openframeworks.cc/ofBook/chapters/image_processing_computer_vision.html>.

LEE, Somet. **Understanding Face Detection with the Viola-Jones Object Detection Framework**. Medium. Disponível em:

<<https://towardsdatascience.com/understanding-face-detection-with-the-viola-jones-object-detection-framework-c55cc2a9da14>>.

RASPBERRY PI. **Teach, Learn, and Make with Raspberry Pi**. Raspberry Pi. Disponível em: <<https://www.raspberrypi.org/>>.

TOBIAS, Simone. **Os perigos (e o teste de avaliação) do sono ao volante**. Quatro Rodas. Disponível em:

<<https://quatorrodas.abril.com.br/auto-servico/os-perigos-e-o-teste-de-avaliacao-do-sono-ao-volante/>>.

VIOLA, P. and JONES, M. **Robust real-time face detection**. Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001,

Welcome to Python.org. Python.org. Disponível em: <<https://www.python.org/>>.