



**FACULDADE DE TECNOLOGIA DO AMAPÁ - META**  
**CURSO DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO**

**HENRIQUE DE SOUZA DAMIÃO**  
**RENATA PRISCILA TAVARES DE MATOS**

**APLICAÇÃO DA INTELIGÊNCIA ARTIFICIAL NO CONTROLE E**  
**GERENCIAMENTO DE ESTACIONAMENTOS PARA BICICLETAS**

**MACAPÁ – AP**  
**2024**

**HENRIQUE DE SOUZA DAMIÃO**  
**RENATA PRISCILA TAVARES DE MATOS**

**APLICAÇÃO DA INTELIGÊNCIA ARTIFICIAL NO CONTROLE E  
GERENCIAMENTO DE ESTACIONAMENTOS PARA BICICLETAS**

Trabalho de Conclusão de Curso na  
Modalidade de Monografia apresentada à  
Faculdade de Tecnologia do Amapá -  
META, como requisito final à obtenção do  
grau de Bacharel em Engenharia de  
Computação, sob a orientação do Prof.  
Me. Sergio Clayton Viana Pinheiro.

**MACAPÁ – AP**  
**2024**

**HENRIQUE DE SOUZA DAMIÃO**

**RENATA PRISCILA TAVARES DE MATOS**

**APLICAÇÃO DA INTELIGÊNCIA ARTIFICIAL NO CONTROLE E  
GERENCIAMENTO DE ESTACIONAMENTOS PARA BICICLETAS**

Trabalho de Conclusão de Curso na Modalidade de Monografia apresentada à Faculdade de Tecnologia do Amapá - META, como requisito final à obtenção do grau de Bacharel em Engenharia da computação.

DATA DE APROVAÇÃO: \_\_\_\_/\_\_\_\_/\_\_\_\_.

---

Prof. Me. Sergio Clayton Viana Pinheiro.

---

Prof. XXXXXXXXXXXXXXXXXXXX (Avaliador 1)

---

Profª. XXXXXXXXXXXXXXXXXXXX (Avaliador 2)

## AGRADECIMENTOS

Eu, **Renata Priscila Tavares de Matos** quero agradecer primeiramente a Deus, fonte inesgotável de sabedoria e mestre constante em minha jornada acadêmica. À minha mãe Maria de Deus, cujo apoio incondicional, amor e incentivo foram alicerces fundamentais para alcançar esta conquista.

À minha avó, Raimunda Tavares, quero expressar minha profunda gratidão por todo o amor, apoio, inspiração e orações. Sua sabedoria, paciência e força foram fundamentais para minha formação e sucesso. Obrigada por sempre acreditar em mim e por ser um exemplo de dedicação e perseverança.

Às minhas tias Socorro, Ercília e Edna, que sempre estiveram presentes com palavras de encorajamento e apoio, agradeço por serem fontes de inspiração e motivação ao longo dessa jornada.

À minha querida prima Rosane, agradeço pela colaboração, apoio e compreensão ao longo desta jornada acadêmica, sempre me dando os melhores conselhos. Sua presença tornou esta jornada ainda mais especial.

Ao meu orientador e professor Sérgio, expresso profunda gratidão pela orientação sábia, paciência e dedicação ao longo deste projeto. Suas orientações foram fundamentais para o meu crescimento acadêmico e profissional.

Aos meus queridos colegas de turma, agradeço pela colaboração, troca de experiências e apoio mútuo. Compartilhar essa jornada com vocês tornou cada desafio mais significativo e cada conquista mais especial.

A todos os mencionados, meu sincero agradecimento. Este trabalho acadêmico é fruto não apenas do meu esforço, mas também do apoio valioso de cada uma dessas pessoas especiais em minha vida.

Eu, **Henrique de Souza Damião** quero primeiramente agradecer a minha família por sempre me apoiar nas minhas decisões, ter me ajudado em momentos mais difíceis nessa longa jornada, quero agradecer minha parceira por ter me ajudado, quero também deixar uma menção para minha mãe que não está entre nós, gostaria muito que ela tivesse visto meu trabalho.

## **APLICAÇÃO DA INTELIGÊNCIA ARTIFICIAL NO CONTROLE E GERENCIAMENTO DE ESTACIONAMENTOS PARA BICICLETAS**

Henrique de Souza Damião<sup>1</sup>

Renata Priscila Tavares de Matos<sup>2</sup>

Prof. Me. Sergio Clayton Viana Pinheiro<sup>3</sup>

**RESUMO:** No cenário atual, marcado pelo crescente congestionamento nas grandes cidades, há uma tendência em crescimento na escolha da bicicleta como um meio de transporte ágil e ecologicamente responsável. O propósito principal desta pesquisa consiste em criar um sistema destinado ao estacionamento de bicicletas da instituição SENAC, integrando inteligência artificial. Esse sistema proporcionará aos funcionários que utilizam bicicletas um acesso seguro e automatizado para armazenar seus veículos. Em resposta a essa demanda, propõe-se uma solução inovadora: um estacionamento de bicicletas com inteligência artificial. A introdução de tecnologias avançadas, como a inteligência artificial, busca não apenas otimizar a gestão e a segurança desse estacionamento, mas também contribuir para a eficiência global do sistema de transporte urbano. Ao elaborar o protótipo destinado à instituição SENAC, a pesquisa exige a aplicação de conceitos fundamentais, incluindo inteligência artificial, linguagens de programação, placas protoboard, Arduino, algoritmos disponíveis no mercado e processamento de imagens. Esses elementos desempenham um papel crucial ao longo do desenvolvimento do projeto, não apenas para atender às demandas específicas da instituição, mas também para contribuir significativamente para o aprimoramento contínuo dos processos internos do SENAC. Essa abordagem alinha-se com a busca incessante da instituição por inovação e excelência no setor educacional.

**Palavras-chave:** Inteligência artificial. SENAC. Estacionamento.

---

<sup>1</sup> Henrique de Souza Damião acadêmico do Curso de Bacharelado em Engenharia da Computação da Faculdade de Tecnologia do Amapá – META; E-mail: henriquedamiao1709@gmail.com

<sup>2</sup> Renata Priscila Tavares de Matos acadêmico do Curso de Bacharelado em Engenharia da Computação da Faculdade de Tecnologia do Amapá – META; E-mail: rptmpriscila33@gmail.com

<sup>3</sup> Prof. Mestre Sergio Clayton Viana Pinheiro em Ciências da Computação - UFPA, docente de Engenharia da Computação; E-mail:sergio@meta.edu.br

## APPLICATION OF ARTIFICIAL INTELLIGENCE IN THE CONTROL AND MANAGEMENT OF BICYCLE PARKING

Henrique de Souza Damião<sup>1</sup>

Renata Priscila Tavares de Matos<sup>2</sup>

Prof. Me. Sergio Clayton Viana Pinheiro<sup>3</sup>

**ABSTRACT:** In the current scenario marked by increasing congestion in large cities, there is a growing trend in choosing bicycles as a fast and environmentally responsible means of transportation. The main purpose of this research is to create a system for SENAC bicycle parking, integrating artificial intelligence. This system will provide employees who use bicycles with secure and automated access to store their vehicles. In response to this demand, an innovative solution is proposed: a bicycle parking system with artificial intelligence. The introduction of advanced technologies, such as artificial intelligence, aims not only to optimize the management and security of this parking facility but also to contribute to the overall efficiency of the urban transportation system. When developing the prototype for SENAC, the research requires the application of fundamental concepts, including artificial intelligence, programming languages, protoboard, Arduino, algorithms available in the market, and image processing. These elements play a crucial role throughout the project development, not only to meet the specific demands of the institution but also to contribute significantly to the continuous improvement of SENAC internal processes. This approach aligns with the institution's relentless pursuit of innovation and excellence in the education sector.

**Keywords:** Artificial intelligence. SENAC. Parking.

## LISTA DE ILUSTRAÇÕES

|                                                                                             |    |
|---------------------------------------------------------------------------------------------|----|
| <b>Figura 1-</b> Definições de Inteligência Artificial, organizadas em quatro categorias... | 15 |
| <b>Figura 2-</b> Placa arduino.....                                                         | 21 |
| <b>Figura 3-</b> Placa Protoboard.....                                                      | 22 |
| <b>Figura 4-</b> Servo Motor.....                                                           | 24 |
| <b>Figura 5-</b> Câmera Webcam.....                                                         | 24 |
| <b>Figura 6-</b> Protótipo.....                                                             | 25 |
| <b>Figura 7-</b> Fases do processo de captura da imagem e treinamento do sistema.....       | 28 |
| <b>Figura 8-</b> Código do programa de captura.....                                         | 28 |
| <b>Figura 9-</b> Capturando em execução.....                                                | 29 |
| <b>Figura 10-</b> Arquivos de imagens gerados pela captura.....                             | 30 |
| <b>Figura 11-</b> Código de treinamento.....                                                | 31 |
| <b>Figura 12-</b> Teste de reconhecimento do rosto.....                                     | 32 |
| <b>Figura 13-</b> Teste de reconhecimento com óculos.....                                   | 32 |
| <b>Figura 14-</b> Teste de reconhecimento sem óculos.....                                   | 33 |
| <b>Figura 15-</b> Reconhecimento com óculos bem sucedido.....                               | 34 |
| <b>Figura 16-</b> Registro do algoritmo.....                                                | 34 |
| <b>Figura 17-</b> Arduíno.....                                                              | 35 |
| <b>Figura 18-</b> Código de importação do arduino.....                                      | 36 |
| <b>Figura 19-</b> Led C ++.....                                                             | 37 |
| <b>Figura 20-</b> Led <i>python</i> .....                                                   | 37 |
| <b>Figura 21-</b> Código do servo motor em C++.....                                         | 38 |
| <b>Figura 22-</b> Código do servo motor em <i>python</i> .....                              | 39 |
| <b>Figura 23-</b> Importações.....                                                          | 40 |
| <b>Figura 24-</b> Reconhecimento.....                                                       | 41 |
| <b>Figura 25-</b> Monitoramento.....                                                        | 41 |
| <b>Figura 26-</b> <i>Main</i> .....                                                         | 42 |
| <b>Figura 27-</b> Estrutura de madeira compensado.....                                      | 43 |
| <b>Figura 28-</b> Conexões do protótipo.....                                                | 44 |
| <b>Figura 29-</b> Fechadura e servo motor conectados.....                                   | 45 |
| <b>Figura 30-</b> Sistema de trava aberto.....                                              | 46 |
| <b>Figura 31-</b> Sistema de trava fechado.....                                             | 47 |

## Sumário

|                                                       |           |
|-------------------------------------------------------|-----------|
| <b>1 INTRODUÇÃO.....</b>                              | <b>10</b> |
| 1.1 POSICIONAMENTO E MOTIVAÇÃO.....                   | 10        |
| 1.2 OBJETIVOS.....                                    | 11        |
| 1.3 DEFINIÇÃO DO PROBLEMA.....                        | 12        |
| <b>2 REFERENCIAL TEÓRICO.....</b>                     | <b>13</b> |
| 2.1 INTELIGENCIA ARTIFICIAL.....                      | 13        |
| 2.2 APRENDIZADO DE MÁQUINA.....                       | 15        |
| 2.3 RECONHECIMENTO DE IMAGEM À GERAÇÃO DE IMAGEM..... | 17        |
| 2.4 LINGUAGEM DE PROGRAMAÇÃO <i>PYTHON</i> .....      | 18        |
| 2.5 LINGUAGEM DE PROGRAMAÇÃO C++.....                 | 19        |
| <b>3 MATERIAL E MÉTODOS.....</b>                      | <b>21</b> |
| 3.1 ARDUINO.....                                      | 21        |
| 3.2 PLACA PROTOBOARD.....                             | 22        |
| 3.3 SERVO MOTOR.....                                  | 24        |
| 3.4 CÂMERA WEBCAM.....                                | 25        |
| 3.5 ESTRUTURA DE SUSTENTAÇÃO DO PROTÓTIPO.....        | 26        |
| <b>4 ANÁLISE E DISCUSSÃO DOS RESULTADOS.....</b>      | <b>28</b> |
| <b>5 CONSIDERAÇÕES FINAIS.....</b>                    | <b>49</b> |
| <b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>                | <b>50</b> |

## 1 INTRODUÇÃO

De acordo com Ricardo Cappra (2021), “aprender na era da informação tornou-se uma habilidade no mínimo desafiadora”. Ao longo das últimas décadas, testemunhamos uma rápida evolução tecnológica, impulsionando-nos para a chamada 'era da informação'. Nesse contexto, é cada vez mais evidente a necessidade de incorporar ferramentas inteligentes em nosso cotidiano, a fim de agilizar e aprimorar a execução de tarefas de maneira eficiente .

O surgimento de tecnologias inovadoras em diferentes áreas tem o potencial de transformar a forma como as pessoas vivem, trabalham e se relacionam. Tais tecnologias são geralmente baseadas em avanços em áreas como ciência da computação, eletrônica, materiais, biotecnologia e outras. Já que as pessoas estão cada vez mais submetidas e dependentes da tecnologia, a realidade é que as inovações tecnológicas, como a Inteligência Artificial (IA), estão presentes em áreas que muitas vezes nem imaginam.

Segundo informações do site Autentify (2023), “a tecnologia está mudando rapidamente e novas inovações estão surgindo o tempo todo. Entre as tecnologias mais recentes, podemos citar a inteligência artificial, o *blockchain*, a Internet das Coisas (IoT) e a realidade virtual”. Essas tecnologias estão transformando a forma como fazemos negócios, nos relacionamos, aprendemos e nos divertimos. Elas estão sendo utilizadas em diferentes setores, incluindo saúde, finanças, transporte, educação, entretenimento e muitos outros. Os profissionais da área de engenharia de computação reconhecem que é possível desenvolver soluções inovadoras que facilitem o cotidiano das pessoas e instituições, como responsáveis por essa tarefa, eles trabalham na criação de soluções acessíveis e viáveis para enfrentar os desafios do dia a dia. Dessa forma, contribuem para melhorar a qualidade de vida das pessoas e aumentar a eficiência das instituições.

### 1.1 POSICIONAMENTO E MOTIVAÇÃO

A bicicleta é um meio de transporte escolhido por muitos países há anos por ser sustentável e auxiliar na atividade física. “É necessário um planejamento para tornar a cidade mais agradável, viável e segura para o deslocamento de bicicleta pela população, pois os ciclistas disputam espaço com os automóveis e dependendo da via, não sobra espaço para o uso da bicicleta, e os locais

específicos, raramente são vistos, tornando um risco de vida o deslocamento pelas cidades”(Aozani, 2023).

Para esse transporte um estacionamento com inteligência artificial é uma solução inovadora para promover o uso em áreas urbanas. Segundo a Revista Bicicleta (2023), “Estudo brasileiro prevê aumento de 18% na prática do ciclismo até 2030, com o aumento do trânsito nas grandes cidades, muitas pessoas estão optando por utilizar um meio de transporte mais sustentável”.

A motivação para a execução dessa iniciativa originou-se da carência de estacionamentos seguros para bicicletas, representando um desafio significativo para vários ciclistas, suscetível a roubos e danos materiais. Com isso, foi identificado, na instituição de Serviço Nacional de Aprendizagem Comercial – SENAC, um estacionamento para esse meio de transporte com um sistema considerado obsoleto para uma corporação de grande porte.

## 1.2 OBJETIVOS

O propósito principal desta pesquisa consiste em desenvolver um sistema destinado ao estacionamento para bicicletas da empresa/instituição SENAC, integrando inteligência artificial. Esse sistema proporcionará aos funcionários que utilizam o transporte mencionado anteriormente com um acesso seguro e automatizado para armazenar seus veículos. A implementação utilizará tecnologia avançada de reconhecimento de imagem, assegurando a identificação precisa dos proprietários dos veículos. Dessa forma, somente indivíduos autorizados terão permissão para acessar e retirar o seu respectivo transporte do estacionamento.

A instituição SENAC, ao equipar seu estacionamento de bicicletas com IA, pode oferecer diversos benefícios aos seus funcionários e visitantes. Em primeiro lugar, a adoção da tecnologia de reconhecimento de imagem pode melhorar substancialmente a segurança, assegurando que apenas os proprietários autorizados tenham acesso ao estacionamento. Ademais, a automação do processo de armazenamento e retirada torna a experiência mais eficiente e conveniente para os usuários.

Os objetivos específicos para o desenvolvimento de um estacionamento com inteligência artificial, variam dependendo das necessidades e metas específicas do local onde o estacionamento será instalado. Ademais, para o funcionamento da pesquisa, algumas tarefas precisam ser executadas, tais como:

- Construir um protótipo no qual simula uma vaga no estacionamento. O primeiro passo consiste em definir os requisitos e funcionalidades desejadas. Este processo inclui a determinação da capacidade do estacionamento e a estipulação de critérios para a alocação de vagas, entre outros aspectos relevantes;
- Desenvolver um programa nas linguagens C++ e Python, a equipe tem controle total sobre o sistema de vagas, proporcionando maior flexibilidade e adaptabilidade.
- Aprimorar a segurança das bicicletas, buscando oferecer uma solução segura para os funcionários ciclistas armazenarem seus veículos. Tal atividade envolve a substituição do sistema atual, baseado em um caderno de protocolo para coleta de assinaturas e a eliminação do uso de cartões de responsabilidade de estacionamento. Serão incorporadas medidas de segurança aprimoradas para detectar e prevenir tentativas de roubo e vandalismo.

### 1.3 DEFINIÇÃO DO PROBLEMA

O problema investigado nesta pesquisa visa apresentar uma solução viável para as vagas de bicicletas na instituição SENAC. As dificuldades enfrentadas na organização para a ocupação dessas vagas incluem a falta de flexibilidade. A solução proposta envolve a implementação de uma tecnologia de inteligência artificial (IA) no local.

As alternativas disponíveis para abordar essa questão incluem aumentar o número de estacionamentos e vagas correspondentes, além da implementação do conceito de estacionamento inteligente, que é o foco principal deste estudo. Dessa forma, busca-se otimizar o processo e evitar que os ciclistas percam tempo significativo procurando vagas disponíveis.

## 2 REFERENCIAL TEÓRICO

A presente pesquisa tem como cenário de aplicação a instituição de Serviço Nacional de Aprendizagem Comercial - SENAC, reconhecida por sua atuação no setor educacional. Fundada em 10 de janeiro de 1946, ela tem desempenhado um papel crucial no desenvolvimento de treinamentos, cursos, capacitações, workshops e palestras. Com uma sólida reputação no mercado, busca constantemente inovações que aprimorem seus processos internos e contribuam para sua missão institucional.

No contexto específico deste estudo, a instituição SENAC representa um ambiente propício para a implementação do projeto proposto. A compreensão aprofundada das práticas e desafios enfrentados pela organização proporciona uma base sólida para a identificação de oportunidades de melhoria.

Para a elaboração do protótipo do projeto destinado ao local escolhido para execução do estudo, é crucial adquirir um entendimento abrangente, e aplicar conceitos fundamentais que serão discutidos neste capítulo e empregados ao longo do desenvolvimento. Esses conceitos englobam inteligência artificial, linguagens de programação, placas *protoboard*, Arduino, algoritmos relevantes disponíveis no mercado, e o processamento de imagens, desempenhando um papel auxiliar de grande importância.

### 2.1 INTELIGENCIA ARTIFICIAL

Conforme o *Google Cloud* (2024), “a inteligência artificial (IA) é um campo da ciência preocupado com a criação de computadores e máquinas que podem raciocinar, aprender e atuar de maneira que normalmente exigiria inteligência humana ou que envolvia dados com escala maior que as pessoas podem analisar”.

Segundo Gomes (2010, p.2), a criação da Inteligência Artificial envolve uma longa história de desenvolvimento e pesquisa em diversas áreas do conhecimento, como matemática, lógica, filosofia, neurociência e ciência da computação.

“Foi na década de 1950 que o termo "Inteligência Artificial" foi criado e a pesquisa na área ganhou impulso” PRADO (2019, p.4). Durante essa época, muitos pesquisadores acreditavam que a IA poderia ser alcançada por meio da programação de regras lógicas. Nos anos 80 e 90, o foco da IA mudou para o aprendizado de máquina, que é um método de programação em que os computadores aprendem por conta própria a partir de dados e experiências

passadas. Esse método levou a avanços significativos em áreas como o reconhecimento de fala, reconhecimento de imagem e processamento de linguagem natural.

Ao longo desses anos alguns nomes e marcos importante na evolução da inteligência artificial incluem o seguinte:

1950: Alan Turing publica *Computing Machinery and Intelligence*. No artigo, Turing, famoso por desvendar o código nazista ENIGMA durante a segunda guerra mundial, propõe responder à pergunta "as máquinas podem pensar?" e apresenta o Teste de Turing para determinar se um computador pode demonstrar a mesma inteligência (ou os resultados da mesma inteligência) que um humano. O valor do teste de Turing tem sido debatido desde então.

1956: John McCarthy cunha o termo 'inteligência artificial' na primeira conferência de IA no Dartmouth College. (McCarthy continuaria a inventar a linguagem Lisp). Mais tarde naquele ano, Allen Newell, J.C.Shaw, e Herbert Simon criam o Logic Theorist, o primeiro programa de software de IA em execução.

1967: Frank Rosenblatt constroi o Mark 1 Perceptron, o primeiro computador com base em uma rede neural que "aprende" por tentativa e erro. Apenas um ano depois, Marvin Minsky e Seymour Papert publicaram um livro intitulado Perceptrons, que se torna o trabalho de referência em redes neurais e, pelo menos por um tempo, um argumento contra futuros projetos de pesquisa de redes neurais.

Em linhas gerais, o termo de inteligência artificial (IA) se divide em oito definições, dispostas em quatro dimensões. As duas primeiras se relacionam a processos de pensamento e raciocínio, enquanto as outras duas medem o sucesso em termos de fidelidade ao desempenho humano. Russell (2020). Na figura 1 podemos observar essas quatro definições em um quadro organizado.

As definições do lado esquerdo como mostra na figura 1 medem o sucesso em termos de fidelidade ao desempenho humano, enquanto as definições do lado direito medem o sucesso comparando-o a um conceito ideal de inteligência, chamado de racionalidade.” Um Sistema é racional se faz a coisa certa, dado o que ele sabe”. Russell, S., & Norvig (2020, p.20 ).

**FIGURA 1 - DEFINIÇÕES DE INTELIGÊNCIA ARTIFICIAL, ORGANIZADAS EM QUATRO CATEGORIAS.**

| Pensando como um humano                                                                                                                                                                                                                                                                                                                 | Pensando racionalmente                                                                                                                                                                                       |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>“O novo e interessante esforço para fazer os computadores pensarem (...) <i>máquinas com mentes</i>, no sentido total e literal.” (Haugeland, 1985)</p> <p>“[Automatização de] atividades que associamos ao pensamento humano, atividades como a tomada de decisões, a resolução de problemas, o aprendizado...” (Bellman, 1978)</p> | <p>“O estudo das faculdades mentais pelo uso de modelos computacionais.” (Charniak e McDermott, 1985)</p> <p>“O estudo das computações que tornam possível perceber, raciocinar e agir.” (Winston, 1992)</p> |
| Agindo como seres humanos                                                                                                                                                                                                                                                                                                               | Agindo racionalmente                                                                                                                                                                                         |
| <p>“A arte de criar máquinas que executam funções que exigem inteligência quando executadas por pessoas.” (Kurzweil, 1990)</p> <p>“O estudo de como os computadores podem fazer tarefas que hoje são melhor desempenhadas pelas pessoas.” (Rich and Knight, 1991)</p>                                                                   | <p>“Inteligência Computacional é o estudo do projeto de agentes inteligentes.” (Poole <i>et al.</i>, 1998)</p> <p>“AI... está relacionada a um desempenho inteligente de artefatos.” (Nilsson, 1998)</p>     |

Fonte: <https://www.cin.ufpe.br/~gtsa/Periodo/PDF/4P/SI.pdf>

Historicamente, todas as quatro estratégias para o estudo da IA têm sido seguidas, cada uma delas por pessoas diferentes com métodos diferentes. “Uma abordagem centrada nos seres humanos deve ser em parte uma ciência empírica, envolvendo hipóteses e confirmação experimental. Uma abordagem racionalista envolve uma combinação de matemática e engenharia”. Russell, S., & Norvig, (2020, p. 18).

Ao fazermos distinção entre comportamento humano e racional, não estamos sugerindo que os seres humanos sejam necessariamente “irracionais” no sentido de “emocionalmente instáveis” ou “insanos”. Russell (2018, p.25) “Simplesmente precisamos observar que não somos perfeitos: nem todos os jogadores de xadrez são grandes mestres e, infelizmente, nem todos os seres humanos conseguem conceito A nos exames”. Alguns erros sistemáticos do raciocínio humano estão catalogados em Kahneman et al.

## 2.2 APRENDIZADO DE MÁQUINA

As máquinas estão aprendendo a dirigir automóveis. Já existem carros que podem se mover sem motorista, como os carros do Google e da Tesla. A liberação do uso de alguns desses carros, de forma totalmente autônoma e independente, está aguardando legislação adequada e a realização de mais testes em ambientes reais, assim como em situações excepcionais de alto risco. Tais sistemas devem ser

robustos o suficiente para lidar com situações para as quais não tenham sido treinados, assim como situações que possam colocar as vidas dos motoristas e pedestres em risco (Feng et al., 2021).

A aprendizagem computacional ou aprendizagem de máquina, oferece aos engenheiros e cientistas de computação um paradigma, para desenvolvimento de software, bastante diferente dos tradicionais. Em linhas gerais, neste paradigma não cabe mais ao desenvolvedor a tarefa de encontrar diretamente a solução para um problema específico, mas apenas a de criar um ambiente computacional que permita que esta solução seja automaticamente (ou semi-automaticamente) induzida a partir de exemplos de como instâncias particulares do problema são resolvidas.

Quando falamos sobre Inteligência Artificial (IA), existe um elemento essencial frequentemente abordado que é o aprendizado de Máquina. É verdade que existem alguns assuntos complexos nesse ramo, mas o entendimento básico é muito simples, e a utilização da tecnologia também.

“O aprendizado de máquina (em inglês, *machine learning*) é um método de análise de dados que automatiza a construção de modelos analíticos”. (Izbicki e Santos, 2020). É um ramo da inteligência artificial baseado na ideia de que sistemas podem aprender com dados, identificar padrões e tomar decisões com o mínimo de intervenção humana. O reconhecimento de imagem e o aprendizado de máquina são duas áreas inter-relacionadas da inteligência artificial que frequentemente trabalham juntas para solucionar problemas complexos relacionados a imagens e dados visuais.

Além do aprendizado de máquina, temos a visão computacional que se refere à capacidade dos sistemas de compreender e interpretar informações visuais, como imagens e vídeos. Isso inclui tarefas como detecção e reconhecimento de objetos, segmentação de imagens, rastreamento de movimento e análise de cena. A *machine learning* é uma abordagem que permite que os sistemas de Inteligência Artificial aprendam a partir de dados e identifiquem padrões, sem serem explicitamente programados. É especialmente útil quando se trata de lidar com grandes volumes de dados, como imagens, e extrair informações significativas a partir deles.

Portanto, o aprendizado de máquina e o reconhecimento de imagem trabalham em conjunto para capacitar os sistemas de IA a compreender e interpretar o conteúdo visual, permitindo uma ampla gama de aplicações em campos como

visão computacional, diagnóstico médico, veículos autônomos, segurança, entre outros. No entanto, é importante lembrar que a IA é um campo em constante evolução, e novos conceitos e tecnologias podem surgir ao longo do tempo.

### 2.3 RECONHECIMENTO DE IMAGEM À GERAÇÃO DE IMAGEM.

“O reconhecimento de imagem é uma área da inteligência artificial (IA) que envolve a capacidade de os sistemas computacionais interpretar e compreenderem o conteúdo de imagens digitais ou vídeos” (Russakovsky et al., 2015). Essa tecnologia permite que os computadores identifiquem objetos, pessoas, cenários e padrões visuais em imagens, tornando possível uma série de aplicações práticas.

“Um dos métodos mais comuns utilizados para o reconhecimento de imagem é a chamada aprendizagem profunda (deep learning), que é uma subárea do campo de IA” (LeCun et al., 2015). Especificamente, as redes neurais convolucionais (CNNs) têm sido amplamente aplicadas e são altamente eficazes para tarefas de reconhecimento de imagem.

“As CNNs (Convolutional Neural Networks) são projetadas para imitar o funcionamento do sistema visual humano, processando imagens em camadas hierárquicas” (Krizhevsky et al., 2012). Cada camada é composta por um conjunto de neurônios, que são responsáveis por extrair características específicas das imagens, como bordas, texturas e formas. À medida que as informações são passadas pelas camadas, a rede neural aprende a identificar padrões complexos e distinguir diferentes objetos ou características nas imagens.

“Uma vez que uma rede neural convolucional tenha sido treinada em um grande conjunto de dados rotulados, ela pode ser aplicada a novas imagens para identificar e classificar objetos de interesse” (Simonyan & Zisserman, 2014). Por exemplo, um sistema de reconhecimento de imagem treinado pode ser capaz de identificar pessoas, veículos, animais, sinais de trânsito, entre outros elementos em imagens capturadas por câmeras.

Além do reconhecimento de imagem, as câmeras equipadas com inteligência artificial (IA) também podem gerar imagens sintéticas realistas. Goodfellow (2014) argumenta que por meio de técnicas como a geração adversarial de redes neurais (GANs), é possível treinar redes para gerar imagens que se assemelham a um determinado conjunto de dados de treinamento. Essas imagens sintéticas podem ser

utilizadas em várias aplicações, como jogos, simulações, design de produtos, efeitos especiais em filmes e muito mais.

No contexto de segurança, as câmeras com IA também podem ser usadas para detecção de anomalias ou comportamentos suspeitos em imagens de vigilância. Por exemplo, um sistema de monitoramento pode ser treinado para identificar movimentos não autorizados em áreas restritas ou detectar objetos abandonados.

“O reconhecimento de imagem e a geração de imagens com câmeras equipadas com IA têm possibilitado avanços significativos em diversas áreas, desde a identificação de objetos e pessoas em imagens até a criação de imagens sintéticas realistas. Essas tecnologias têm aplicações em setores como segurança, entretenimento, automação industrial, medicina, transporte, entre outros, desempenhando um papel importante na transformação digital e no desenvolvimento de soluções inovadoras”. (Desafios e Soluções, 2023)

## 2.4 LINGUAGEM DE PROGRAMAÇÃO PYTHON

“*Python*, uma linguagem de programação de alto nível, interpretada e de propósito geral, destaca-se por sua sintaxe simples e legibilidade, tornando-se uma escolha popular entre desenvolvedores de software em diversas áreas, incluindo inteligência artificial” (*Python Software Foundation*, 2022). No contexto do desenvolvimento de inteligência artificial, *Python* desempenha um papel crucial por várias razões.

A facilidade de uso é uma característica distintiva de *Python*, sendo acessível a programadores iniciantes e experientes. “Sua sintaxe clara e concisa permite que os desenvolvedores escrevam código eficientemente, mantendo uma maior legibilidade” (*Python Software Foundation*, 2022).

“*Python* conta com uma vasta biblioteca de inteligência artificial, incluindo frameworks renomados como *TensorFlow*, *PyTorch*, *scikit-learn* e *Keras*” (Abadi et al., 2016; Pedregosa et al., 2011; Chollet & Others, 2015). Essas bibliotecas oferecem funções e algoritmos prontos para uso, facilitando o desenvolvimento de modelos de IA e a implementação de técnicas avançadas.

Além disso, a capacidade de integração de *Python* com outras tecnologias e linguagens é notável. Isso permite que os desenvolvedores combinem ferramentas

de IA com soluções diversas, como bancos de dados, APIs *web* e interfaces gráficas, criando sistemas mais complexos e abrangentes” (Pedregosa et al., 2011).

Em poucas palavras, a linguagem *Python* é amplamente adotada no desenvolvimento de inteligência artificial, oferecendo facilidade de uso, vasta biblioteca de IA, uma comunidade ativa e a capacidade de integração com outras tecnologias. Essa linguagem fornece uma base sólida para a implementação de algoritmos de aprendizado de máquina, construção de modelos de IA e desenvolvimento de soluções avançadas nessa área.

## 2.5 LINGUAGEM DE PROGRAMAÇÃO C++

“C++ é uma linguagem de programação de propósito geral e de alto desempenho. Ela é conhecida por sua eficiência, flexibilidade e ampla aplicabilidade em uma variedade de domínios, incluindo o desenvolvimento de inteligência artificial” (Stroustrup, 2013).

No contexto do desenvolvimento de inteligência artificial, C++ desempenha um papel importante devido a várias razões:

- **Desempenho:** C++ é uma linguagem compilada que permite um controle mais direto sobre o hardware subjacente. Isso significa que é possível escrever código C++ altamente otimizado, aproveitando ao máximo os recursos do sistema. Para tarefas intensivas em termos de computação, como o treinamento de modelos complexos de IA, a eficiência do C++ pode ser uma vantagem significativa.
- **Bibliotecas e *Frameworks*:** Existem várias bibliotecas e *frameworks* de IA populares escritos em C++, como o *TensorFlow* e o *OpenCV*. Essas bibliotecas fornecem funcionalidades prontas para uso e algoritmos otimizados para tarefas específicas de IA, como processamento de imagem, visão computacional e aprendizado de máquina. O uso de C++ nessas bibliotecas permite um desempenho eficiente e uma integração suave com outros componentes do sistema.
- **Integração com outras linguagens:** C++ possui recursos de integração que permitem que ele seja combinado com outras linguagens de programação. Isso é particularmente útil em projetos de IA que exigem a combinação de diferentes tecnologias e componentes. Por exemplo, é possível utilizar bibliotecas escritas em *Python* para tarefas de processamento de dados e IA,

enquanto o núcleo computacional é implementado em C++ para obter um desempenho mais eficiente.

- **Ecossistema de desenvolvimento:** C++ possui um ecossistema de desenvolvimento maduro e uma comunidade ativa, com ampla documentação, recursos de suporte e uma base de código aberto rica em projetos relacionados à IA. Isso facilita a colaboração, o aprendizado e a resolução de problemas ao desenvolver soluções de IA em C++.

(Stroustrup, 2013) “Destaca que a linguagem de programação C++ é uma linguagem poderosa e eficiente que desempenha um papel significativo no desenvolvimento de inteligência artificial”. Sua capacidade de oferecer desempenho de alto nível, integração com bibliotecas de IA e sua flexibilidade para trabalhar com outras linguagens tornam o C++ uma escolha popular para o desenvolvimento de componentes essenciais de IA.

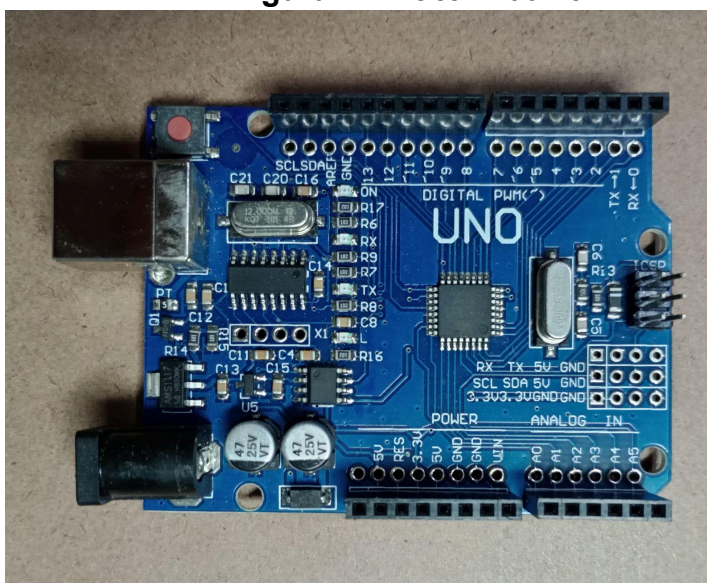
### 3 MATERIAL E MÉTODOS

Neste capítulo, abordam-se as tecnologias empregadas no desenvolvimento do projeto, contemplando tanto hardware quanto *software*, juntamente com a metodologia de pesquisa adotada.

#### 3.1 ARDUINO

“A placa Arduino é uma plataforma de prototipagem eletrônica de código aberto, projetada para facilitar o desenvolvimento de projetos eletrônicos de forma acessível e flexível. Ela consiste em uma placa de circuito com um microcontrolador programável e uma interface de entrada/saída que permite conectar sensores, atuadores e outros dispositivos eletrônicos” (Arduino, 2019) como mostra na figura 2,

**Figura 2** - Placa Arduino



Fonte: Autores, 2024.

“No contexto do desenvolvimento de inteligência artificial, a placa Arduino pode desempenhar um papel complementar, especialmente em projetos que envolvem integração de *hardware* e controle de sensores” (Shin et al., 2018). Embora o Arduino em si não seja capaz de executar algoritmos de IA complexos, ele pode ser usado em conjunto com outras plataformas e tecnologias para criar sistemas de IA interativos e incorporados.

“A placa Arduino pode ser usada para coletar dados de sensores, como dados ambientais, movimento, som, entre outros, e enviá-los para um computador ou

plataforma de processamento de IA” (Alippi et al., 2018). Esses dados podem ser utilizados para treinar modelos de IA, ajustar parâmetros e realizar inferências. A placa Arduino também pode ser usada para controlar atuadores, como motores, servos e LEDs, permitindo que um sistema de IA interaja com o ambiente físico.

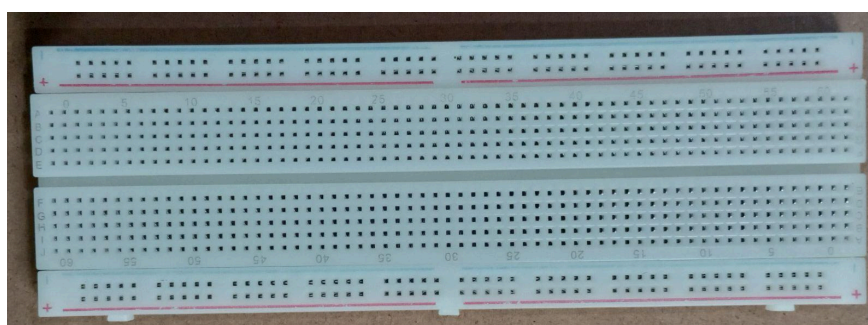
Além disso, a placa “Arduino é compatível com uma variedade de *shields* (módulos de expansão) que podem adicionar funcionalidades específicas ao projeto, como módulos *Wi-Fi*, *Bluetooth*, câmeras, entre outros” (Soloman, 2011). Esses recursos de conectividade e expansibilidade permitem que a placa Arduino seja integrada a outros componentes de IA, como plataformas de processamento de dados em nuvem ou microcontroladores mais avançados.

Embora a placa Arduino por si só não execute algoritmos complexos de IA, ela desempenha um papel importante no desenvolvimento de sistemas de IA incorporados e interativos, fornecendo capacidades de entrada/saída, coleta de dados de sensores e controle de atuadores. Ao ser integrada com outras tecnologias e plataformas, a placa Arduino pode contribuir para a criação de soluções de IA que interagem com o mundo físico.

### 3.2 PLACA PROTOBOARD

“*Protoboard*, matriz de contato, *breadboard* e placa de ensaio. Vários nomes para uma das peças mais importantes quando o assunto é montagem de circuitos elétricos” (Mota, 2018). Seja para fins educacionais ou até para o desenvolvimento de protótipos iniciais de produtos em empresas, a protoboard sempre estará presente. Ela consiste em uma matriz de furos, onde é possível inserir fios, pinos e componentes, formando circuitos elétricos como mostra na figura 3.

**Figura 3 - Placa Protoboard**



Fonte: Autores, 2024

Segundo Johnson (2018), no desenvolvimento de inteligência artificial, a placa de ensaio desempenha um papel importante ao permitir a montagem e conexão dos diversos componentes eletrônicos necessários para criar um sistema de IA. Ela oferece uma maneira prática de prototipar e testar circuitos antes de serem implementados de forma definitiva.

“Através da placa protoboard, é possível conectar componentes como sensores, atuadores, microcontroladores, placas de desenvolvimento, entre outros, fundamentais para a coleta de dados, processamento de informações e interação com o ambiente, aspectos essenciais na implementação de sistemas de inteligência artificial” (Smith, 2020).

“Ao utilizar a placa, os desenvolvedores de IA podem realizar experimentos, testes e ajustes nos circuitos, facilitando a integração dos componentes e permitindo a criação de soluções personalizadas” (Jones et al., 2019). Essa flexibilidade e versatilidade no prototipamento contribuem para o desenvolvimento iterativo e aprimoramento contínuo de sistemas de IA.

A placa *breadboard* desempenha um papel crucial no desenvolvimento do sistema de trava segurança, fornecendo uma plataforma flexível e conveniente para montagem e conexão de componentes eletrônicos. Ela facilita o prototipamento, testes e a implementação de soluções personalizadas, contribuindo para o avanço e a aplicação prática da IA.

Ao reconsiderar o sistema de trava, foram exploradas diversas opções. Inicialmente, cogitamos utilizar um motor de carrinho de controle remoto, mas percebemos que seria necessário um sistema de engrenagens. Para prototipar essa ideia, utilizamos o software *TinkerCAD*. No entanto, devido à falta de experiência na fabricação de peças com impressora 3D e à indisponibilidade deste serviço em nossa região, decidimos descartar esse plano.

### 3.3 SERVO MOTOR

Por conseguinte, foi considerada a possibilidade de empregar um sistema magnético, contudo, verificou-se que seu consumo de energia seria elevado e não estava disponível para integração com o Arduino. Após uma pesquisa mais detalhada, identificou-se uma solução adequada para o propósito almejado: o *Tower Pro Micro Servo 9g*, conforme ilustrado na figura 8. Este sistema atende aos

requisitos do projeto, proporcionando a funcionalidade necessária para a trava do estacionamento de bicicletas.

**Figura 4 - Micro servo motor**



Fonte: Autores, 2024

### 3.4 CÂMERA WEBCAM

A câmera selecionada para o projeto é uma câmera *webcam* Full HD 30FPS, com a capacidade de realizar a captura de imagens em tempo real, aprimorando assim a identificação de padrões. A sua utilização no projeto envolve a integração eficiente com o sistema de inteligência artificial por meio de uma interface de comunicação dedicada, a qual suporta a transmissão contínua de dados em tempo real. Este recurso permite que a câmera contribua de maneira significativa para a funcionalidade geral do projeto, ao proporcionar uma entrada visual precisa e em tempo real para o sistema de inteligência artificial, fortalecendo assim as capacidades de reconhecimento e processamento de informações do dispositivo.

**Figura 5** - Câmera web c3tech wb-100bk full hd 30fps cor preto



Fonte: <https://www.c3technology.com.br/webcam-full-hd-1080p-wb-100bk-c3t/p>

### 3.5 ESTRUTURA DE SUSTENTAÇÃO DO PROTÓTIPO

A estrutura de sustentação do protótipo foi meticulosamente projetada para proporcionar estabilidade e suporte essenciais ao dispositivo. Composta por duas sustentações de madeira, esta parte crucial do protótipo desempenha um papel fundamental no encaixe e suporte da roda da bicicleta. Ambas as sustentações são confeccionadas com compensado de 15mm, um material escolhido não apenas por sua durabilidade, mas também pela sua maleabilidade, possibilitando ajustes conforme necessário.

Além disso, a estrutura abriga componentes eletrônicos essenciais, como a placa de protoboard e suas conexões, a placa Arduino integrada ao sistema *Python* e o micro servo motor de 9g com um ferrolho para a operação da trava. Esses elementos são cuidadosamente posicionados para garantir eficiência operacional e durabilidade da estrutura de sustentação, tornando-a uma parte integral e funcional do projeto do protótipo como mostrado na figura 6.

**Figura 6 - Protótipo**



Fonte: Autores, 2024

A disposição estratégica das sustentações visa acomodar a roda da bicicleta de forma eficiente, garantindo que o dispositivo funcione harmoniosamente. A escolha da madeira como material de construção leva em consideração não apenas sua disponibilidade econômica, mas também sua facilidade de modificação, permitindo ajustes precisos para se adequar às necessidades específicas do protótipo.

## 4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

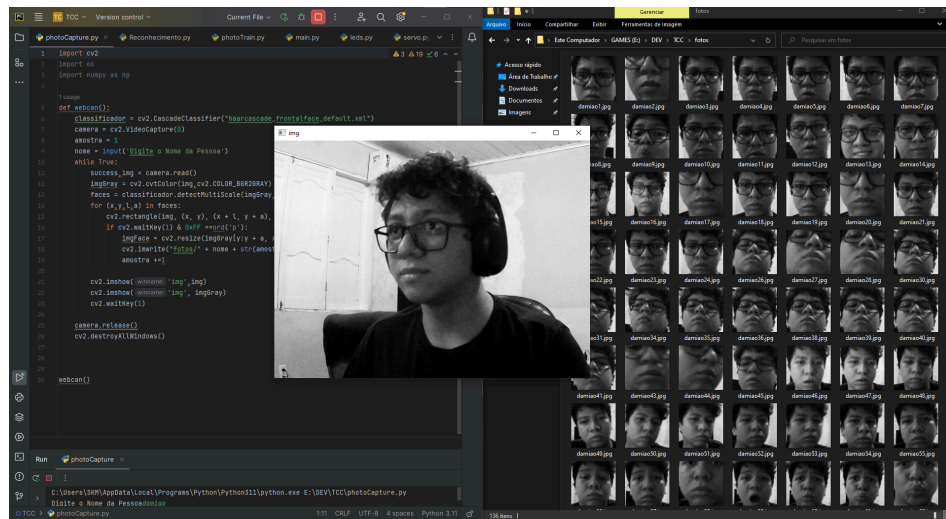
Inicialmente, para o desenvolvimento da pesquisa, foram utilizados materiais como um *notebook* da marca *Samsung*, modelo NP270E5, equipado com um processador *Intel Core i5* de 4ª geração e 8 GB de memória RAM. Apesar de possuir uma câmera integrada de 2MP, considerada de qualidade inferior para identificar imagens, foi utilizada uma câmera *Webcam* modelo WB-100bk, ela proporciona uma alta resolução que garante qualidade nas captações de vídeos. Além do microfone embutido e da redução de ruídos que auxiliam na comunicação.

Para estabelecer a conexão entre o smartphone e o notebook, foi utilizado o programa e aplicativo *DroidCam App*, desenvolvido pela Dev47Apps. No entanto, devido ao alto consumo de energia pelo smartphone, buscamos uma câmera de melhor qualidade para proporcionar um reconhecimento mais preciso das imagens capturadas. Atualmente, encontra-se em andamento o desenvolvimento de um sistema de reconhecimento facial utilizando o programa PyCharm. Até o momento, foram empregadas as bibliotecas *OpenCV* e *NumPy*. Segundo Passarelli (2017), “a biblioteca *OpenCV* destaca-se como uma poderosa ferramenta de processamento de imagem de código aberto, permitindo a criação de aplicativos de visão computacional em diversas linguagens de programação e sistemas operacionais”.

Konzen (2024, p.27) a biblioteca *NumPy* é extensivamente utilizada na linguagem *Python* para computação numérica, proporcionando suporte para operações em arrays multidimensionais, conhecidos como *ndarrays*. Essas bibliotecas desempenham papéis fundamentais no processamento e análise de imagens necessárias para a implementação do reconhecimento facial.

A figura 7 ilustra todas as fases e os processos, desde a captura pela máquina até o treinamento do sistema.

**Figura 7 - Fases do processo de captura da imagem e treinamento do sistema**



Fonte: Autor, 2024

Na figura 8, pode-se observar o código que capta as imagens do usuário para armazenar em um diretório, onde dentro dele possui várias imagens com diferentes ângulos para ter uma precisão ao reconhecer o rosto.

**Figura 8 - Código do programa de captura**

```
import cv2
import os
import numpy as np

1 usage
def webcam():
    classificador = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
    camera = cv2.VideoCapture(0)
    amostra = 1
    nome = input('Digite o Nome da Pessoa')
    while True:
        success, img = camera.read()
        imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        faces = classificador.detectMultiScale(imgGray, scaleFactor=1.5, minSize=(50, 50))
        for (x, y, l, a) in faces:
            cv2.rectangle(img, (x, y), (x + l, y + a), (1, 255, 0), 2)
            if cv2.waitKey(1) & 0xFF == ord('p'):
                imgFace = cv2.resize(imgGray[y:y + a, x:x + l], (220, 220))
                cv2.imwrite("fotos/" + nome + str(amostra) + ".jpg", imgFace)
                amostra += 1

        cv2.imshow('img', img)
        cv2.imshow('img', imgGray)
        cv2.waitKey(1)

    camera.release()
    cv2.destroyAllWindows()

webcam()
```

Fonte: Autores, 2024

Na figura 9, nota-se que o sistema de captura de imagens está funcionando corretamente, tendo uma abundância de imagem sobre o “damião.jpg” com algumas expressões faciais.

**Figura 9** - Capturando em execução



Fonte: Autor, 2024

Na figura 10, observa-se demasiadas imagens e dados para o treinamento da máquina, a fim de transformar essa imagem em código de reconhecimento do rosto do usuário.

**Figura 10** - Arquivos de imagens gerado pela capturar



Fonte: Autor, 2024

Após captar as imagens do usuário, essa imagem vai passar para o treinamento da máquina, assim foi construído um código (na figura 11) para ensinar a máquina a reconhecer a anatomia do rosto humano, com permissão para ter acesso à trava.

Neste código, o sistema realiza a verificação das imagens capturadas pelo código. Ele verifica se os nomes dos arquivos estão presentes na pasta para iniciar o processo de treinamento. Além disso, examina se as fotos estão no formato de escala de cinza, pois esse método foi empregado para aumentar a precisão no reconhecimento facial.

Subsequentemente, o código percorre três arquivos .yml, que foram utilizados para transformar as fotos em códigos na qual definem as métricas do rosto. Cada

rosto possui dimensões específicas, como distância entre os olhos, boca, nariz, e outras características distintas. Essa individualidade é crucial devido às variações nas proporções faciais entre diferentes indivíduos. Após esse processo, o sistema armazena as informações de treinamento na máquina como mostra na figura 11.

**Figura 11 - Código de treinamento**

```

1 import cv2
2 import os
3
4 import numpy as np
5
6 eigenface = cv2.face.EigenFaceRecognizer_create(threshold=10000)
7 eigenface = cv2.face.EigenFaceRecognizer_create(num_components=50, threshold=5)
8 fisherface = cv2.face.FisherFaceRecognizer_create()
9 lbph = cv2.face.LBPHFaceRecognizer_create()
10
11 usage
12 def getImagem():
13     caminhos = [os.path.join('fotos', f) for f in os.listdir('fotos')]
14     print(caminhos)
15     faces = []
16     ids = []
17     for caminho in caminhos:
18         imagemFace = cv2.imread(caminho)
19         imagemFace = cv2.cvtColor(imagemFace, cv2.COLOR_BGR2GRAY)
20         faces.append(imagemFace)
21         if "damiao" in caminho:
22             ids.append(1)
23         elif "henrique" in caminho:
24             ids.append(2)
25
26     return np.array(ids), faces
27
28 id, faces = getImagem()
29 print('treinando ....')
30 eigenface.train(faces, id)
31 eigenface.write('classificadorEigen.yml')
32
33 fisherface.train(faces, id)
34 fisherface.write('classificadorFisher.yml')
35
36 lbph.train(faces, id)
37 lbph.write('classificadorLBPH.yml')
38
39 print('treinamento realizado')

```

Fonte: Autores, 2024

No código na figura 12, a funcionalidade treinada anteriormente é testada utilizando a *webcam* para realizar o reconhecimento facial. Quando o sistema identifica um rosto, ele destaca o rosto da pessoa com um quadrado verde. A principal finalidade desse código é verificar se a máquina possui a capacidade de reconhecer rostos, oferecendo uma interface visual para visualização durante o processo de teste.

**Figura 12 - Teste de reconhecimento do rosto**

```

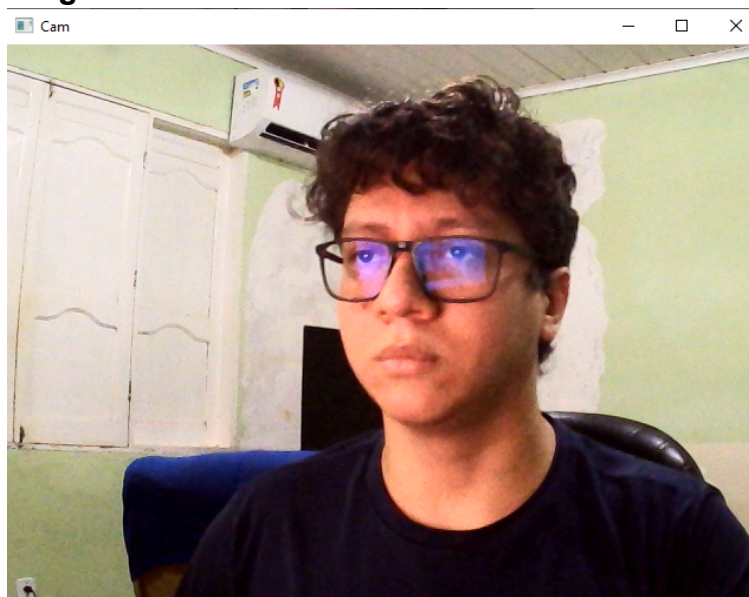
1 import cv2
2
3 camera = cv2.VideoCapture(0)
4 classificador = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
5
6 reconhecedor = cv2.face.EigenFaceRecognizer_create()
7 reconhecedor.read("classificadorEigen.yml")
8
9 while True:
10     success, img = camera.read()
11     imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
12     faces = classificador.detectMultiScale(imgGray, scaleFactor=1.5, minSize=(30, 30))
13
14     for (x, y, l, a) in faces:
15         imgFace = cv2.resize(imgGray[y:y+a, x:x+l], dsize=(220, 220))
16         cv2.rectangle(img, (x, y), (x + l, y + a), (1, 237, 0), 2)
17         id, confianca = reconhecedor.predict(imgFace)
18         print(id, confianca)
19
20     cv2.imshow( winname: 'Cam', img)
21     cv2.waitKey(1)

```

Fonte: Autores, 2024

Pode-se observar na figura 13 que o sistema não consegue reconhecer o usuário ao utilizar óculos. Isso ocorre porque, durante o treinamento, a máquina foi especificamente ensinada a reconhecer o rosto do usuário sem óculos. Essa escolha foi intencional, com o propósito de demonstrar que a máquina reconhece apenas aquilo que foi explicitamente instruído a identificar.

**Figura 13 - Teste de reconhecimento com óculos**

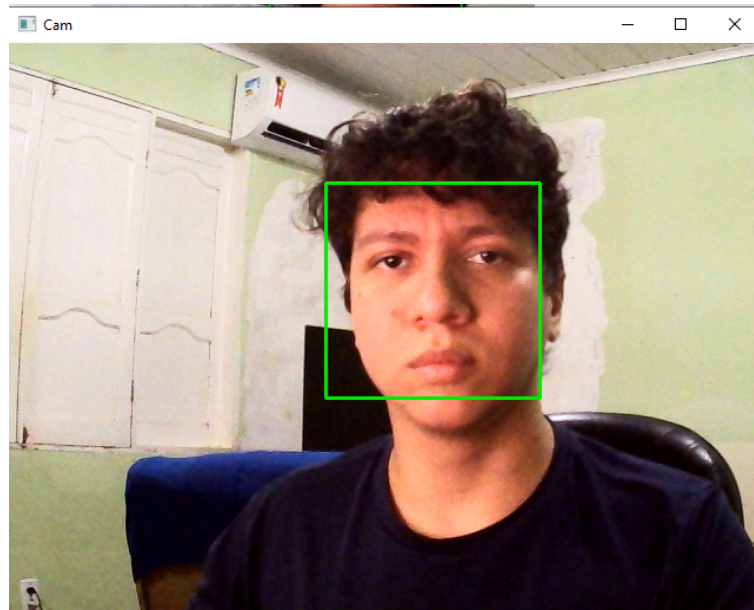


Fonte: Autor, 2024

Na figura 14, o usuário está sem óculos, e a máquina reconhece seu rosto sem dificuldades, evidenciando sua capacidade de identificar características faciais humanas. No processo de treinamento, foram incluídas diversas expressões faciais, considerando tanto nos casos de utilização de óculos e na ausência dos mesmos. Atualmente, o sistema reconhece rostos de forma eficaz, independentemente da presença de óculos como na figura 15.

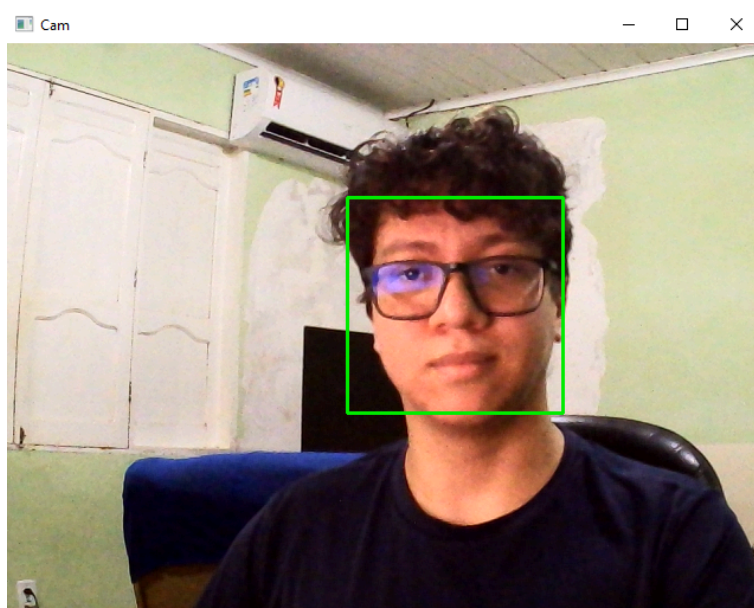
O referido procedimento deve ser realizado no usuário para assegurar que não haja dificuldades ao desbloquear sua bicicleta, mesmo diante de diversas expressões faciais. Ao realizar o reconhecimento facial, o sistema identifica não as fotos ou os nomes, mas sim o código específico criado durante o treinamento e reconhecimento do rosto que pode ser visto na figura 16.

**Figura 14** -Teste de reconhecimento sem óculos



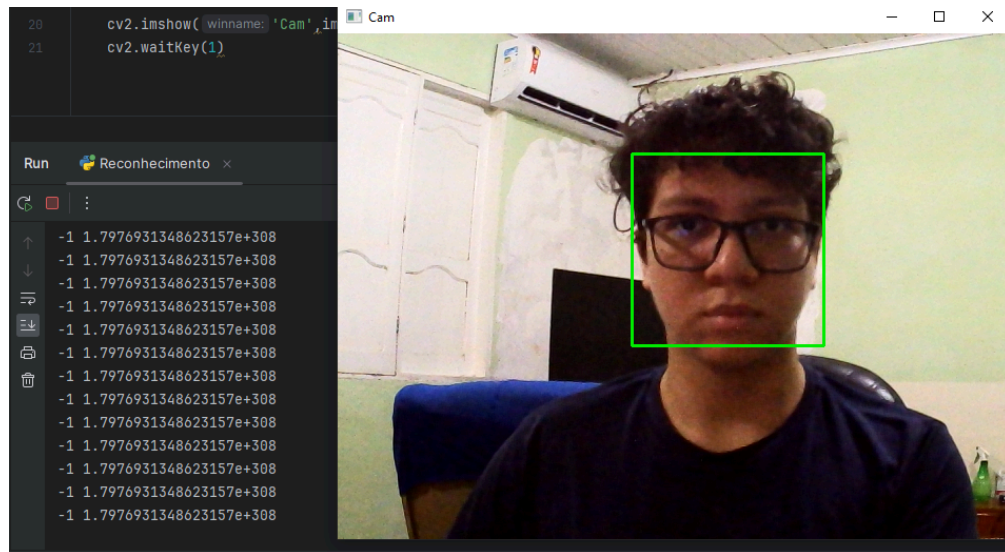
Fonte: Autor, 2024

**Figura 15** - Reconhecimento com óculos bem sucedido



Fonte: Autor, 2024

**Figura 16 - Registro do algoritmo**



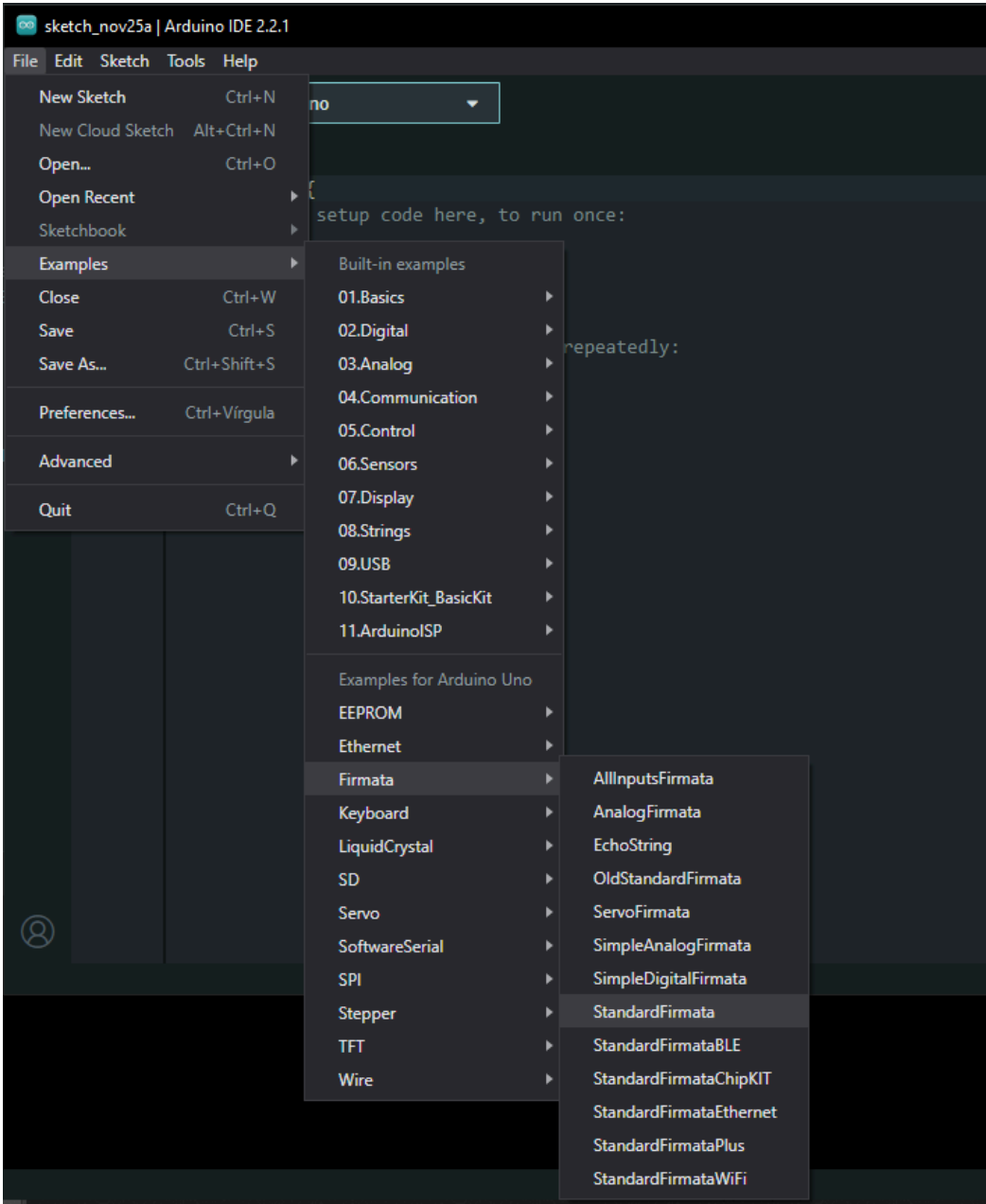
Fonte: Autor, 2024

Após a conclusão do processo de captura, treinamento e reconhecimento, a equipe direcionou seus esforços para a implementação da trava utilizando o Arduino.

Embora o Arduino utilize a linguagem de programação C++, o programa foi desenvolvido em *Python* pois a inteligência artificial foi desenvolvida em *Python* tendo um conflito em relação com Arduino. Inicialmente, elaborou-se o código em C++ para realizar testes. Posteriormente, efetuou-se a modificação do código do Arduino para *Python*, permitindo que o Arduino reconhecesse a linguagem *Python*. Isso foi possível por meio da utilização da biblioteca *pyfirmata* no *Python* e do Arduino como mostra a figura 17, estabelecendo uma conexão serial entre o computador e a placa Arduino e o código *Python*.

Para assegurar o reconhecimento do código *Python*, foi necessário instalar o *pyfirmata* no Arduino. Esse processo possibilitou a transferência do código para o Arduino, viabilizando o reconhecimento e integração eficientes entre as duas linguagens.

Figura 17 - Arduino



Fonte: Autores, 2024

**Figura 18 - Código de importação do arduino**

```

1  /*
2   Firmata is a generic protocol for communicating with microcontrollers
3   from software on a host computer. It is intended to work with
4   any host computer software package.
5
6   To download a host software package, please click on the following link
7   to open the list of Firmata client libraries in your default browser.
8
9   https://github.com/firmata/arduino#firmata-client-libraries
10
11  Copyright (C) 2006-2008 Hans-Christoph Steiner. All rights reserved.
12  Copyright (C) 2010-2011 Paul Stoffregen. All rights reserved.
13  Copyright (C) 2009 Shigeru Kobayashi. All rights reserved.
14  Copyright (C) 2009-2016 Jeff Hoefs. All rights reserved.
15
16  This library is free software; you can redistribute it and/or
17  modify it under the terms of the GNU Lesser General Public
18  License as published by the Free Software Foundation; either
19  version 2.1 of the License, or (at your option) any later version.
20
21  See file LICENSE.txt for further informations on licensing terms.
22
23  Last updated August 17th, 2017
24  */
25
26  #include <Servo.h>
27  #include <Wire.h>
28  #include <Firmata.h>
29
30  #define I2C_WRITE          B00000000
31  #define I2C_READ           B00001000
32  #define I2C_READ_CONTINUOUSLY B00010000
33  #define I2C_STOP_READING   B00011000
34  #define I2C_READ_WRITE_MODE_MASK B00011000
35  #define I2C_10BIT_ADDRESS_MODE_MASK B00100000
36  #define I2C_END_TX_MASK    B01000000
37  #define I2C_STOP_TX        1
38  #define I2C_RESTART_TX     0

```

Fonte: Autores, 2024

A primeira modificação realizada envolveu a configuração do LED para reconhecimento por meio do *Python*. Inicialmente, foram desenvolvidos dois códigos, sendo o primeiro em C++ como mostrado na figura 19, posteriormente adaptado para *Python*.

Na figura 20 mostra a diferença entre eles reside na sintaxe, onde se destaca que, no Arduino, não é necessário importar nenhuma biblioteca, apenas especificar o tipo de variável, como no caso do "int" para números inteiros. Por outro lado, no *Python*, não é necessário especificar o tipo de variável, mas a biblioteca precisa ser importada. Essa distinção entre as duas linguagens decorre do fato de que C++ é uma linguagem orientada a objetos, enquanto o *Python* não segue esse paradigma.

**Figura 19 - Led- C++**

```

1
2  int vermelho = 2;
3  int amarelo = 4;
4  int verde = 3;
5
6  void setup() {
7
8      pinMode(vermelho, OUTPUT);
9      pinMode(amarelo, OUTPUT);
10     pinMode(verde, OUTPUT);
11 }
12
13 void loop() {
14
15     digitalWrite(vermelho, HIGH);
16     digitalWrite(amarelo, HIGH);
17     digitalWrite(verde, HIGH);
18
19     delay(2000);
20
21     digitalWrite(vermelho, LOW);
22     digitalWrite(amarelo, LOW);
23     digitalWrite(verde, LOW);
24 }

```

Fonte: Autores, 2024

**Figura 20 - Led- Python**

```

1  from pyfirmata import Arduino, SERVO
2  from time import sleep
3
4  port = 'COM3'
5  board = Arduino(port)
6
7  redLed = 2
8  yeLed = 4
9  greenLed = 3
10
11  board.digital[redLed].write(1)
12  board.digital[greenLed].write(1)
13  board.digital[yeLed].write(1)
14
15  sleep(2)
16
17  board.digital[redLed].write(0)
18  board.digital[greenLed].write(0)
19  board.digital[yeLed].write(0)

```

Fonte: Autores, 2024

Neste contexto, é possível observar o código referente ao motor servo no Arduino, que necessita identificar o motor. Nesse caso, foi essencial importar a

biblioteca "servo" durante a programação em C++. No entanto, na programação em *Python*, houve a necessidade de importar tanto a biblioteca "servo" quanto a "time". O uso do tempo, neste caso, está mais direcionado ao Arduino, uma vez que ele requer a introdução de um atraso ("*delay*") para realizar determinadas ações. A implementação do tempo no Arduino é crucial para evitar a execução simultânea de todas as ações, garantindo que determinadas etapas sejam concluídas antes do início de outras. Essa abordagem visa otimizar e evitar complicações em qualquer processo mostrado na figura 21.

**Figura 21** - Código do servo motor em c++

```
1  #include <Servo.h>
2
3  #define SERVO 6
4  Servo s;
5  int pos;
6
7  void setup ()
8  {
9      s.attach(SERVO);
10     Serial.begin(9600);
11     s.write(0);
12 }
13
14 void loop()
15 {
16     for(pos = 0; pos < 90; pos++)
17     {
18         s.write(pos);
19         delay(15);
20     }
21     delay(1000);
22     for(pos = 90; pos >= 0; pos--)
23     {
24         s.write(pos);
25         delay(15);
26     }
27 }
```

Fonte: Autores, 2024

**Figura 22** - Código do servo motor em *python*

```

1  from pyfirmata import Arduino, SERVO
2  from time import sleep
3
4  port = 'COM3'
5  pin = 11
6  board = Arduino(port)
7
8  board.digital[pin].mode = SERVO
9
10 2 usages
11 def rotateServo(pin, angle):
12     board.digital[pin].write(angle)
13     sleep(0.015)
14
15 while True:
16     for x in range(0, 180):
17         rotateServo(pin, x)
18     for i in range(180, 1, -1):
19         rotateServo(pin, i)

```

Fonte: Autores, 2024

Após testar o funcionamento do Arduino e de seus componentes, como o motor servo destinado a acionar a trava, e o LED com a função de indicar se o rosto é desconhecido (LED vermelho) ou conhecido (LED verde), com o LED amarelo indicando o monitoramento de reconhecimento facial, a equipe combinou todos os códigos. O resultado foi a conclusão do projeto final de trava por reconhecimento facial.

No código a seguir na figura 23, todas as bibliotecas utilizadas nos códigos anteriores foram importadas, e as portas do Arduino foram previamente definidas para iniciar a execução do código, importando arquivos *.yml* e *.xml*, que são dados de reconhecimento do rosto e do rosto já cadastrado.

Figura 23 - Importações

```
1 import cv2
2 import statistics
3 from pyfirmata import Arduino, SERVO
4 from time import sleep
5
6 port = 'COM3'
7 pin = 10
8 board = Arduino(port)
9 redLed = 5
10 yeLed = 3
11 greenLed = 4
12
13
14 board.digital[pin].mode = SERVO
15
16 3 usages
17 def rotateServo(pin, angle):
18     board.digital[pin].write(angle)
19     sleep(0.015)
20
21 camera = cv2.VideoCapture(1)
22 classificador = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
23
24 reconhecedor = cv2.face.EigenFaceRecognizer_create()
25 reconhecedor.read("classificadorEigen.yml")
```

Fonte: Autores, 2024

Na segunda Figura 24 nesta etapa, ocorre o reconhecimento, e algumas alterações na estrutura do código foram realizadas para verificar e reconhecer o rosto que está previamente cadastrado no banco de dados do sistema.

**Figura 24 - Reconhecimento**

```

26 def capturar():
27     print('Reconhecendo Rosto')
28     idSeq= []
29     i = 0
30
31     while i <101:
32         success, img = camera.read()
33         imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
34         print(imgGray)
35         faces = classificador.detectMultiScale(imgGray, scaleFactor=1.5, minSize=(50, 50))
36
37         for (x, y, l, a) in faces:
38             imgFace = cv2.resize(imgGray[y:y+a,x:x+l], dsize= (220,220))
39             cv2.rectangle(img, (x, y), (x + l, y + a), (1, 237, 0), 2)
40             id, confianca = reconhecedor.predict(imgFace)
41             idSeq.append(id)
42
43         cv2.imshow( winname: 'Cam',img)
44         cv2.waitKey(1)
45         i =i+1
46     cv2.destroyAllWindows()
47
48     if idSeq:
49         return statistics.mode(idSeq)
50     else:
51
52     return -1
53

```

Fonte: Autores, 2024

Na terceira etapa, representada pela Figura 25, o monitoramento é destacado. Neste código, a função é monitorar os rostos que estão sendo capturados pela câmera, visando o reconhecimento facial dos usuários.

**Figura 25 - Monitoramento**

```

55 def monitor():
56
57     while True:
58         success, img = camera.read()
59         imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
60         faces = classificador.detectMultiScale(imgGray, scaleFactor=1.5, minSize=(100, 100))
61
62         if len(faces) >0:
63             print('Face encontrada!')
64             board.digital[yeLed].write(1)
65             return True
66             break
67         else:
68             return False
69

```

Fonte: Autores, 2024

No último código (Figura 26 - *Main*), encontra-se o código principal que reúne todas as instruções mencionadas anteriormente. Inicialmente, ele posiciona o servo motor na sua posição inicial (aberto) e desativa todos os LEDs. Em seguida, ele invoca os códigos de monitoramento e reconhecimento facial. Com ambos, verifica

se o rosto está no banco de dados. Se não estiver, mantém o motor inalterado (travado) e acende o LED vermelho como indicação. Caso o rosto esteja no cadastro, aciona o LED verde e destrava o dispositivo, permitindo a abertura da trava. Após alguns segundos, a trava é automaticamente fechada, finalizando assim a operação da trava inteligente para bicicleta.

**Figura 26 - Main**

```
def main():
    rotateServo(pin, angle: 100)
    while True:
        board.digital[redLed].write(0)
        board.digital[greenLed].write(0)
        board.digital[yeLed].write(0)

        print('Monitorando ...')
        num = 0
        comando = monitor()
        if comando:
            num = capturar()
            if num == 0:
                pass
            elif num == -1:
                board.digital[yeLed].write(0)
                board.digital[redLed].write(1)
                print('pessoa desconhecida!')
                sleep(3)
            elif num >= 1:
                board.digital[yeLed].write(0)
                board.digital[greenLed].write(1)
                rotateServo(pin, angle: 0)
                sleep(5)
                rotateServo(pin, angle: 100)

    main()
```

Fonte: Autores, 2024

Após a conclusão do código principal, a equipe direcionou seus esforços para a fase do protótipo, optando pelo uso de madeira devido à sua acessibilidade econômica e à facilidade de modificação. A escolha pela madeira oferece a vantagem de ser facilmente adaptada, possibilitando ajustes por meio da adição ou remoção de cortes, entre outras modificações.

Na Figura 27, foi elaborada uma estrutura de madeira com um formato semelhante a uma meia lua, projetada para acomodar a roda sem expô-la

diretamente. Essa estrutura possui dimensões de 50cm de altura e 27cm de comprimento, sendo confeccionada com compensado de 15mm para proporcionar maior durabilidade e resistência ao conjunto. Essa escolha visa atender aos requisitos necessários para o suporte eficiente da trava inteligente para bicicleta, enquanto a configuração em meia lua foi estrategicamente projetada para acomodar a roda de maneira discreta e funcional.

**Figura 27** - Estrutura de madeira compensado



Fonte: Autores, 2023

As conexões entre o protótipo e a placa Arduino, realizadas por meio da protoboard, desempenham um papel crucial na funcionalidade integrada do projeto. A breadboard, que serve como plataforma para a conexão dos componentes eletrônicos, proporciona uma maneira eficiente e organizada de interligar os diversos elementos do protótipo à placa Arduino como mostra na figura 28.

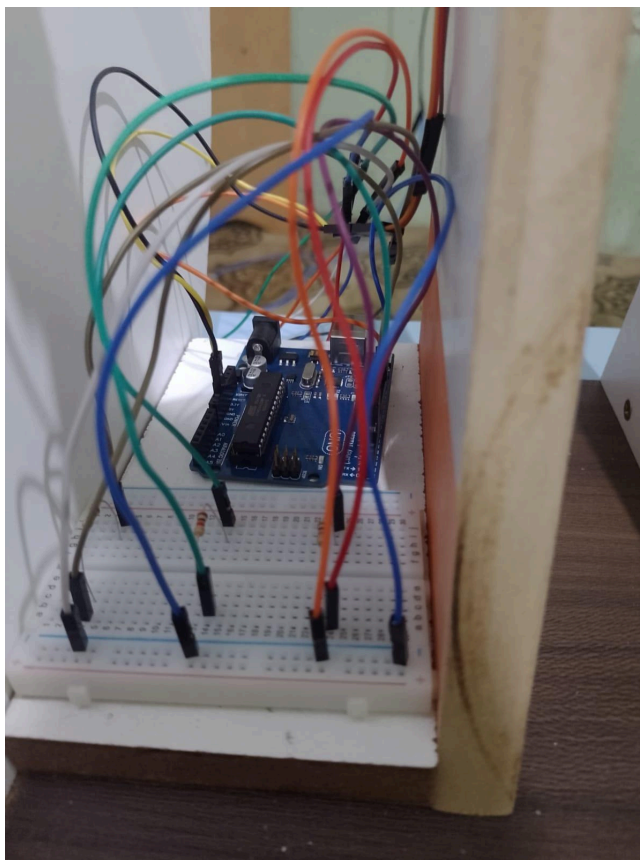
A interface entre a *breadboard* e a placa Arduino inclui a conexão estratégica de fios que vinculam sensores, atuadores e outros componentes eletrônicos ao controlador central. Por exemplo, os fios provenientes da câmera, do micro servo

motor e dos LEDs são habilmente conectados à *protoboard*, estabelecendo uma rede de comunicação eficaz.

No caso específico do micro servo motor, as conexões com a protoboard garantem a sincronização adequada entre o sistema de reconhecimento facial e o mecanismo de travamento da estrutura do protótipo. Além disso, as conexões elétricas são cuidadosamente organizadas para assegurar a integridade do sinal e a eficiência operacional.

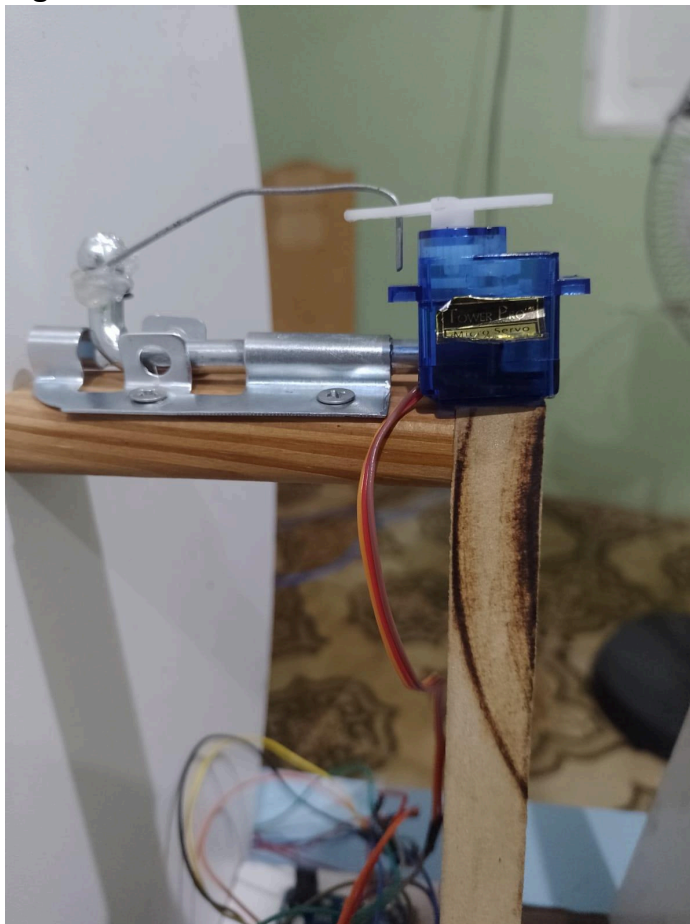
O código desenvolvido para a placa Arduino é fundamental para interpretar os dados recebidos da protoboard, processar as informações provenientes dos sensores e controlar os atuadores conforme necessário. Essa interconexão entre a placa Arduino e a protoboard constitui o elemento central do funcionamento do protótipo, permitindo uma integração eficiente entre os componentes eletrônicos e a lógica de controle do sistema.

**Figura 28** - Conexões do protótipo



Fonte: Autores, 2024

**Figura 29** - Fechadura e servo motor conectados



Fonte: Autores, 2024

Na figura 30, o sistema de trava é evidenciado, composto pelo servo motor e o ferrolho. Nesse instante, é possível observar que o mecanismo está em posição aberta, aguardando a colocação da bicicleta. No entanto, é importante ressaltar que, apesar de estar nesse estado, o sistema permanece em um estado de prontidão, pronto para efetuar o travamento assim que a bicicleta for posicionada adequadamente.

**Figura 30 - Sistema de trava aberto**



Fonte: Autores, 2024

Na figura 31, o sistema de trava é evidenciado, demonstrando sua ativação ao prender a bicicleta no protótipo. Nesse momento, fica claro que o sistema permanece aguardando, mas está pronto para ser desativado pelo usuário quando este desejar retirar sua bicicleta. Esse processo de destravamento é efetuado utilizando a câmera como meio de autenticação, adicionando uma camada de segurança ao sistema.

**Figura 31** - Sistema de trava fechado



Fonte: Autores, 2024

## **5 CONSIDERAÇÕES FINAIS**

Concluindo este estudo, no qual teve como propósito investigar a viabilidade da implementação de um sistema de estacionamento de bicicletas na instituição SENAC, integrando inteligência artificial, é possível destacar que os resultados obtidos contribuem de maneira expressiva para a compreensão e aplicação prática dessa tecnologia inovadora.

A pesquisa evidenciou a eficácia do protótipo do sistema implantado no ambiente de foco da aplicação do projeto. Esse novo sistema representa uma abordagem diferenciada em relação ao método anterior, que dependia de um caderno de protocolo para coleta de assinaturas e cartões de responsabilidade para o estacionamento. A integração do sistema proposto revelou-se uma solução promissora para o monitoramento do estacionamento, garantindo que apenas indivíduos autorizados tenham permissão para acessar o local e recuperar suas bicicletas.

Essa evolução na gestão do estacionamento não apenas simplifica o processo, mas também fortalece a segurança e o controle de acesso. A aplicação bem-sucedida da inteligência artificial nesse contexto demonstra o potencial transformador dessa tecnologia na otimização de processos organizacionais.

## REFERÊNCIAS BIBLIOGRÁFICAS

ABADI, M. et al. TensorFlow: Large-scale machine learning on heterogeneous systems. Disponível em: <<https://www.tensorflow.org/>>. Acesso em: 15 de abril de 2024.

ANSELMO, Neto. Introdução ao Numerical Python. Disponível em: <<http://www.opl.ufc.br/post/numpy/#:~:text=O%20NumPy%20é%20uma%20biblioteca,%20conhecido%20pelo%20nome%20tensor>>. Acesso em: 15 de abril de 2024.

AOZANI, Nayane Araujo. A bicicleta como alternativa sustentável para a mobilidade urbana. UNIPAMPA, 2023. Disponível em: <<https://repositorio.unipampa.edu.br/jspui/handle/riu/8915>>. Acesso em 01 de julho de 2024.

Biblioteca OpenCV. Disponível em: <<https://opencv.org/>>. Acesso em: 18 de maio de 2024.

Biblioteca OS Python. Disponível em: <<https://docs.python.org/3/library/os.html#module-os>>. Acesso em: 10 de junho de 2024.

CAPPRA, Ricardo. Na Era da Informação o Aprendizado é Data-Driven. 14 de dezembro de 2021. Disponível em: <<https://mittechreview.com.br/na-era-da-informacao-o-aprendizado-e-data-driven/>>. Acesso em: 20 de junho de 2024.

DETECÇÃO DE ANOMALIAS EM REDES DE COMPUTADORES UTILIZANDO REDES ADVERSÁRIAS GENERATIVAS. Universidade Estadual de Londrina (UEL), 2022. Disponível em: <[https://sites.uel.br/dc/wp-content/uploads/2022/09/PROJETO\\_TCC\\_VITOR\\_RUFFO.pdf](https://sites.uel.br/dc/wp-content/uploads/2022/09/PROJETO_TCC_VITOR_RUFFO.pdf)>. Acesso em: 10 de maio de 2024.

Desafios e Soluções: O avanço das Inteligências Artificiais na geração de imagens realistas. PET Sistemas de Informação, 2023. Disponível em: <<https://www.ufsm.br/pet/sistemas-de-informacao/2023/04/17/o-avanco-das-inteligencias-artificiais-na-geracao-de-imagens-realistas>>. Acesso em: 20 de fevereiro de 2024.

DOS, D.; GOMES, S. Inteligência Artificial: Conceitos e Aplicações. Revista Olhar Científico - Faculdades Associadas de Ariquemes, n. 2, [s.d.]. Disponível em: <[https://www.professores.uff.br/screspo/wp-content/uploads/sites/127/2017/09/ia\\_intro.pdf](https://www.professores.uff.br/screspo/wp-content/uploads/sites/127/2017/09/ia_intro.pdf)>. Acesso em: 21 de maio de 2024.

FENG, S. et al. Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. Nat Commun, v. 12, p. 748, 2021. Disponível em: <<https://www.nature.com/articles/s41467-021-21007-8>>. Acesso em: 21 de maio de 2024.

GOODFELLOW, I. et al. Generative adversarial nets. In Advances in neural information processing systems, 2014, pp. 2672-2680. Disponível em: <[https://proceedings.neurips.cc/paper\\_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf)>. Acesso em: 21 de junho de 2024.

GOOGLE CLOUD. O que é inteligência artificial (IA)?. Disponível em: <<https://cloud.google.com/learn/what-is-artificial-intelligence?hl=pt-br#take-the-next-step>>. Acesso em: 20 de março de 2024.

ISHIDA, Tales. Sistema de controle de acesso por reconhecimento facial utilizando sistemas embarcados. 2022. 50 f. Trabalho de Conclusão de Curso (Graduação em Engenharia e Ciências) - Universidade Estadual Paulista "Júlio de Mesquita Filho", Guaratinguetá, 2022. Disponível em: <[https://repositorio.unesp.br/bitstream/handle/11449/242780/ishida\\_thc\\_tcc\\_guara.pdf?sequence=4&isAllowed=y](https://repositorio.unesp.br/bitstream/handle/11449/242780/ishida_thc_tcc_guara.pdf?sequence=4&isAllowed=y)>. Acesso em: 14 de maio de 2024.

IZBICKI, Rafael; SANTOS, Tiago. Aprendizado de máquina: uma abordagem estatística. São Paulo: UICLAP, 2022.

KONZEN, Pedro. Python para Matemática. Disponível em: <<https://notaspedrok.com.br/notas/MiniPython/main.pdf>>. Acesso em: 15 de maio de 2024.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, 2012, pp. 1097-1105. Disponível em: <[https://proceedings.neurips.cc/paper\\_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf)>. Acesso em: 14 de maio de 2024.

LAND, Regis. Big Data e suas tecnologias em organizações orientadas por dados: uma revisão sistemática de literatura. Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2022. Disponível em: <<http://riut.utfpr.edu.br/jspui/bitstream/1/29807/1/bigdataferramentasimpactos.pdf>>. Acesso em: 14 de maio de 2024.

Larissa Gabriela. As linguagens C e C++: qual a diferença entre elas?. Disponível em: <[https://www.alura.com.br/artigos/linguagens-c-c-qual-diferenca-entre-elas?gclid=CjwKCAjwsvujBhAXEiwA\\_UXnAG7WdzilJa-y1A7d1lvS1mKI3F6zr6fre9nG4g3H66lneyVr1ntLfRoCOB8QAvD\\_BwE](https://www.alura.com.br/artigos/linguagens-c-c-qual-diferenca-entre-elas?gclid=CjwKCAjwsvujBhAXEiwA_UXnAG7WdzilJa-y1A7d1lvS1mKI3F6zr6fre9nG4g3H66lneyVr1ntLfRoCOB8QAvD_BwE)>. Acesso em: 15 de maio de 2024.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. Nature, v. 521, n. 7553, p. 436-444, 2015. Disponível em: <[https://creativecoding.soe.ucsc.edu/courses/cs523/slides/week3/DeepLearning\\_Lecun.pdf](https://creativecoding.soe.ucsc.edu/courses/cs523/slides/week3/DeepLearning_Lecun.pdf)>. Acesso em: 20 de maio de 2024.

LI, W.; MAHADEVAN, S.; VASCONCELOS, N. Anomaly detection and localization in crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 36, n. 1, p. 18-32, 2014. Disponível em: <[http://svcl.ucsd.edu/~nicolas/publications/anomaly/pami\\_anomaly.pdf](http://svcl.ucsd.edu/~nicolas/publications/anomaly/pami_anomaly.pdf)>. Acesso em: 20 de junho de 2024.

LTSI. Arduino. Disponível em: <<https://sites.usp.br/ltsi/introducao/#:~:text=O%20Arduino%20%C3%A9%20uma%20plataforma,a%20eletr%C3%B4nica%20b%C3%A1sica%20e%20programa%C3%A7%C3%A3o>>. Acesso em: 16 de maio de 2024.

Lugares onde as pessoas andam mais de bicicleta do que de carro. Disponível em: <<http://ecoparking.io/2021/01/27/nessas-cidades-do-mundo-as-pessoas-andam-mais-de-bicicleta-do-que-de-carro/>>. Acesso em: 25 de abril de 2024.

MACHINE LEARNING. Aprendizado profundo x Aprendizado de máquina em Azure Machine Learning versus Machine Learning. Microsoft, 2023. Disponível em: <<https://learn.microsoft.com/pt-br/azure/machine-learning/concept-deep-learning-vs-machine-learning?view=azureml-api-2>>. Acesso em: 11 de maio de 2024.

MELLISH, Chris. MACHINE LEARNING. 1997. Disponível em: <<https://example.com>>. Acesso em: 10 de junho de 2024.

MENDONÇA, C. M. C.; ANDRADE, A. M. V. Uso da IoT, Big Data e inteligência artificial nas capacidades dinâmicas: um estudo comparativo entre cidades do Brasil e de Portugal. Disponível em: <<http://hdl.handle.net/20.500.11959/brapci/147962>>. Acesso em: 15 de maio de 2024.

MORAES, Rodrigo. Uso de rede neural convolucional no reconhecimento de artrópodes da classe Insecta. Disponível em: <<http://riut.utfpr.edu.br/jspui/bitstream/1/29594/1/redeneuralreconhecimentootropode s.pdf>>. Acesso em: 14 de maio de 2023.

MOTA, A. Como usar uma protoboard?. Disponível em: <<https://portal.vidadesilicio.com.br/protoboard/>>. Acesso em: 17 de maio de 2024.

PASSARELLI, L. Aplicação de Visão Computacional com OpenCV. Embarcados, 2017. Disponível em: <<https://www.embarcados.com.br/aplicacao-de-visaocomputacional-com-opencv/>>. Acesso em: 7 set. 2024.

Paulo Silveira. O que é Python?. Disponível em: <<https://www.alura.com.br/artigos/o-que-e-python>>. Acesso em: 15 de maio de 2024.

PRADO, Charles. A era da inteligência artificial. Disponível em: <<https://cienciahoje.org.br/artigo/a-era-da-inteligencia-artificial/>>. Acesso em: 14 maio de 2024.

Python Software Foundation. Python 3.10.1 documentation. 2022. Disponível em: <<https://www.python.org/about/>>. Acesso em: 15 de agosto de 2024.

RUSSAKOVSKY, Olga. ImageNet Large Scale Visual Recognition Challenge. 2015. Disponível em: <<https://arxiv.org/pdf/1409.0575.pdf>>. Acesso em: 14 de maio de 2024.

RUSSEL, Stuart; NORVIG, Peter. Artificial Intelligence: A Modern Approach. 4. ed. New Jersey: Pearson, 2020.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014. Disponível em: <<https://example.com>>. Acesso em: 14 de maio de 2024.

Tower Pro micro Servo 9g. Disponível em: <<https://www.eletrogate.com/micro-servo-9g-sg90-towerpro>>. Acesso em: 25 de maio de 2024.