UNIVERSIDADE DA INTEGRAÇÃO INTERNACIONAL DA LUSOFONIA AFRO-BRASILEIRA

INTITUTO DE ENGENHARIAS E DESENVOLVIMENTO SUSTENTÁVEL - IEDS

CURSO DE GRADUAÇÃO EM ENGENHARIA DE COMPUTAÇÃO

JOSÉ LUCAS DA SILVA PINHEIRO

DETECÇÃO DE SONOLÊNCIA AO VOLANTE: UMA SOLUÇÃO PRÁTICA COM VISÃO COMPUTACIONAL E RASPBERRY PI

REDENÇÃO - CE

2024

JOSÉ LUCAS DA SILVA PINHEIRO

DETECÇÃO DE SONOLÊNCIA AO VOLANTE: UMA SOLUÇÃO PRÁTICA COM VISÃO COMPUTACIONAL E RASPBERRY PI

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Intituto de Engenharias e Desenvolvimento Sustentável - IEDS da Universidade da Integração Internacional da Lusofonia Afro-Brasileira, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Vandilberto Pereira Pinto

REDENÇÃO - CE

2024

Universidade da Integração Internacional da Lusofonia Afro-Brasileira
Sistema de Bibliotecas da UNILAB
Catalogação de Publicação na Fonte.

Pinheiro, José Lucas da Silva.

P654d

Detecção de sonolência ao volante: Uma solução prática com Visão Computacional e Raspberry PI / José Lucas da Silva Pinheiro. - Redenção, 2024.
49f: il.

Monografia - Curso de Engenharia De Computação, Instituto De Engenharias E Desenvolvimento Sustentável, Universidade da Integração Internacional da Lusofonia Afro-Brasileira, Redenção, 2024.

Orientador: Prof. Dr. Vandilberto Pereira Pinto.

1. Trânsito. 2. Sono. 3. Visão Computacional. 4. MediaPipe. I. Título

CE/UF/BSCA CDD 363.1251

JOSÉ LUCAS DA SILVA PINHEIRO

DETECÇÃO DE SONOLÊNCIA AO VOLANTE: UMA SOLUÇÃO PRÁTICA COM VISÃO COMPUTACIONAL E RASPBERRY PI

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Computação do Intituto de Engenharias e Desenvolvimento Sustentável - IEDS da Universidade da Integração Internacional da Lusofonia Afro-Brasileira, como requisito parcial à obtenção do grau de bacharel em Engenharia de Computação.

Data de aprovação: 03/12/2024.

BANCA EXAMINADORA

Prof. Dr. Vandilberto Pereira Pinto (Orientador)
Universidade da Integração Internacional da Lusofonia Afro-Brasileira - UNILAB

Profa. Dra. Lígia Maria Carvalho Sousa
Universidade da Integração Internacional daLusofonia Afro-Brasileira - UNILAB

Prof. Dr. Herivelton Alves de Oliveira
Universidade da Integração Internacional daLusofonia Afro-Brasileira - UNILAB

AGRADECIMENTOS

Primeiramente, agradeço a Deus, pela força, sabedoria e por ter me guiado ao longo de toda essa jornada. Agradeço por manter minha fé firme, mesmo nos momentos mais desafiadores, e por me permitir chegar até aqui.

Aos meus pais, Liduino e Elda, pelo amor incondicional, apoio constante e pelos ensinamentos que moldaram minha vida. Vocês sempre foram minha base e fonte de inspiração. Sem vocês, nada disso seria possível.

À minha namorada, Iris Paulino, por estar sempre ao meu lado, me incentivando e compreendendo as dificuldades de toda a jornada. Sua presença foi fundamental para superar as dificuldades durante essa trajetória.

Ao meu orientador, Prof. Dr. Vandilberto Pereira Pinto, pela confiança depositada em mim, pela orientação precisa e pelo comprometimento em estar sempre disponível para auxiliar em todas as etapas deste trabalho.

Ao grupo de pesquisa de Processamento e Gerenciamento de Energias Renováveis e Controle (PGERC) e a todos os seus membros, com destaque para João Pedro, Leonardo Sousa e Adriano Barroso, cuja participação direta no desenvolvimento deste projeto foi essencial. O apoio, as trocas de conhecimento e as contribuições técnicas de vocês foram indispensáveis para alcançar os resultados apresentados.

Por fim, agradeço aos meus amigos, em especial Cleilton Sousa, João Matheus e Ítalo Guilherme, que estiveram ao meu lado durante toda a jornada. Cada gesto e palavra de motivação fizeram a diferença para que eu pudesse chegar até aqui.

RESUMO

O uso de ferramentas de Inteligência Artificial, como Visão Computacional e Aprendizado de Máquina, está cada vez mais presente em diversas áreas, incluindo o trânsito, onde o alto índice de acidentes e a necessidade de coleta de dados para análise impulsionam o desenvolvimento de soluções tecnológicas. Essas ferramentas têm potencial para atuar desde a análise do tráfego até a identificação de infratores e a prevenção de acidentes, contribuindo para a segurança viária. Neste contexto, o presente trabalho visa desenvolver um sistema para detecção de sonolência ao volante, utilizando técnicas de Visão Computacional com o *framework* MediaPipe. O projeto resultou no desenvolvimento de um protótipo composto por uma placa Raspberry Pi 3B, uma PiCamera, LEDs indicadores e um buzzer que emite alertas sonoros ao detectar que o motorista está com os olhos fechados, indicando que ele pode estar dormindo ao volante. Esse protótipo oferece uma alternativa acessível e compacta para minimizar o risco de acidentes relacionados à fadiga e à sonolência ao volante, integrando facilmente componentes de baixo custo e consumo energético. Ao ajudar a reduzir as estatísticas de acidentes e promover um trânsito mais seguro, o sistema contribui para o avanço das tecnologias assistivas no setor automobilístico, com potencial para futuras melhorias e integrações em veículos modernos.

Palavras-chave: Trânsito. Sono. Visão Computacional. MediaPipe. Detecção. Protótipo.

ABSTRACT

The use of Artificial Intelligence tools, such as Computer Vision and Machine Learning, is increasingly present in various fields, including traffic management, where the high rate of accidents and the need for data collection for analysis drive the development of technological solutions. These tools have the potential to support tasks ranging from traffic analysis to the identification of offenders and accident prevention, thereby contributing to road safety. In this context, the present work aims to develop a system for detecting driver drowsiness using Computer Vision techniques with the MediaPipe framework. The project led to the development of a prototype consisting of a Raspberry Pi 3B board, a PiCamera, indicator LEDs, and a buzzer that emits sound alerts when it detects that the driver has closed eyes, indicating the possibility of falling asleep at the wheel. This prototype offers an affordable and compact alternative to minimize the risk of accidents related to fatigue and drowsiness while driving, integrating low-cost and energy-efficient components. By helping reduce accident statistics and promoting safer traffic conditions, the system contributes to the advancement of assistive technologies in the automotive sector, with potential for future enhancements and integrations into modern vehicles.

Keywords: Traffic. Sleep. Computer Vision. MediaPipe. Detection. Prototype.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE CÓDIGOS-FONTE

SUMÁRIO

1 INTRODUÇÃO

Atualmente, pode-se afirmar que acidentes de trânsito é uma das principais causas de morte no país. De acordo com a pesquisa realizada pelo Ministério da Saúde em 2021, houve aproximadamente 35.032 mortes causadas por acidentes de trânsito (GLOBAL, 2023). Fora as perdas de vidas, há também um custo social (ferimentos, invalidez) e um custo financeiro (remoção e conserto de veículos, internação em hospitais), que afetam a economia do país. Uma pesquisa realizada pelo Instituto de Pesquisa Econômica Aplicada (Ipea) e pela Associação Nacional de Transportes Públicos (ANTP) na década passada, mostra que o prejuízo é de cerca de R$ 50 bilhões por ano (GLOBAL, 2023).

A sonolência no volante está entre as três principais causas de acidentes. De acordo com a Associação Brasileira de Medicina de Tráfego (ABRAMET), o sono e fadiga são a 3ª maior causa de acidentes no trânsito no país, ficando atrás apenas do uso de álcool ao volante e excesso de velocidades (SOUSA, 2023).

Em contrapartida, há a Inteligência Artificial (IA), que desde o seu surgimento da década de 1950, sofre constantemente melhorias. Koenigkam-Santos *et al.* (2019), explica que hoje em dia, esse constante avanço se dá, principalmente, por três fatores: (i) o baixo custo de processamento e de memória; (ii) o surgimento de novos modelos, como as Redes Neurais Profundas; (iii) e à uma grande quantidade de dados disponível na internet.

Esse avanço, permite que técnicas, como o Aprendizado de Máquina e a Visão Computacional estejam cada vez mais presentes em nosso cotidiano. As aplicações dessas ferramentas estão presentes nas mais diversas áreas, como: medicina, agropecuária, indústria e logística. E no trânsito não seria diferente, seja voltada para a análise de tráfego ou para soluções de segurança.

No trânsito, a Visão Computacional é uma forte aliada na detecção de infrações, como violadores de sinais de trânsito por exemplo, e na identificação e análise do fluxo de tráfego através de sistemas inteligentes.

A Visão Computacional pode ser estabelecida como uma tecnologia que utiliza Inteligência Artificial e permite que os computadores obtenham dados a partir de entradas visuais, como fotos e vídeos. Dados esses que, posteriormente, são utilizados para as mais diversas aplicações (CHAVES, 2022).

Niero *et al.* (2018) estudaram uma das aplicações dessa ferramenta para o trânsito, que foi a utilização de técnicas de Visão Computacional para realizar a coleta de dados referentes

ao fluxo de veículos em um determinado cruzamento. Os autores aplicaram um algoritmo para a detecção de veículos na imagem através de uma câmera localizada próximo ao semáforo e posteriormente processadas em um computador dedicado.

Outra tecnologia que auxilia bastante aos problemas citados e que está cada vez mais sendo utilizada na sociedade é o Aprendizado de Máquina (do inglês, *Machine Learning* - ML). Se trata de uma área da Inteligência Artificial focada em utilizar dados e algoritmos para simular o aprendizado humano.

Matheus e Salina (2019) desenvolveram um sistema para detecção e reconhecimento de sinalizações de trânsito. Para a detecção, os autores optaram por realizar segmentação por cores e seleção da área de interesse. Já para a segunda parte do trabalho, o reconhecimento, foram utilizados algoritmos de *Machine Learning*, para a classificação das áreas obtidas no processo anterior.

O uso de ferramentas de Visão Computacional e *Machine Learning* em ambientes de trânsito tem sido amplamente estudado na literatura. Dessa forma, o objetivo deste trabalho é aplicar essas tecnologias para desenvolver um sistema de detecção de sonolência do motorista no trânsito, acionando um sinal sonoro como alerta para manter o mesmo acordado durante a condução.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Desenvolver um sistema para auxílio de detecção de sono ao volante, empregando técnicas de Visão Computacional para identificar sinais de sonolência e emitir alertas sonoros para o usuário.

1.1.2 Objetivos Específicos

- Realizar uma análise na literatura sobre Visão Computacional e trabalhos relacionados ao tema;
- Desenvolver um protótipo para detectar o fechamento dos olhos do motorista e emitir um alerta sonoro preventivo;
- Realizar testes em ambientes controlados e reais, com o objetivo de validar a eficácia do projeto.

1.2 Organização do Trabalho

O presente trabalho está estruturado em seis capítulos, conforme descrito a seguir.

- **Capitulo 1**: apresenta a introdução do trabalho e a descrição dos objetivos gerais e específicos.
- **Capitulo 2**: apresenta o referencial teórico, contextualizando as principais ferramentas e tecnologias relacionadas ao conteúdo abordado.
- **Capitulo 3**: apresenta alguns dos mais diversos trabalhos relacionados ao tema da pesquisa.
- **Capitulo 4**: traz, de forma detalhada, todo o processo metodológico que foi seguido para o desenvolvimento do trabalho.
- **Capitulo 5**: expõe os resultados obtidos no desenvolvimento do trabalho.
- **Capitulo 6**: apresenta as conclusões e questões que podem ser exploradas futuramente.

2 REFERENCIAL TEÓRICO

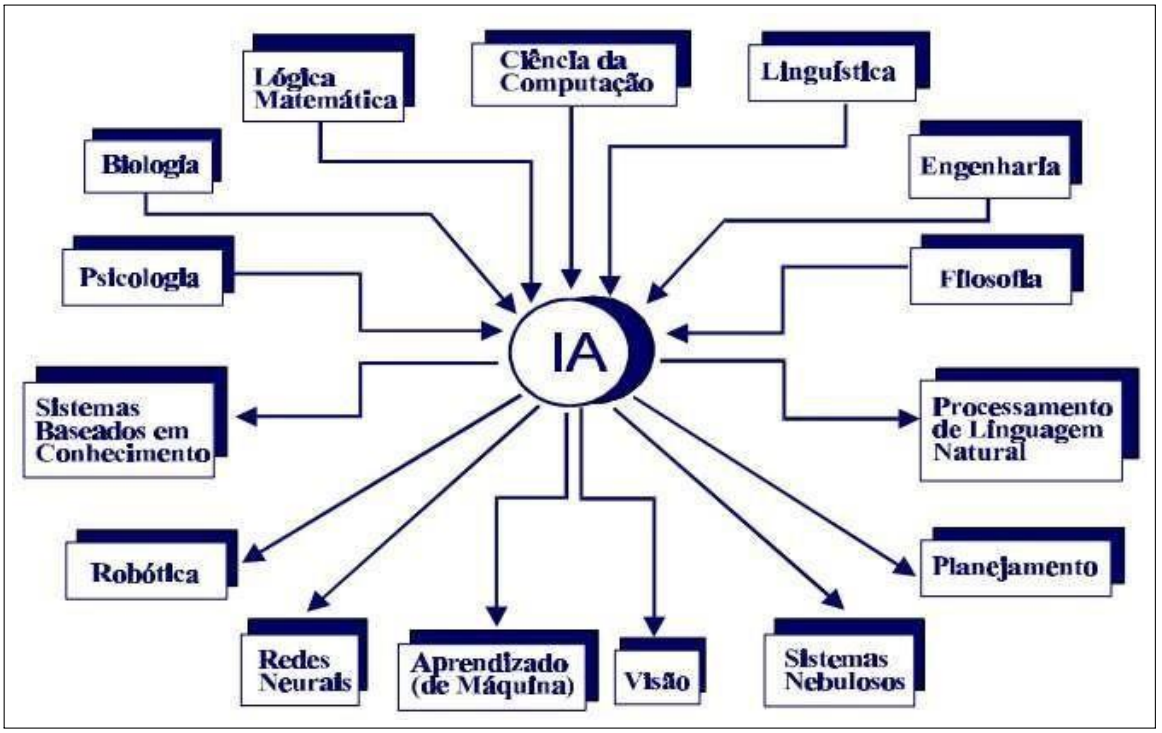
Esta Seção se destina a apresentar o resultado do levantamento bibliográfico realizado sobre os principais temas e ferramentas utilizadas no trabalho.

2.1 Inteligência Artificial

De acordo com Fernandes (2003), o termo inteligência vem do latim que se divide em inter (entre) e legere (escolher), ou seja, é a possibilidade de o homem escolher entre uma coisa ou outra. Dessa forma, considera-se inteligência artificial um tipo de inteligência desenvolvida pelo homem para ser utilizada pelas as máquinas de forma que simule sua inteligência natural.

Outra definição semelhante é apresentada por MONARD e BARANAUKAS (2000), no qual diz que a Inteligência Artificial é um ramo da Ciência da Computação que possui o objetivo de fazer os computadores pensarem e se comportarem de forma inteligente. O autor faz uma observação também sobre o fato da Inteligência Artificial não ser um ramo exclusivo da Ciência da Computação, como mostra a Figura 1.

Figura 1 – Áreas relacionadas com IA

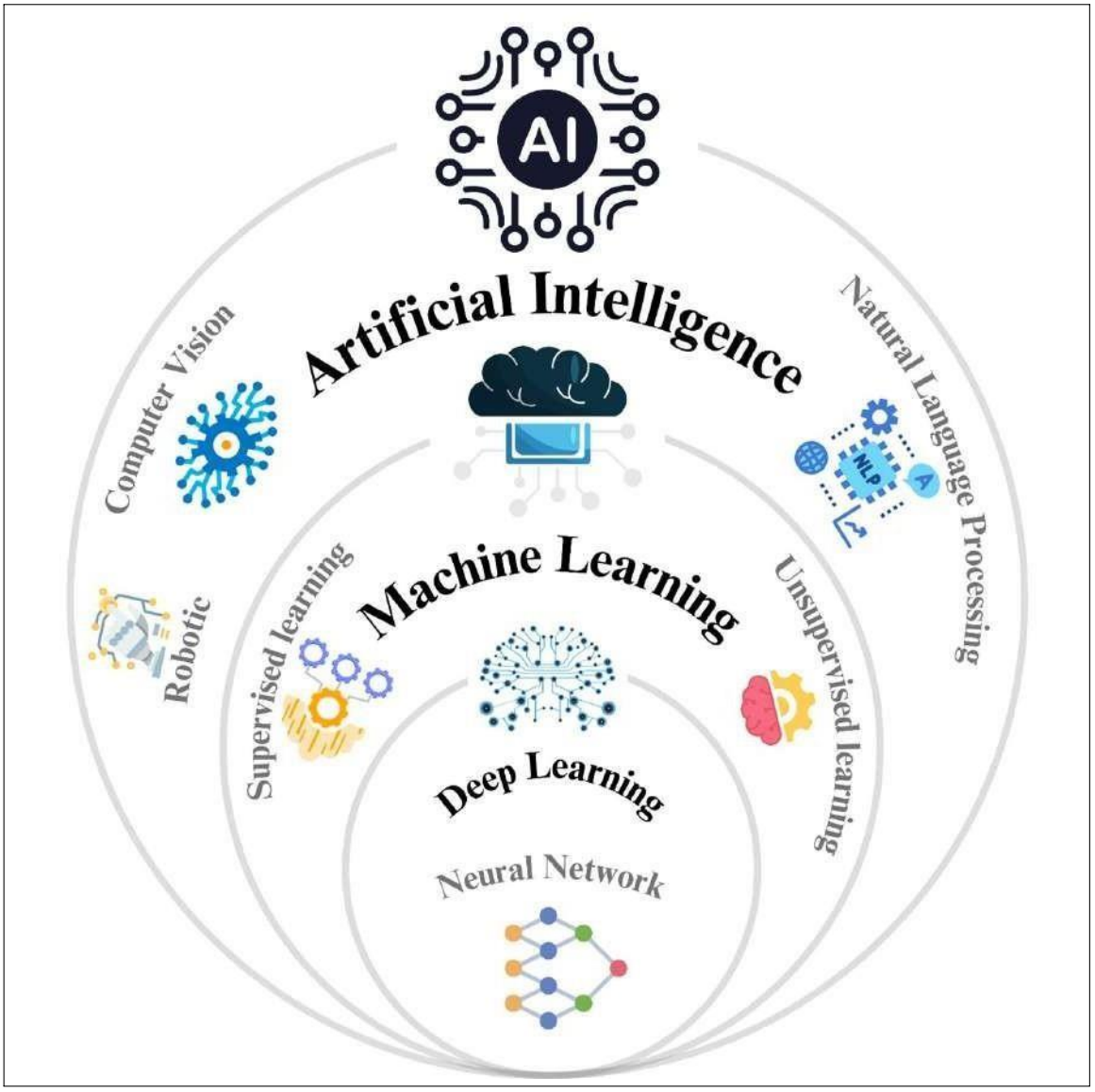


Fonte: (MONARD; BARANAUKAS, 2000)

Dentro do amplo campo da Inteligência Artificial, é possível separar em dois sub-campos: o Aprendizado de Máquina e o Aprendizado Profundo. Cada um deles compostos por

outras subáreas, como mostra a Figura 2.

Figura 2 – Diagrama dos subconjuntos da IA



Fonte: (SALES, 2023)

2.2 Aprendizado de Máquina

O Aprendizado de Máquina, ou *Machine Learning*, é uma subárea da Inteligência Artificial cujo objetivo é simular o aprendizado humano através do uso de dados e algoritmos. Segundo Mitchell (1997), é um aprendizado por experiência, que de acordo com a execução da tarefa, o problema aprende o método mais eficaz de resolver.

2.2.1 Tipos de aprendizado de máquina

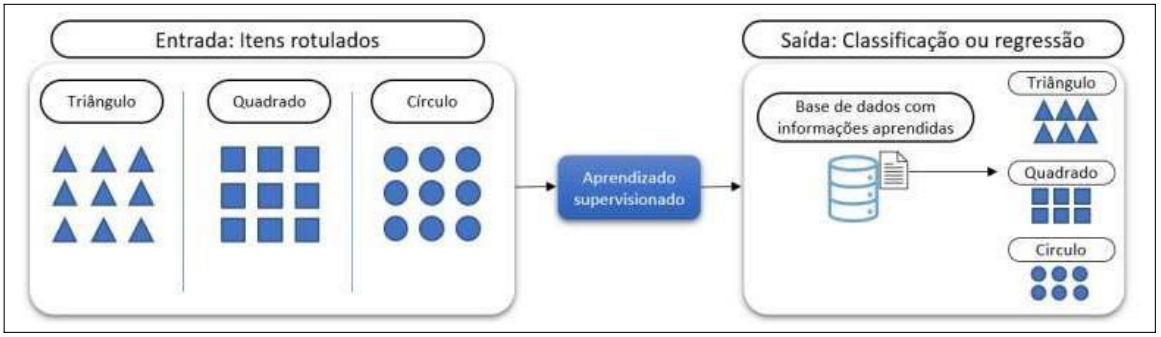
Em geral, é possível dividir o aprendizado de máquina em dois subcampos principais: aprendizado supervisionado e aprendizado não supervisionado.

2.2.1.1 Aprendizado supervisionado

No aprendizado supervisionado é disponibilizado ao sistema um conjunto de dados, onde cada um possui um rótulo associado a ele. Esse rótulo define a classe a qual o dado pertence (BATISTA, 2003). Esse processo de atribuir um rótulo para cada dado, é chamado de classificação. Esses dados posteriormente são submetidos ao algoritmo que será capaz de aprender e identificar elementos de cada grupo (CORTEZ, 2022).

A Figura 3 apresenta o funcionamento genérico do algoritmo de aprendizado supervisionado. É possível observar que na entrada, os itens já estão separados de acordo com suas classes. E na saída a máquina é capaz de identificar e classificar cada item de acordo com suas classes.

Figura 3 – Funcionamento do algoritmo de aprendizado supervisionado



Fonte: (CORTEZ, 2022)

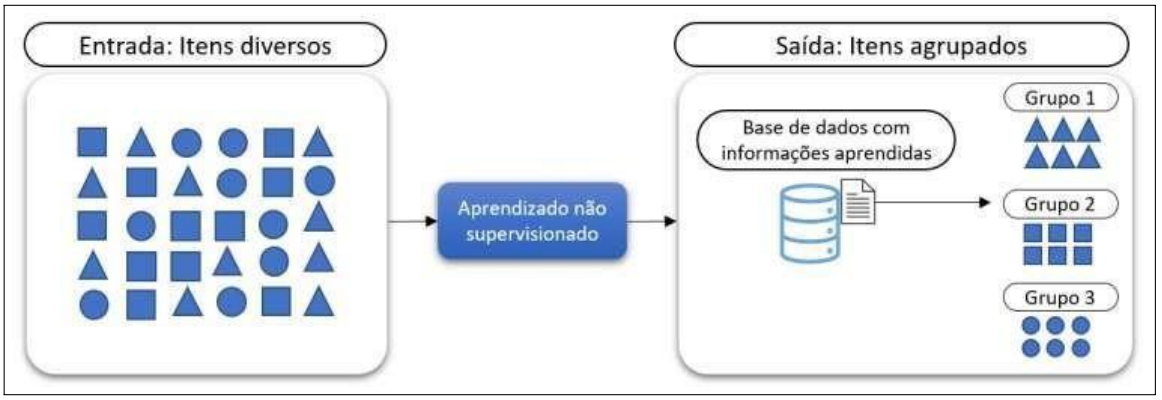
2.2.1.2 Aprendizado não supervisionado

Nesse tipo de aprendizado, os dados são inseridos no sistema sem nenhuma identificação ou classificação. E a partir disso, o algoritmo deve realizar a detecção de tendências nos dados (CORTEZ, 2022). O objetivo desse tipo de algoritmo é produzir um modelo que realize uma busca por regularidades e semelhanças entres os dados, de forma que seja possível agrupá-los e classificá-los (BATISTA, 2003).

Na Figura 4 é possível observar o funcionamento desse algoritmo. Onde a entrada é composta por dados diversos, sem classificação, e na saída a máquina realiza o agrupamento dos

mesmos a partir de características semelhantes.

Figura 4 – Funcionamento do algoritmo de aprendizado não supervisionado



Fonte: (CORTEZ, 2022)

2.3 Visão Computacional

Utilizando como referência a visão dos seres humanos, a Visão Computacional é a técnica capaz de fazer a máquina aprender e identificar componentes em uma determinada foto ou vídeo (CORTEZ, 2022).

Segundo Ballard e Brown (1982), a Visão Computacional é a ciência que estuda e incrementa tecnologias que permitem às máquinas a capacidade de extrair características de imagens capturadas por diferentes tipos de dispositivos.

Outra definição semelhante é apresentada por Chaves (2023), no qual diz que a Visão Computacional é um ramo da Inteligência Artificial que se utiliza de um conjunto de técnicas e modelos matemáticos para capturar dados a partir mídias digitais, como imagens e vídeos.

Dentro do ramo de Visão Computacional, é possível encontrar algumas ferramentas que auxiliam a execução e implementação de algoritmos voltados para essa área, como o OpenCV e a linguagem de programação Python.

2.3.1 OpenCV

O OpenCV (*Open Source Computer Vision Library*), é uma biblioteca de código aberto que possui mais de 2500 algoritmos otimizados de Visão Computacional e Aprendizado de Máquina. Essa ferramenta pode ser utilizada para diversas aplicações, como: captura e processamento de vídeos em tempo real, detecção e reconhecimento de rostos, identificação

de objetos, rastreamento de movimentos e objetos através de camêras, entre outros (OPENCV, 2024).

2.3.2 Python

O Python é uma linguagem de programação de uso geral, alto nível, tipagem dinâmica e orientada a objetos, mas também suporta outros paradigmas (PYTHON, 2024). Devido ao seu suporte às mais variadas bibliotecas e ferramentas, ela é uma das linguagens mais utilizadas em projetos que envolvem aprendizado de máquina e criação de algoritmos complexos.

2.4 MediaPipe

O MediaPipe é um *framework* de código aberto desenvolvido pela Google que é capaz de detectar elementos em imagens e vídeos. Ele foi projetado para pesquisadores, estudantes e desenvolvedores que trabalham com implementação de aplicações de *Machine Learning* (LUGARESI *et al.*, 2019).

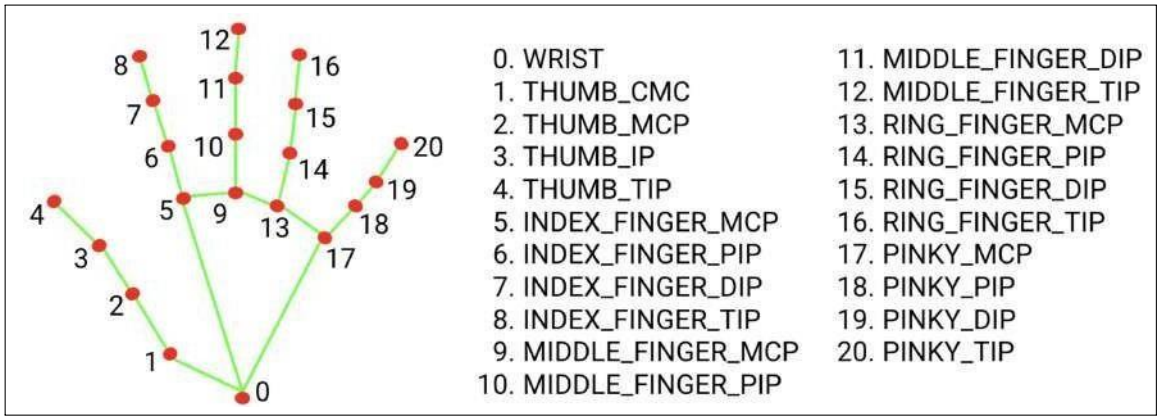
Essa ferramenta oferece diversas soluções, como: detecção de objetos, classificação e segmentação de imagem, reconhecimento de gestos, detecção de mãos, dentre outras. Vale ressaltar que cada solução disponibilizada pelo MediaPipe se utiliza de um ou mais algoritmos de *Machine Learning* para o seu funcionamento e elas podem ser utilizadas em diferentes plataformas, como web, mobile e computadores (MEDIAPIPE, 2024).

Abaixo será apresentado algumas dessas soluções e suas principais características.

2.4.1 Hand Landmarker

O conjunto de modelos de pontos de referência da mão identifica a posição de 21 pontos nas articulações da mão dentro das áreas detectadas da mão. Esse modelo foi treinado com cerca de 30 mil imagens reais, além de vários modelos sintéticos de mãos renderizadas sobre diferentes fundos (MEDIAPIPE, 2024).

Figura 5 – Modelo Hand Landmarks

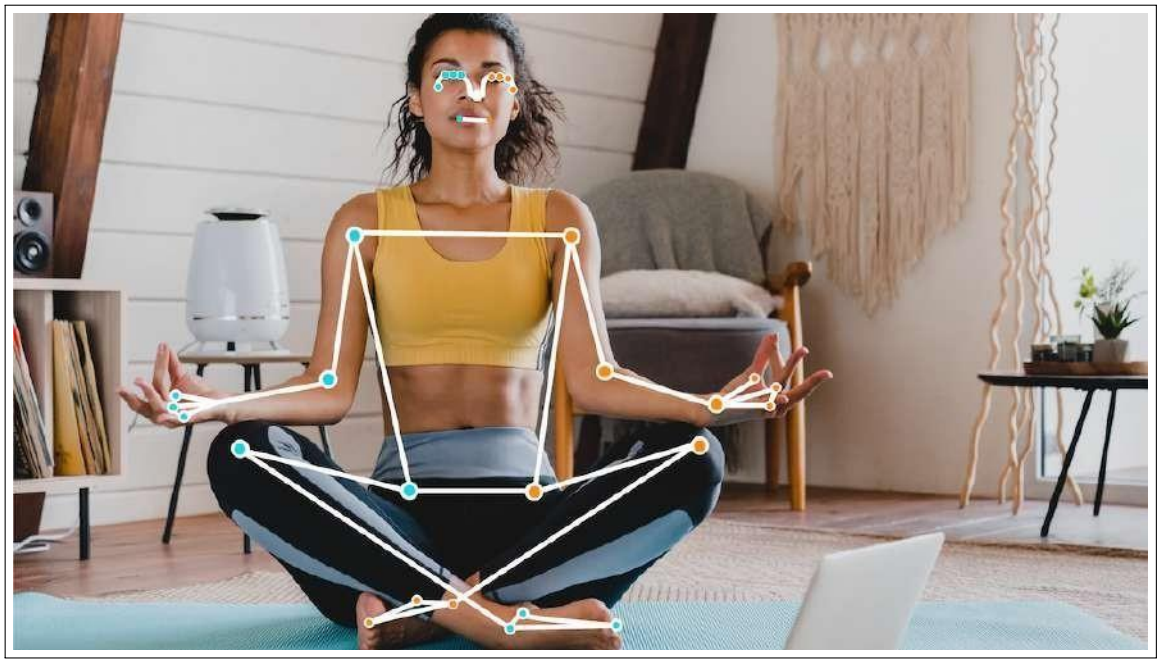


Fonte: (MEDIAPIPE, 2024)

2.4.2 Pose Landmarker

O *Pose Landmarker* utiliza uma série de modelos para prever pontos de referência de poses. O primeiro modelo identifica a presença de corpos humanos em uma imagem, enquanto o segundo oferece um mapeamento completo, fornecendo cerca de 33 pontos de referência em 3 dimensões.

Figura 6 – Modelo Pose Landmarks

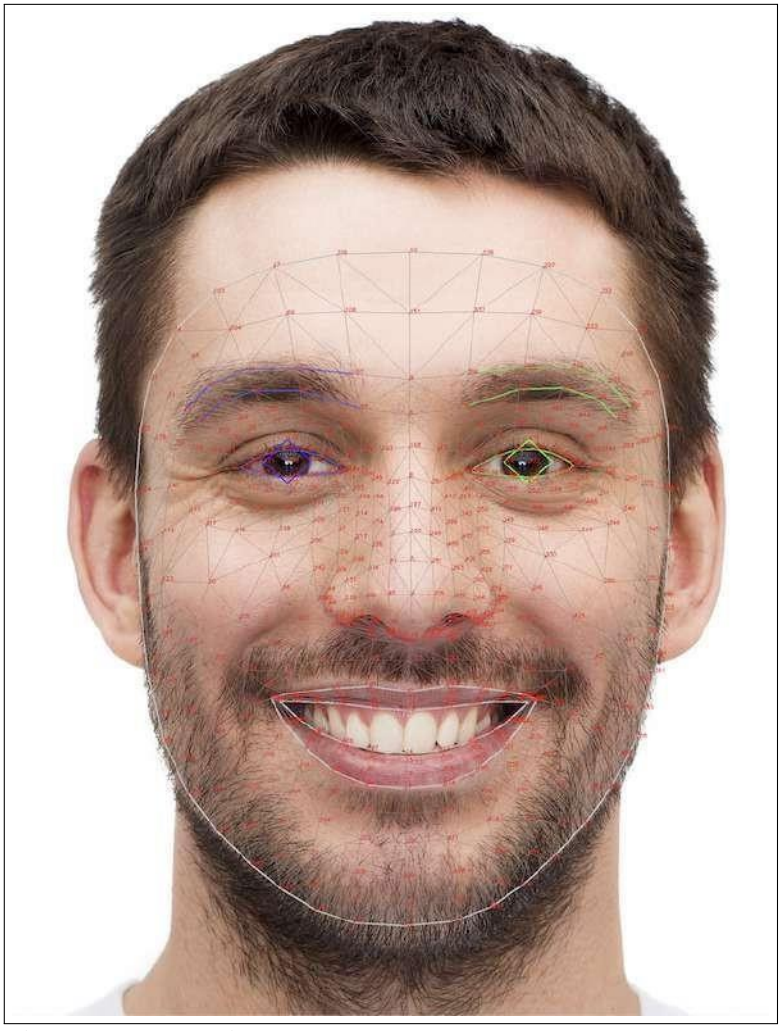


Fonte: (MEDIAPIPE, 2024)

2.4.3 Face Landmarker

Para o projeto desenvolvido, foi utilizado o modelo de detecção de pontos na face, chamado *ace Landmarker*. Este é composto por três algoritmos de *Machine Learning*: o primeiro detecta rostos, o segundo localiza 478 pontos de referência e o terceiro identifica características e expressões faciais.

Figura 7 – Modelo Face Landmarker e pontos de referência.



Fonte: (MEDIAPIPE, 2024)

2.5 Sistemas Embarcados

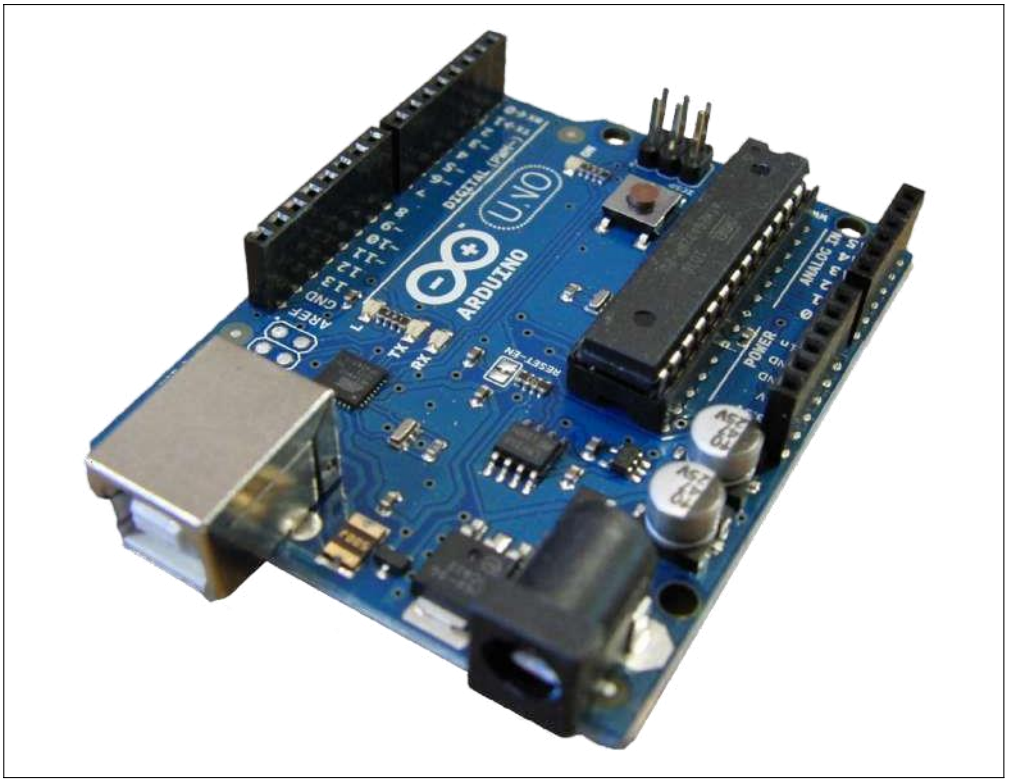
Segundo Souza (2022), um sistema embarcado integra hardware e software projetado para uma tarefa específica. Esses sistemas, geralmente são desenvolvidos com um conjunto limitados de componentes. Dentre esses componentes, há duas placas que são bastantes utilizadas para o desenvolvimento de sistemas embarcados: a plataforma Arduino e o Raspberry Pi.

2.5.1 Arduino

Segundo informações do site, Arduino (2024), ela é uma plataforma open-source caracterizada por sua eletrônica flexível, com hardware e software de fácil utilização. Devido à essas características alinhadas ao seu baixo custo, ela é uma das principais placas de desenvolvimento utilizadas para projetos de prototipagem e sistemas embarcados.

Dentre tantos modelos e versões existentes dessa pltaforma, o Arduino UNO foi o escolhido para a utilização no desenvolvimento do protótipo.

Figura 8 – Arduino UNO.



Fonte: (ARDUINO, 2024)

2.5.2 Raspberry Pi

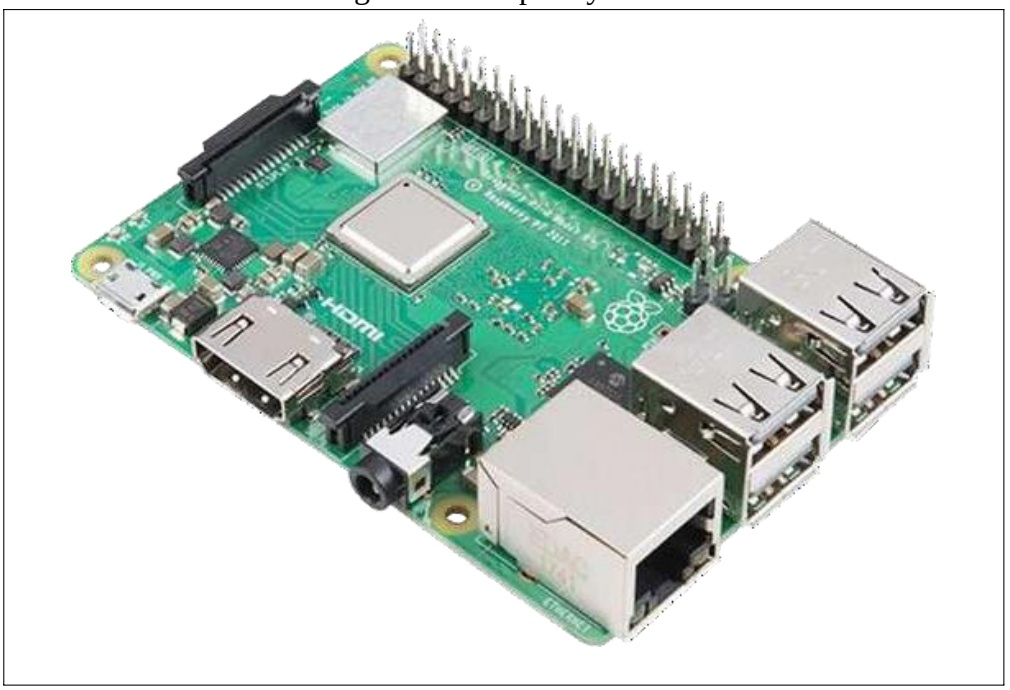
Essa plataforma foi desenvolvida pela Raspberry Pi Foundation, uma organização do Reino Unido sem fins lucrativos, com o objetivo de disponibilizar uma alternativa de baixo custo e bom desempenho para os mais diversos tipos de projetos (MARTINS, 2022).

Segundo Gogoni (2019), o Raspberry Pi é considerado um microcomputador, pois possui todos os componentes básicos de um computador convecional, como: processador, memória RAM e portas de entrada/saída. Com isso, pode ser utilizada com os periférios básicos

de um computador: mouse, teclado e monitor. Além disso, também possui entrada de cartão microSD para instalação de sistema operacional.

O Raspberry Pi possui diversos modelos e versões desde o seu lançamento. Dentre eles, foi escolhida para o desenvolvimento do projeto o modelo Pi 3B, por ser o mais acessível no momento. As especificações técnicas desse modelo incluem: Processador Broadcom BCM2837 64bits ARMv8 Cortex-A53 Quad-Core; Memória RAM de 1GB; Adaptador Wi-fi 802.11n integrado; Bluetooth 4.1 BLE integrado; Conector de vídeo HDMI; 4 portas USB 2.0; Conector Ethernet; GPIO de 40 pinos; e Interface para câmera (CSI). Além disso, possui suporte para diversas linguagens de programação, dentre elas o Python, que é a linguagem principal no desenvolvimento do projeto.

Figura 9 – Raspberry Pi 3B.



Fonte: (RASPBERRY, 2024)

3 TRABALHOS RELACIONADOS

3.1 Aplicações da Inteligência Artificial e Visão Computacional na mobilidade urbana

Aguiar *et al.* (2021) realizaram uma pesquisa com o objetivo de auxiliar os orgãos de trânsito e de saúde na prevenção de acidentes de trânsito através do uso de tecnologias como *Big Data* e algoritmos de Inteligência Artificial. Para alcançar esse objetivo, os autores realizaram a análise de dados extraídos do Observatório de Segurança Viária de Fortaleza, abrangendo o período de 2015 a 2018.A partir desses dados, foi aplicada uma metodologia quantitativa que envolveu uma série de etapas, como a coleta, organização e transformação dos dados, além da aplicação de algoritmos de Redes Neurais. Os resultados da pesquisa mostraram uma redução de 18,4% nos acidentes com feridos e 14,6% nos acidentes com vítimas fatais entre os anos de 2017 e 2018. Além disso, a análise dos dados permitiu identificar padrões temporais de ocorrências, com picos de acidentes pela manhã, entre 06h e 09h, e à tarde, entre 15h e 18h, horários em que as pessoas geralmente saem para o trabalho ou escolas. Esses dados são fundamentais para a gestão pública, pois auxiliam no direcionamento de esforços preventivos para os períodos de maior risco. O modelo preditivo desenvolvido pelos autores obteve uma acurácia de 89,04%, demonstrando sua eficácia em prever acidentes de trânsito e evidenciando seu potencial como uma ferramenta importante para apoiar a redução de acidentes e os impactos na saúde pública.

Outro trabalho utilizando Inteligência Artificial nessa área foi apresentado por Modesto *et al.* (2022), cujo o objetivo é facilitar o uso de veículos no trânsito por pessoas com deficiência auditiva. Os autores desenvolveram um modelo de Rede Neural Artificial capaz de classificar diferentes sons bastante comuns no trânsito, como: apito, buzina e sirene. O objetivo deles é a aplicação desse projeto em um sistema de alerta para auxiliar condutores com deficiência auditiva.

Ferreira e Sassi (2011) desenvolveram um projeto semelhante no qual consistia em prever o comportamento do tráfego de veículos urbanos na cidade de São Paulo. Para isso, os autores aplicaram duas técnicas da Inteligência Artificial combinadas, a Lógica Fuzzy ou Lógica Difusa e as Redes Neurais Artificiais, que unidas formam uma rede chamada de Neuro Fuzzy. Após a aplicação dessas técnicas, foi possível concluir que a roteirização dinâmica de veículos combinada à previsão do comportamento do tráfego, permite uma maior eficácia da roteirização em uma cidade.

Olivatto (2021) desenvolveu um sistema responsável pela detecção automática de

rampas de acessibilidade a partir de imagens panorâmicas. Para a construção do banco de imagens, o autor se utilizou da ferramenta *Google Street View* e posteriormente, as rotulou manualmente para ser possível a utilização das mesmas em uma rede neural convolucional do detector de objetos YOLOv4. Após os treinamentos com técnicas de pré-processamento variadas, na fase de testes a rede apresentou uma precisão média de 85%. Com isso, o autor conclui que o sistema se mostrou eficaz de forma que possa auxiliar aplicações futuras que envolvam o mapeamento destas infraestruturas, ou até a inclusão de novas classes para aplicações mais elaboradas.

Outro trabalho a citar foi desenvolvido por Filho (2023), que a partir de uma rede neural convolucional, foi possível a detecção de emoções em vídeos visando uma solução para aprimorar a segurança no trânsito, através da análise dos condutores e possíveis comportamentos perigosos. Como resultado, foi possível obter uma precisão média e acurácia média de 82,82% e 81,00%, respectivamente.

4 METODOLOGIA

O presente trabalho se caracteriza pelo desenvolvimento de um protótipo responsável pela detecção de sono do motorista ao volante. Para tal, o sistema realizará a detecção através de técnicas de Visão Computacional utilizando o framework MediaPipe e, para o desenvolvimento do protótipo, um Raspberry Pi 3B juntamente com uma PiCamera V3.

Nesta seção, serão apresentados todos os métodos e procedimentos adotados para o desenvolvimento do trabalho.

4.1 Funcionamento do sistema

Utilizando a plataforma Google, foram realizadas pesquisas sobre o funcionamento e documentação do *framework* MediaPipe. A partir dos resultados foi possível compreender o funcionamento da ferramenta e considerar técnicas para realizar a detecção.

Foi decidido então, realizar a detecção de sono a partir da identificação de pontos situados nas pálpebras superiores e inferiores. A lógica aplicada consiste em calcular a distância entre dois pontos e se um limiar pré-determinado for ultrapassado, significa que o olho da pessoa está fechado.

4.1.1 Cálculo da distância euclidiana

A distância euclidiana é um conceito da matemática que mede a distância entre dois pontos em um espaço bidimensional. Ela se baseia no Teorema de Pitágoras, onde estabelece que em um triângulo retângulo, o quadrado da hipotenusa é igual a soma dos quadrados dos catetos. Com isso, a fórmula geral para a distância euclidiana entre dois pontos é dada por:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{4.1}$$

onde x_1 e y_1 são as coordenadas do primeiro ponto e x_2 e y_2 as coordenadas do segundo ponto. Relacionando ao problema de detecção, caso a distância seja menor que um valor *n* predeterminado, significa que o olho está fechado.

4.2 Implementação do algoritmo

No processo de implementação, os conceitos adquiridos nas pesquisas sobre o funcionamento do *framework* MediaPipe foram aplicados junto ao conhecimento de outras tecnologias para a obtenção dos resultados.

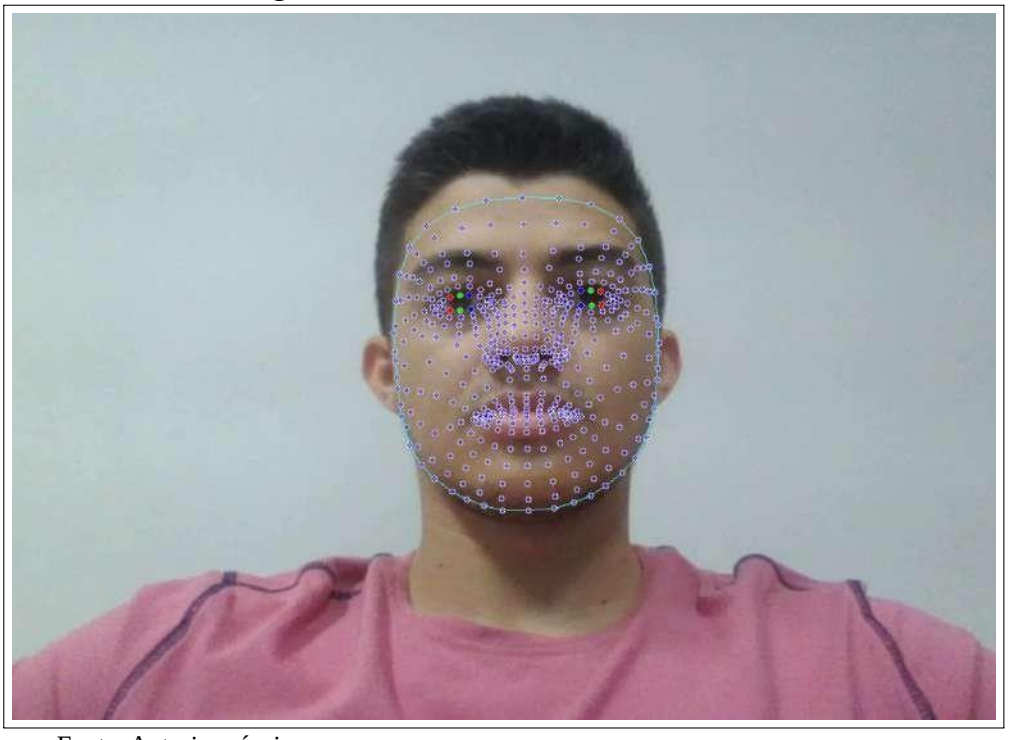
O algoritmo foi desenvolvido utilizando a linguagem de programação Python e compilado no editor de código Visual Studio Code. É possível separar a implementação em algumas etapas:

4.2.1 Detecção dos pontos faciais

Para essa primeira etapa, foi utilizada a solução *Face Landmarker* do MediaPipe, que realiza a detecção de 478 pontos de referência na face. Cada um desses *landmarks* possuem valores para as suas coordenadas x, y e z. E a partir desses valores que o sistema se baseia para realizar a detecção.

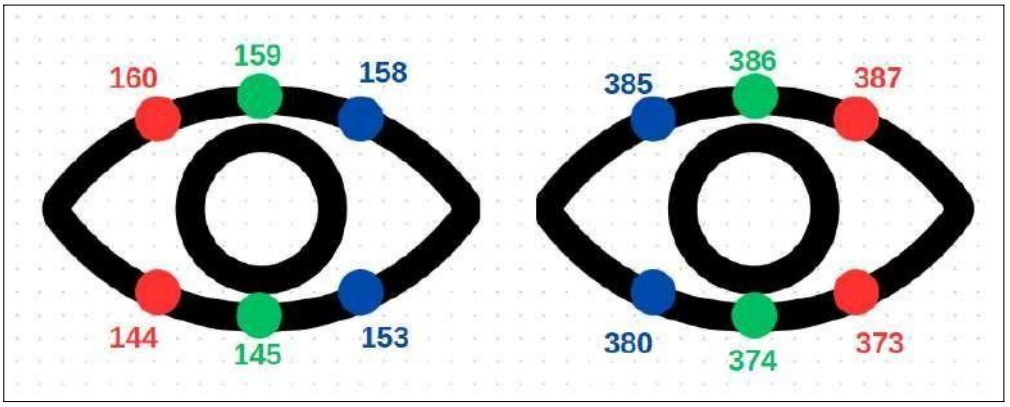
Após análise, foi constatado que não haveria a necessidade de utilizar todos esses pontos para o projeto. Então foram escolhidos 6 pontos situados ao redor das pálbebras para a identificação de quando o olho estivesse aberto ou fechado, como mostra a Figura 10 e 11.

Figura 10 – Pontos de referência na face.



Fonte: Autoria própria.

Figura 11 – Pontos de detecção escolhidos.



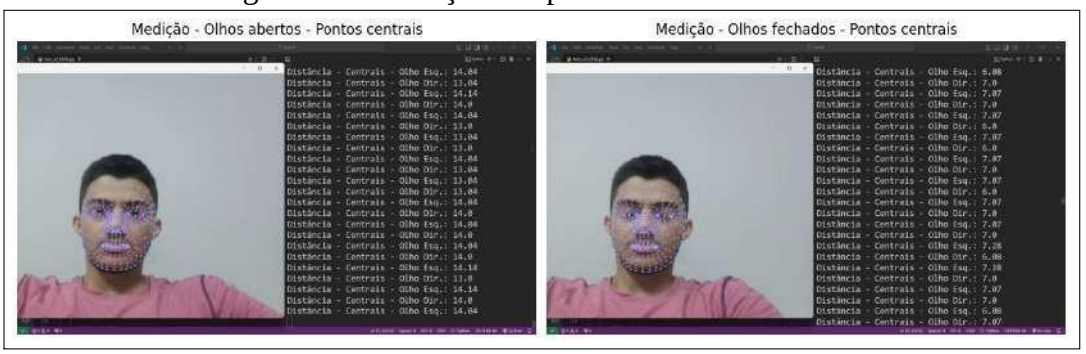
Fonte: Autoria própria.

4.2.2 Aplicando o cálculo da distância euclidiana

Utilizando a lógica do cálculo da distância euclidiana apresentada anteriormente, foi realizada a implementação da mesma se baseando nos pontos mostrados na Figura 11. O cálculo da distância é realizado para cada par de pontos da pálpebra superior e inferior, por exemplo: é realizado o cálculo entre os pontos 160 e 144.

Com o cálculo aplicado para cada par de pontos, foi realizada a medição dos valores com os olhos abertos e fechados, como mostra a Figura 12 e 13. A partir dessa comparação, foi determinado um limiar de valor 7 para a distância dos pontos das extremidades e outro de valor 8 para a distância dos pontos centrais.

Figura 12 – Medição dos pontos centrais dos olhos.



Fonte: Autoria própria.

Figura 13 – Medição dos pontos das extremidades dos olhos.

Fonte: Autoria própria.

4.2.3 Adicionando uma mensagem de alerta

Iniciando a implementação do sistema, foi decidido realizar uma contagem de 3 segundos para verificar se os olhos se mantinham fechados. Com essa condição determinada, foi adicionada uma função para apresentar uma mensagem indicando se o olho estivesse fechado ou não, indicando assim se o motorista estaria dormindo. A Figura 14 apresenta esse teste inicial do sistema.

Figura 14 – Teste inicial - Apresentando mensagem.

Fonte: Autoria própria.

4.3 Integração do algoritmo com sistemas embarcados

Após finalizada a implementação do sistema na parte do software, foi iniciada a fase de integração do algoritmo com uma plataforma física. O objetivo final é a utilização de uma Raspberry Pi 3B juntamente com uma câmera compatível com a plataforma e um buzzer para emitir um alerta sonoro.

4.3.1 Testes com a plataforma Arduino

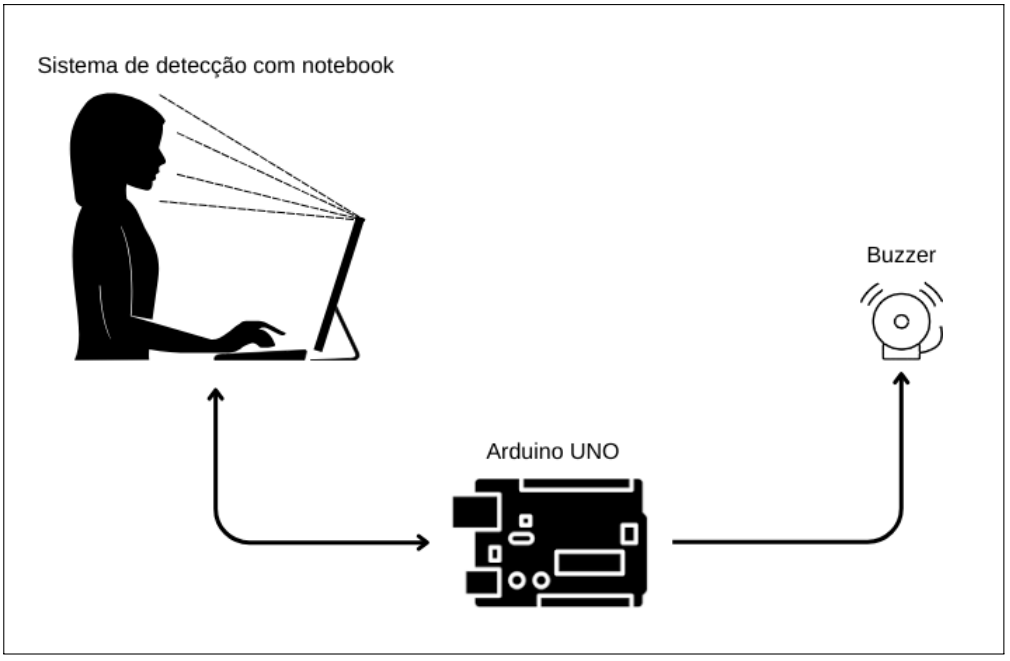
Primeiramente, com o intuito de realizar um teste inicial com uma plataforma física, foi utilizado um Arduino UNO juntamente com um buzzer para emitir um sinal sonoro de alerta. A Figura 15 apresenta um diagrama para facilitar a visualização das conexões desta etapa.

O código em python foi executado no notebook com sua própria webcam, como na etapa anterior. E para a conexão do Arduino com o sistema de detecção, foi utilizada a porta USB do notebook e a biblioteca *serial* do python, como mostra o Código-Fonte 1.

Código-fonte 1 – Utilização da biblioteca serial

```
1 import serial
2 arduino = serial.Serial('COM6', 9600)
```

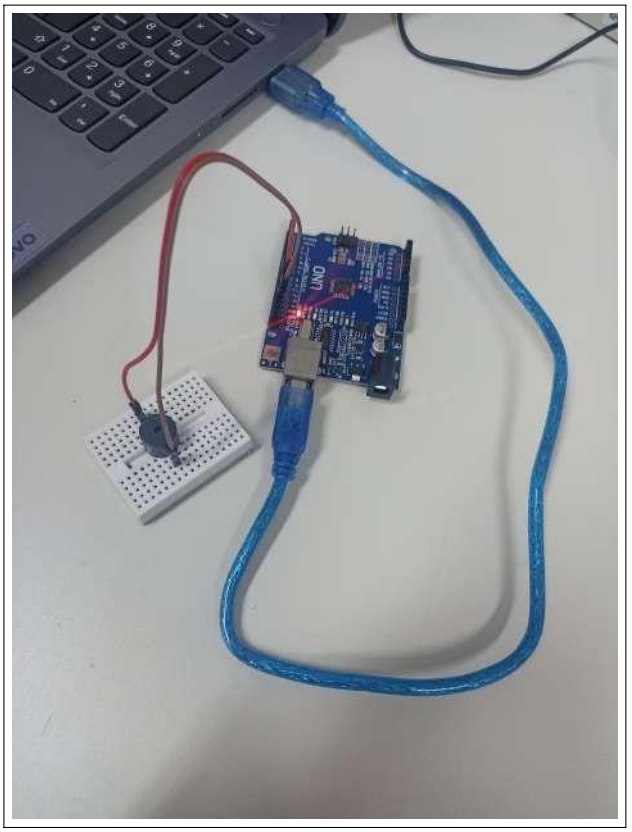
Figura 15 – Diagrama de Conexões do teste com Arduino.



Fonte: Autoria própria.

Após a montagem do sistema com o Arduino foi possível obter um primeiro resultado com uma plataforma física, como apresenta a Figura 16. Essa primeira solução atendeu às necessidades, realizando a detecção e emitindo um sinal sonoro ao detectar que os olhos estavam fechados por mais de 3 segundos.

Figura 16 – Testes com o protótipo utilizando Arduino UNO.



Fonte: Autoria própria.

4.3.2 Utilizando a plataforma Raspberry Pi

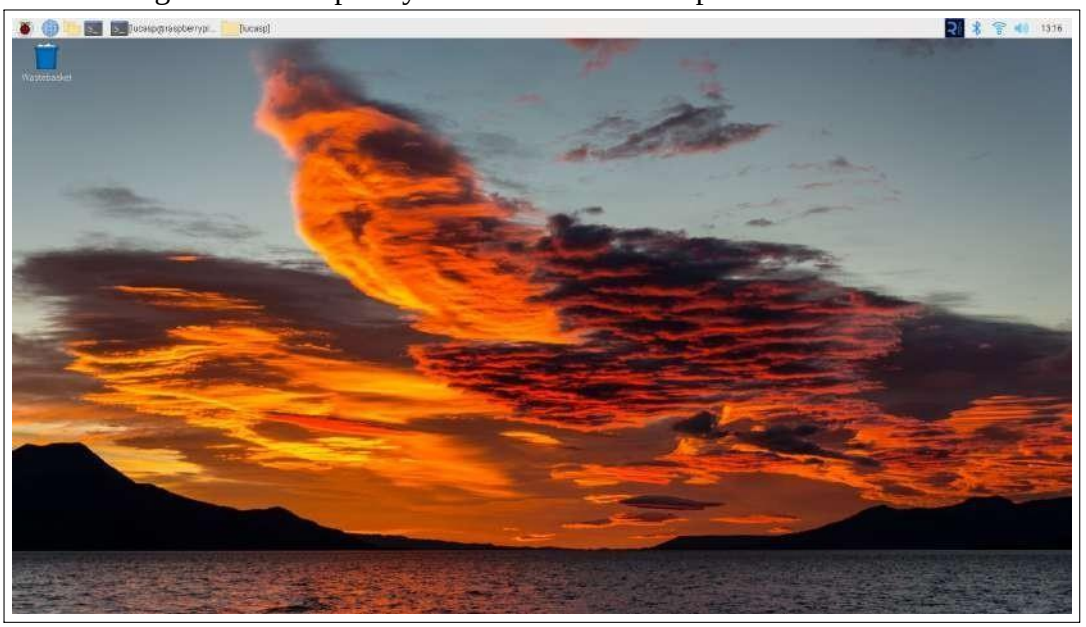
Nesta etapa, se deu início à utilização da plataforma escolhida para o protótipo final, o Rapsberry Pi 3B. Para tornar a plataforma operável para o devido sistema, foi preciso realizar algumas etapas:

1. **Realizar o download e instalação do programa Raspberry Pi Imager:**
   Esse programa pode ser baixado no próprio site do fabricante.
2. **Instalação do Sistema Operacional no cartão microSD:**
   Utilizando o Rasbperry Pi Imager, é possível instalar o sistema operacional que desejar no cartão. Para o projeto, foi instalado o sistema mais atual, o Raspberry Pi OS (64-bit).
3. **Instalação das bibliotecas e pacotes necessários:**
   Para a instalação das bibliotecas, foi utilizado o próprio terminal do sistema. As principais bibliotecas e pacotes instalados foram: Python, MediaPipe e OpenCV.

Com o sistema plenamente funcional, como ilustrado na Figura 17, foram iniciados os primeiros testes com a plataforma Raspberry Pi 3B, utilizando uma webcam USB modelo Fortrek HD GT para realizar a detecção e o reconhecimento. A Figura 18 apresenta o teste

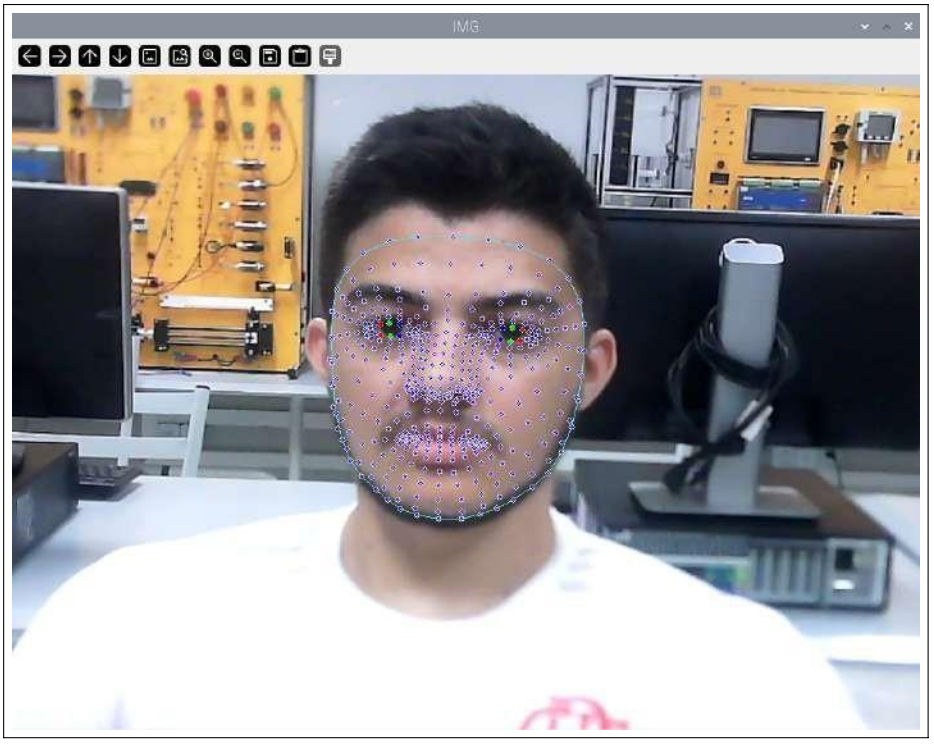
inicial, cujo objetivo foi avaliar a compatibilidade e o desempenho do framework MediaPipe na nova placa.

Figura 17 – Raspberry Pi 3B com Sistema Operacional instalado.



Fonte: Autoria própria.

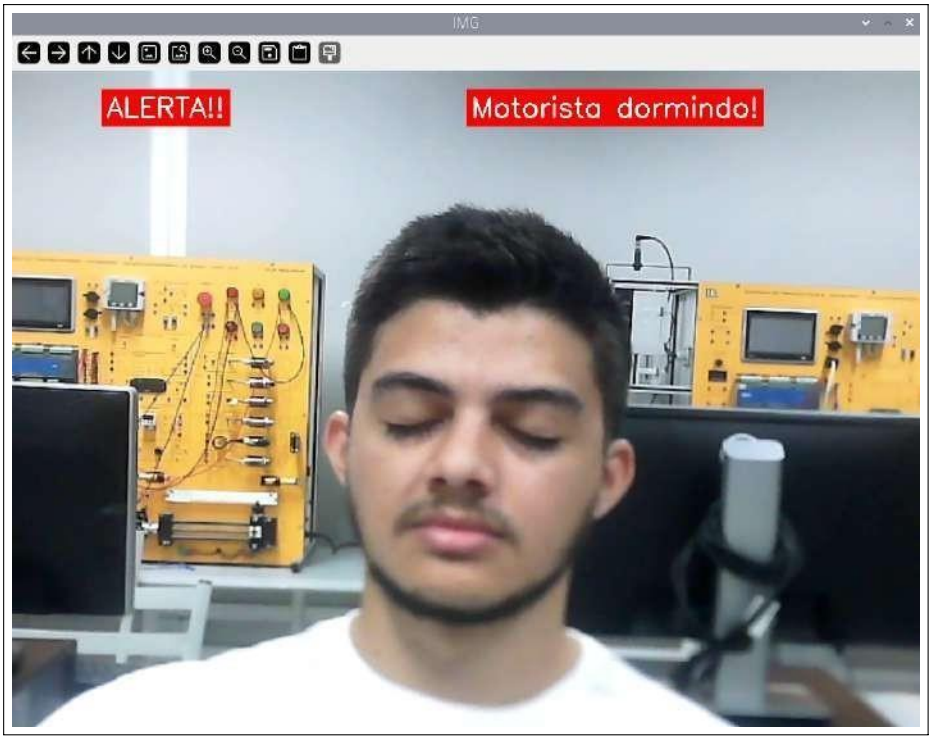
Figura 18 – Teste de reconhecimento do Mediapipe com o Raspberry Pi 3B.



Fonte: Autoria própria.

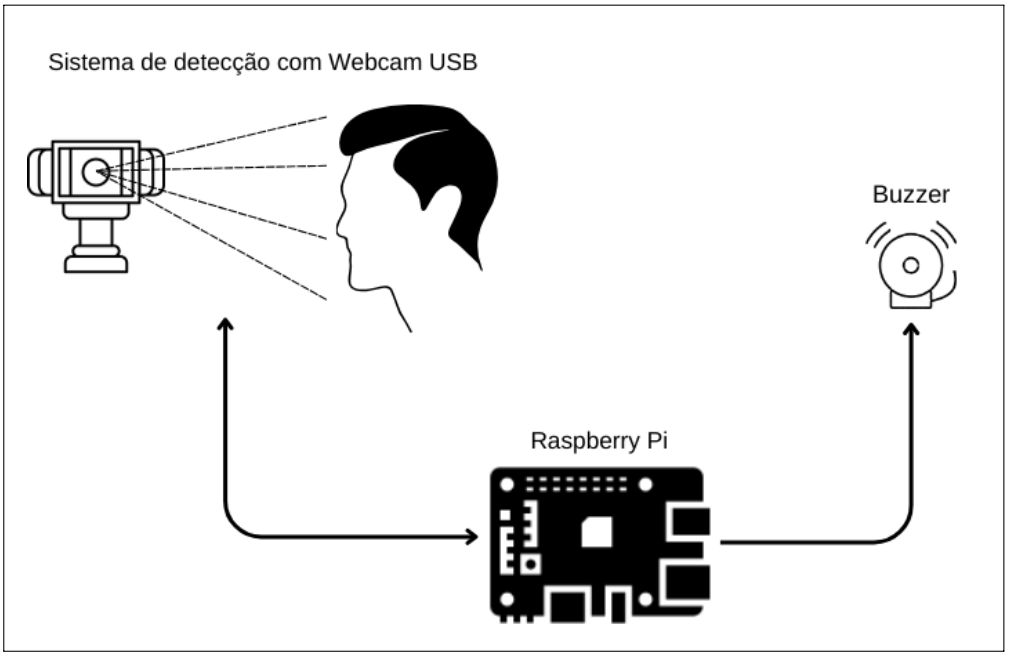
Após confirmar que o sistema funcionava corretamente com o *framework* MediaPipe, o buzzer foi adicionado novamente para um teste completo. As Figuras 19 e 20 apresentam, respectivamente, o teste realizado e o diagrama de funcionamento do sistema.

Figura 19 – Teste do sistema utilizando o Raspberry Pi 3B.



Fonte: Autoria própria.

Figura 20 – Diagrama de Conexões do teste com o Raspberry Pi 3B.

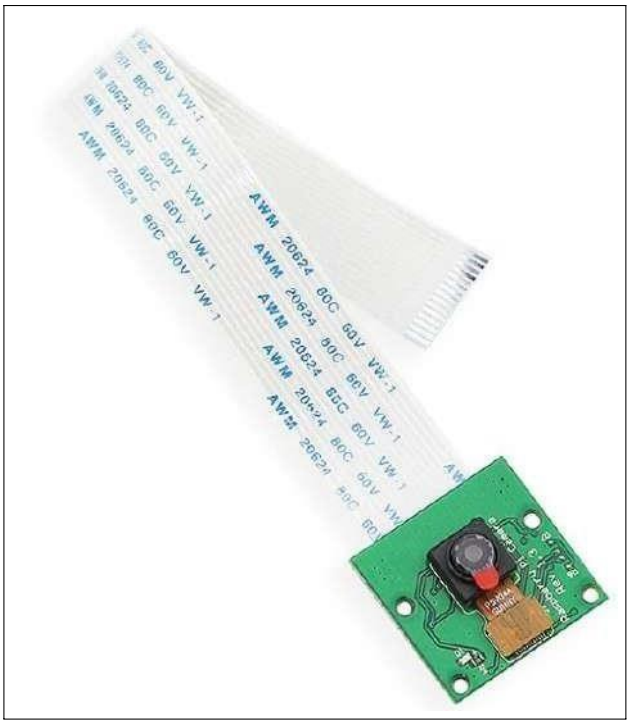


Fonte: Autoria própria.

4.3.3 Aperfeiçoando o sistema

Após realizar testes com o protótipo utilizando a plataforma Raspberry Pi 3B e uma webcam USB, conforme apresentado na Figura 20, verificou-se que a câmera empregada apresentava limitações, como seu tamanho inadequado e o *delay* nas capturas. Diante disso, optou-se por substituí-la por um módulo de câmera específico para o Raspberry Pi: a PiCamera V3.

Figura 21 – Picamera V3



Fonte: Autoria própria.

A substituição da webcam USB pela PiCamera V3 foi motivada pelas diferenças significativas entre os dois dispositivos, tanto em termos de especificações técnicas quanto de funcionalidade. Enquanto a webcam USB apresentou limitações que comprometiam o desempenho do sistema, a PiCamera V3 demonstrou ser mais adequada para as necessidades do projeto, especialmente devido ao seu design compacto e integração direta com a Raspberry Pi. Na Tabela 1 é possível observar as principais diferenças que influenciaram a escolha da PiCamera V3.

Tabela 1 – Comparativo entre as duas câmeras

Características	Webcam USB	PiCamera V3
Resolução	1 MP	5 MP
Quadros por segundo	30 fps	30 fps
Peso	72,8 g	20 g
Dimensões	8 x 8,5 x 3 cm	2,5 x 2,4 x 0,7 cm

Fonte: Elaborado pelo autor.

Para integrar a PiCamera ao sistema, foi necessário instalar sua respectiva biblioteca e realizar algumas modificações no código para permitir a captação da imagem a partir do novo componente, como mostra o Código-fonte 2.

Código-fonte 2 – Modificações para utilização da PiCamera

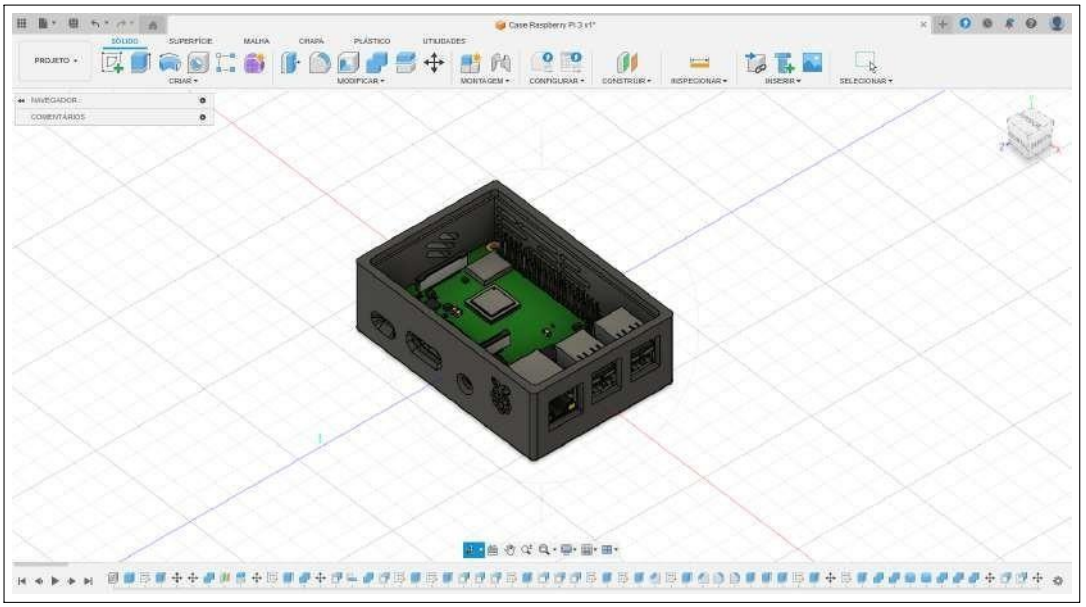
```
from picamera2 import Picamera2

# Inicializando a camera
picam2 = Picamera2()
camera_config = picam2.create_preview_configuration(main={"
    size": (1000, 720)})
picam2.configure(camera_config)
picam2.start()
```

Com essa substituição, novos testes foram realizados para avaliar a estabilidade e a precisão da detecção. O sistema mostrou uma melhora significativa na captação de imagens, com menor atraso e maior eficiência no processamento. Além disso, o design compacto da PiCamera facilitou sua instalação e permitiu avançar para a montagem do protótipo final.

4.4 Desenvolvimento do protótipo

Um modelo foi desenvolvido no software Fusion 360 para integrar todos os componentes do sistema de forma eficiente e organizada. O projeto foi planejado com foco na funcionalidade e praticidade, considerando as dimensões exatas da Raspberry Pi 3B, a fim de garantir um encaixe seguro e preciso na case, sem folgas ou desalinhamentos. A parte inferior do modelo foi projetada para acomodar a placa e os componentes eletrônicos de maneira estruturada, conforme ilustrado na Figura 22.

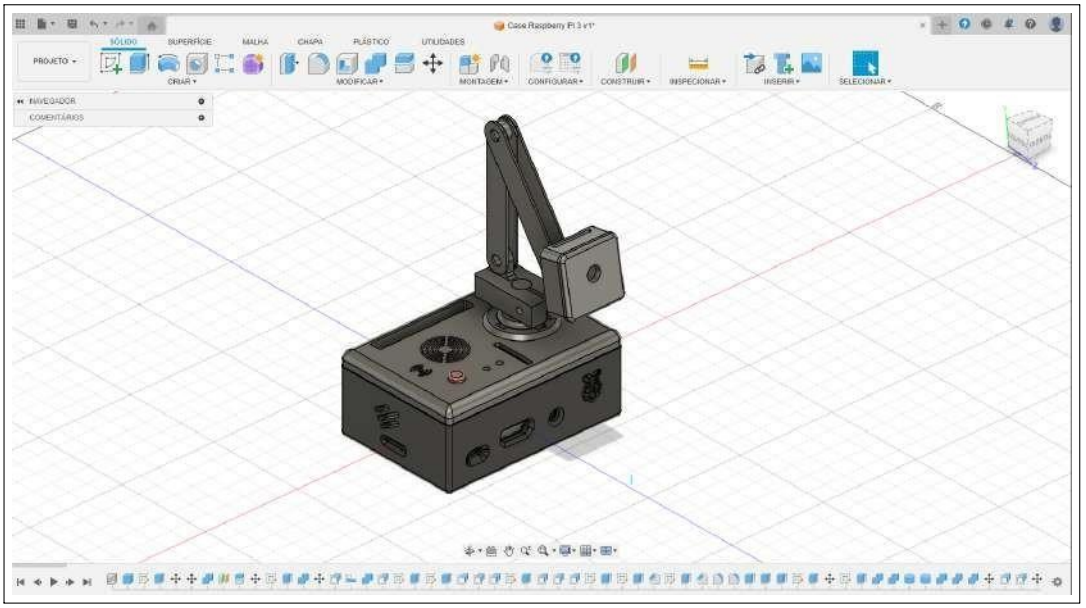
Figura 22 – Parte inferior do modelo 3D



Fonte: Autoria própria.

Na etapa seguinte, foi projetada uma tampa superior com a finalidade de proteger os componentes internos e acomodar elementos adicionais, como o buzzer, LEDs indicadores e um botão de ligar/desligar. A tampa foi planejada com furos para ventilação, garantindo a eficiência do sistema de refrigeração integrado, e incorporou também um suporte articulado para a PiCamera. O modelo 3D completo, já com a tampa e o suporte articulável, é apresentado na Figura 23.

Figura 23 – Modelo 3D completo

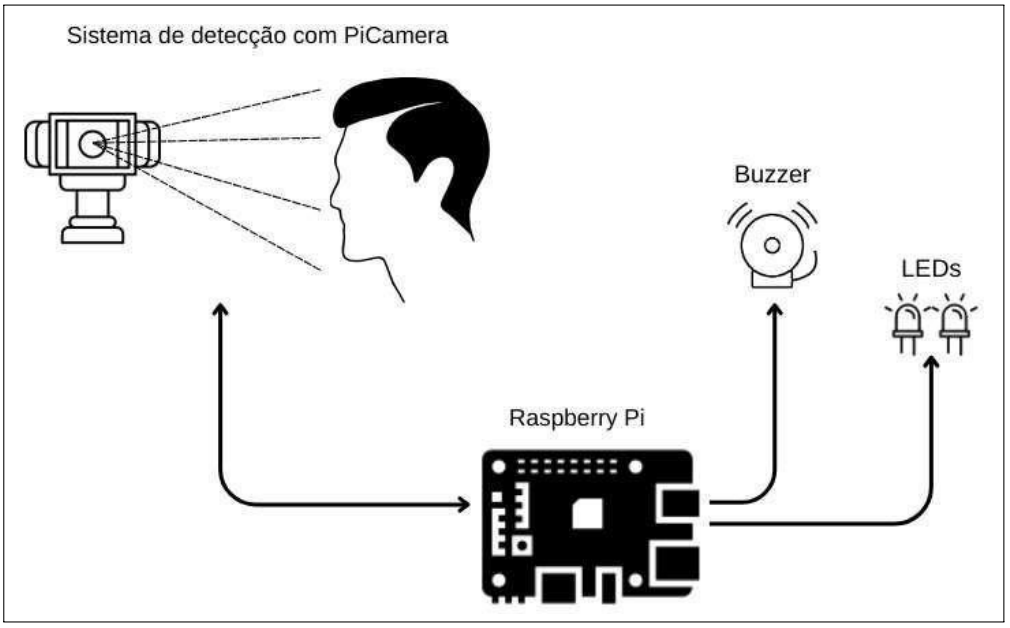


Fonte: Autoria própria.

4.5 Protótipo final

Após a execução de todas as etapas descritas anteriormente e os testes correspondentes, foi possível definir a configuração final do protótipo, composta pelo Raspberry Pi, a PiCamera, um buzzer, dois LEDs indicadores e um botão para ligar/desligar, conforme ilustrado no diagrama da Figura 24. Os dois LEDs adicionados têm a função de orientar o usuário sobre o posicionamento correto da câmera, indicando se o rosto está sendo detectado ou não.

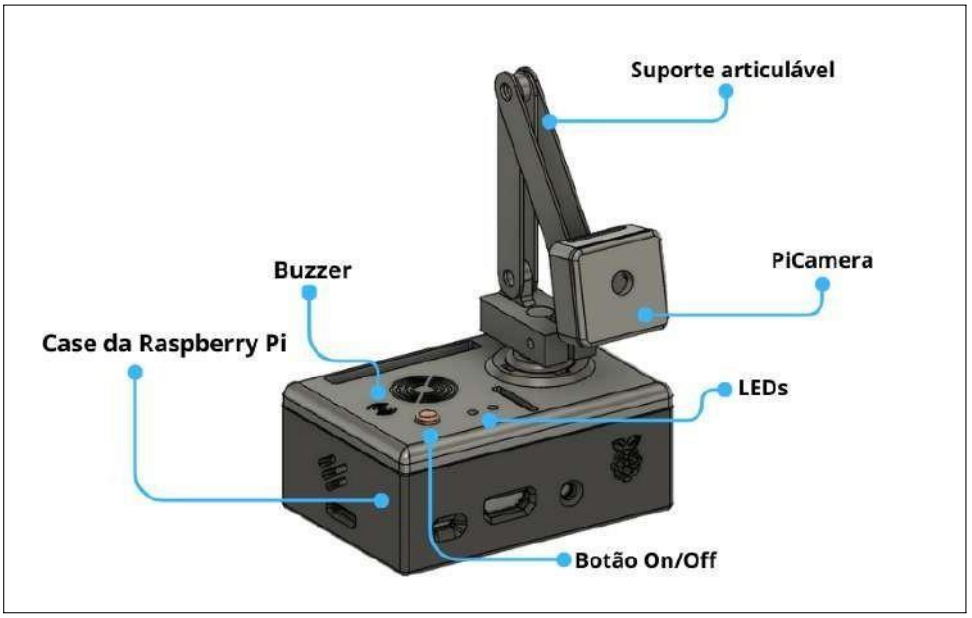
Figura 24 – Diagrama final



Fonte: Autoria própria.

O protótipo físico finalizado inclui uma case para a placa Raspberry Pi 3B e os demais componentes, além de um suporte articulado para a PiCamera, que possibilita o ajuste da câmera nos três eixos (x, y e z), como apresentado na Figura 25. O projeto também incorpora um sistema de refrigeração com cooler e dissipadores, bem como um botão de liga/desliga.

Figura 25 – Protótipo desenvolvido



Fonte: Autoria própria.

4.5.1 Viabilidade e custos dos componentes

Uma das principais preocupações durante o desenvolvimento do protótipo foi garantir a viabilidade econômica do sistema, facilitando sua replicação em cenários reais. O custo total estimado do protótipo, com valores aproximados para os principais componentes utilizados e suas respectivas fontes de preço, está apresentado na Tabela 2.

Tabela 2 – Custo estimado dos componentes utilizados

Componente	**Custo (R$)**	**Fonte (Link)**
Raspberry Pi 3B	385,60	https://abrir.link/KHsOO
PiCamera V3	48,00	https://abrir.link/cPjtd
LEDs	0,50	https://abrir.link/lOHtd
Buzzer	2,90	https://abrir.link/DWysX
Botão On/Off	0,20	https://abrir.link/eruEj
Total	**437,20**	

Fonte: Elaborado pelo autor.

É importante destacar que o custo total estimado pode ser reduzido com a substituição de alguns componentes por alternativas mais acessíveis, como a troca do Raspberry Pi 3B por um modelo de menor custo. O Raspberry Pi 3B foi escolhido principalmente devido à sua disponibilidade durante o desenvolvimento do protótipo, mas o uso desse modelo em específico não é obrigatório para o funcionamento do sistema.

5 EXPERIMENTOS

Nesta seção, serão apresentados os testes realizados para validar a eficácia do protótipo. Serão discutidos os métodos de avaliação utilizados, os dados coletados em diferentes condições de operação e as melhorias alcançadas durante o processo de aperfeiçoamento do sistema.

5.1 Testes Realizados

Com o protótipo finalizado, deu-se início à fase de testes. Primeiramente, foram realizados testes em um ambiente controlado e posteriormente, o protótipo foi testado em um carro para verificar sua eficiência em um ambiente real.

5.1.1 Testes no ambiente controlado

Na fase inicial de testes, foram realizados experimentos com diferentes indivíduos para avaliar a precisão do sistema em usuários variados. Adicionalmente, para cada participante, o protótipo foi testado com o rosto em diversas posições, conforme ilustrado na Figura 26.

Figura 26 – Ângulos utilizados para o teste

Fonte: Autoria própria.

Definidas as variações a serem testadas, deu-se início aos testes. Na Figura 27 é possível observar os primeiros testes com o Usuário 1.

Figura 27 – Teste com o Usuário 1

Fonte: Autoria própria.

O mesmo teste foi realizado com o segundo e terceiro voluntário, como mostrado nas Figuras 28 e 29.

Figura 28 – Teste com o Usuário 2

Fonte: Autoria própria.

Figura 29 – Teste com o Usuário 3

Fonte: Autoria própria.

Ao analisar os testes com cada usuário e as variações de posição do rosto, constatou-se que o sistema realizou corretamente todas as detecções previstas para o teste. A Tabela 3 consta com os resultados obtidos de cada teste.

Tabela 3 – Resultados dos testes

Usuários	Inclinação p/ esquerda	Centralizado	Inclinação p/ direita
Usuário 1	Sono detectado	Sono detectado	Sono detectado
Usuário 2	Sono detectado	Sono detectado	Sono detectado
Usuário 3	Sono detectado	Sono detectado	Sono detectado

Fonte: Elaborado pelo autor.

5.1.2 Testes no ambiente real

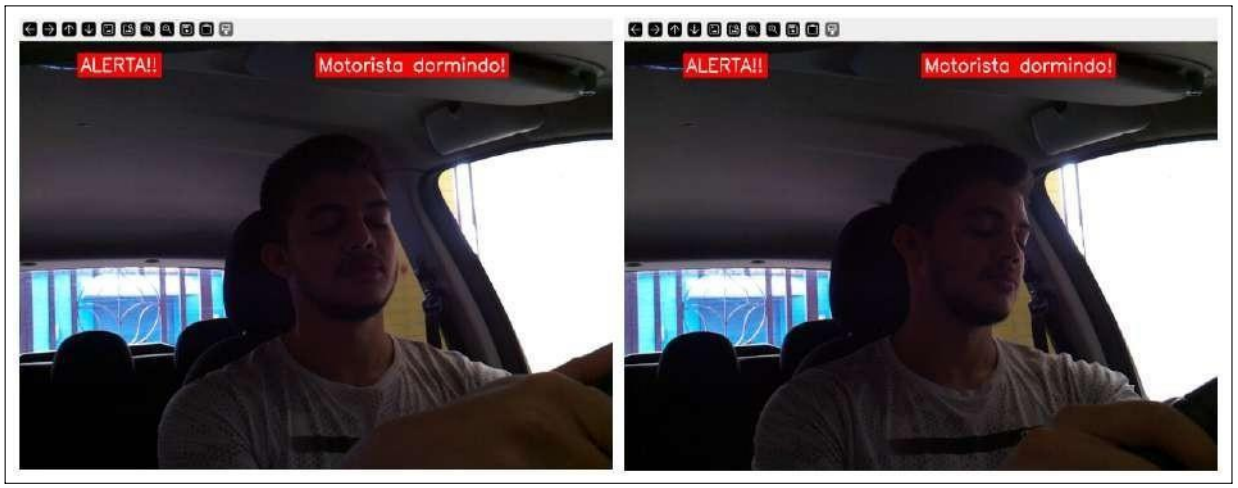
Após confirmar a eficiência do protótipo em ambiente controlado, ele foi instalado em um veículo para avaliar sua eficácia em condições reais. O dispositivo foi posicionado no local mais adequado para garantir a visibilidade do motorista e evitar qualquer obstrução durante a condução, conforme ilustrado na Figura 30.

Figura 30 – Protótipo instalado no carro

Fonte: Autoria própria.

Com o protótipo devidamente instalado, foram conduzidos testes de desempenho para verificar sua funcionalidade em situações mais próximas do uso cotidiano. Diferentemente dos testes realizados em ambiente controlado, esses experimentos ocorreram em condições de iluminação reduzida, conforme ilustrado na Figura 31. Apesar dessa variação na luminosidade, o protótipo demonstrou ser eficiente, detectando com precisão os olhos fechados e emitindo alertas sonoros de forma consistente. Esses resultados reforçam a robustez do sistema, indicando que ele pode operar com confiabilidade mesmo em cenários desafiadores de iluminação.

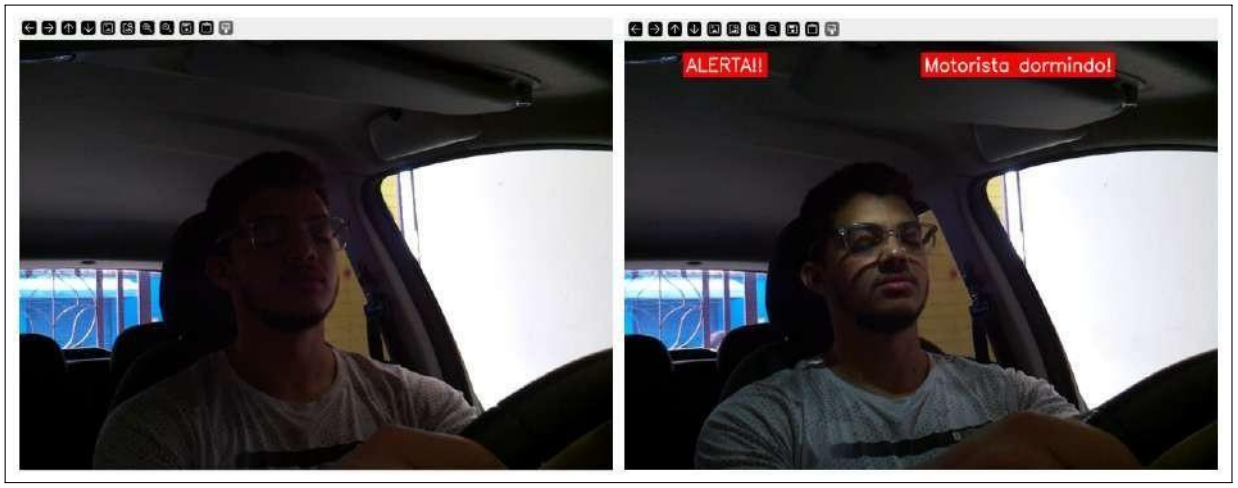
Figura 31 – Teste final do protótipo



Fonte: Autoria própria.

Além disso, foi realizado um teste com o usuário utilizando óculos de grau, a fim de avaliar o impacto desse acessório no desempenho do sistema. Em condições de baixa luminosidade, foi constatado que o protótipo enfrentava dificuldades para realizar a detecção correta, devido ao reflexo gerado pelos óculos, que interferia na identificação dos olhos. Para validar essa limitação, foi adicionada uma fonte de iluminação externa ao ambiente, o que resultou em uma detecção precisa e funcionamento adequado do sistema. A Figura 32 ilustra essa comparação, evidenciando a importância de condições de iluminação adequadas para maximizar a eficácia do protótipo ao monitorar o estado dos olhos do motorista.

Figura 32 – Teste final do protótipo utilizando óculos



Fonte: Autoria própria.

6 CONCLUSÃO

O trabalho proposto apresentou um sistema para detecção de sonolência ao volante por meio do desenvolvimento de um protótipo que utiliza um algoritmo de Visão Computacional com o *framework* MediaPipe, uma placa Raspberry Pi 3B, uma PiCamera, um buzzer, dois LEDs indicadores e um botão para ligar/desligar. Após a realização dos testes, tanto em ambiente controlado quanto em um ambiente real, o projeto demonstrou-se eficiente e promissor no auxílio à prevenção de acidentes de trânsito causados pela sonolência.

Com o teste final utilizando óculos de grau e ajustes de iluminação, foi possível observar tanto a robustez do protótipo em detectar o estado de alerta do motorista quanto algumas limitações que podem ser aprimoradas em trabalhos futuros. No geral, o sistema se mostrou eficiente em um ambiente realista, detectando com precisão quando o motorista apresentava sinais de fechamento dos olhos, mas também evidenciou a influência de variáveis externas, como iluminação e reflexos, que podem comprometer a detecção.

6.1 Trabalhos Futuros

Este trabalho contribui para o campo dos sistemas de segurança veicular, com diversas possibilidades de aprimoramento em estudos futuros. As principais sugestões incluem:

- **Aplicação de Técnicas de *Machine Learning***: Explorar algoritmos avançados para detectar outros sinais de sonolência, como bocejos, ampliando a precisão e robustez do sistema.
- **Otimização do Sistema Embarcado**: Implementar técnicas de otimização para melhorar o desempenho do protótipo em plataformas de recursos limitados, garantindo eficiência sem comprometer a qualidade da detecção
- **Substituição da PiCamera**: Utilizar uma câmera mais adequada para capturar imagens noturnas, o que possibilitará o funcionamento eficaz do sistema mesmo em condições de baixa luminosidade, ampliando sua aplicabilidade em diferentes horários.

Com essas melhorias, o sistema terá potencial para se tornar uma solução mais completa e eficiente. Dessa forma, poderá contribuir de maneira significativa para a segurança viária e para a redução de acidentes relacionados à sonolência ao volante, promovendo um trânsito mais seguro e responsável.

REFERÊNCIAS

AGUIAR, W. S.; FONTENELLE, J. M. B.; SILVA, V. R.; NETO, I. A. d. C.; NASCIMENTO, F. W. S. d.; SILVA, L. P. C. Uso da inteligência artificial para predição de acidentes no trânsito. In: **Anais do 11 º Congresso Brasileiro de Epidemiologia**. [S.l.: s.n.], 2021.

ARDUINO. **Arduino Home**. 2024. <https://www.arduino.cc>. Accessed: (12-setembro-2024).

BALLARD, D. H.; BROWN, C. M. **Computer Vision**. [S.l.]: Prentice Hall, 1982.

BATISTA, G. E. d. A. P. A. Pré-processamento de dados em aprendizado de máquina supervisionado. In: **Universidade de São Paulo**. [S.l.: s.n.], 2003.

CHAVES, L. E. **O que é Visão Computacional e as 10 Principais Aplicações em 2022**. 2022. Accessed: (8-maio-2024). Disponível em: <https://www.engenhariahibrida.com.br/post/o-que-e-visao-computacional-10-principais-aplicacoes>.

CHAVES, T. Z. L. Uso de visão computacional com redes neurais convolucionais para classificação de fungos. In: **Universidade Federal de Santa Catarina**. [S.l.: s.n.], 2023.

CORTEZ, D. E. d. S. Desenvolvimento de um sistema de controle de tráfego inteligente baseado em visão computacional. In: **Universidade Federal do Rio Grande do Norte**. [S.l.: s.n.], 2022.

FERNANDES, A. M. d. R. **Inteligência artificial: noções gerais**. [S.l.]: Visual Books, 2003.

FERREIRA, R. P.; SASSI, R. J. Combinação de técnicas da inteligência artificial para previsão do comportamento do tráfego veicular urbano na cidade de são paulo. In: **Universidade Nove de Julho**. [S.l.: s.n.], 2011.

FILHO, C. L. S. S. Classificação de emoções usando inteligência artificial para maior segurança no trânsito. In: **Unidade de Ensino Superior Dom Bosco**. [S.l.: s.n.], 2023.

GLOBAL, B. **Acidentes de trânsito: principais causas e soluções de um problema que gera 1 morte no Brasil a cada 15 minutos**. 2023. Accessed: (8-maio-2024). Disponível em: <https://www.bosch.com.br/noticias-e-historias/mobilidade/acidentes-de-transito-principais-causas-e-solucoes/>.

GOGONI, R. **O que é o Raspberry Pi?** 2019. <https://embarcados.com.br/o-que-sao-sistemas-embarcados/>. Accessed: (12-setembro-2024).

KOENIGKAM-SANTOS, M.; FERREIRA-JúNIOR, J. R.; WADA, D. T.; TENóRIO, A.; NOGUEIRA-BARBOSA, M. H.; AZEVEDO-MARQUES, P. Inteligência artificial, aprendizado de máquina, diagnóstico auxiliado por computador e radiômica: avanços da imagem rumo à medicina de precisão. **Radiol Bras**, 2019.

LUGARESI, C.; TANG, J.; NASH, H.; MCCLANAHAN, C.; UBOWEJA, E.; HAYS, M.; ZHANG, F.; CHANG, C.-L.; YONG, M. G.; LEE, J.; CHANG, W.-T.; HUA, W.; GEORG, M.; GRUNDMANN, M. Mediapipe: A framework for building perception pipelines. In: **Cornell University**. [S.l.: s.n.], 2019.

MARTINS, E. A. Sistema de segurança residencial com raspberry pi. In: **Faculdade de Tecnologia de São Paulo**. [S.l.: s.n.], 2022.

MATHEUS, G. R.; SALINA, F. V. Algoritmos para detecção e reconhecimento de placas de trânsito. In: **Workshop de Inovação, Pesquisa, Ensino e Extensão**. [S.l.: s.n.], 2019.

MEDIAPIPE. **Google for Developers**. 2024. <https://developers.google.com/mediapipe>. Accessed: (10-maio-2024).

MITCHELL, T. M. **Machine Learning**. [S.l.]: McGraw-Hill Science/Engineering/Math, 1997.

MODESTO, M. C.; JUNIOR, P. G. d. S.; BRITO, R. M. Desenvolvimento de uma inteligência artificial para a classificação de sinais sonoros de trânsito. In: **Centro Universitário do Estado do Pará**. [S.l.: s.n.], 2022.

MONARD, M. C.; BARANAUKAS, J. A. Aplicações de inteligência artificial: Uma visão geral. **São Carlos: Instituto de Ciências Matemáticas e de Computação de São Carlos**, 2000.

NIERO, G.; SOUZA, G. B. d.; FABRE, D. T. Controle de trânsito através de visão computacional. In: **7º Simpósio de Integração Científica e Tecnológica do Sul Catarinense – SICT-Sul**. [S.l.: s.n.], 2018.

OLIVATTO, T. F. Identificação automática de rampas de acessibilidade apoiada por visão computacional a partir de imagens panorâmicas street-level. In: **Universidade Federal de São Carlos**. [S.l.: s.n.], 2021.

OPENCV. **OpenCV**. 2024. Accessed: (30-maio-2024). Disponível em: <https: //opencv.org/about/>.

PYTHON. **Python**. 2024. Accessed: (30-maio-2024). Disponível em: <https://www.python.org/ about/>.

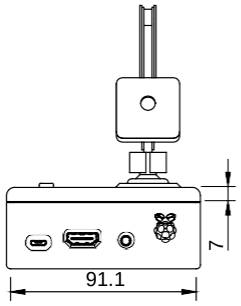
RASPBERRY. **Raspberry Home**. 2024. <https://www.raspberrypi.com>. Accessed: (25-novembro-2024).

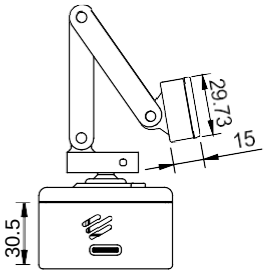
SALES, M. B. Uma revisão abrangente sobre o papel transformador da inteligência artificial em estudos com enzimas e sua relevância na indústria. In: **Universidade da Integração Internacional da Lusofonia Afro-Brasileira**. [S.l.: s.n.], 2023.

SOUSA, N. C. **Dormir no volante está entre principais causas de acidentes**. 2023. Accessed: (8-maio-2024). Disponível em: <https://www.al.pi.leg.br/radio/noticias-radio/ dormir-no-volante-esta-entre-principais-causas-de-acidentes>.

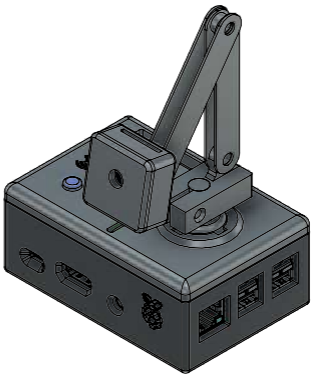
SOUZA, F. **O que são sistemas embarcados?** 2022. <https://embarcados.com.br/ o-que-sao-sistemas-embarcados/>. Accessed: (12-setembro-2024).

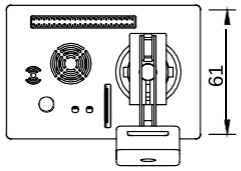
APÊNDICE A – DESENHO TÉCNICO DO PROTÓTIPO





PROTÓTIPO PARA
DETECÇÃO DE SONO





Dept.	Technical reference	Created by: Adriano Barroso & Lucas Pinheiro	Approved by
PGERC		Document type	Document status
		Title: **Protótipo para detecção de sono**	DWG No.
			Rev. / Date of issue / Sheet: 1/2