

**UNIVERSIDADE ESTADUAL DO SUDOESTE DA BAHIA
CAMPUS VITÓRIA DA CONQUISTA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

JOÃO PEDRO ALVES PASSOS

SISTEMA DE RECONHECIMENTO BIOMÉTRICO PALMAR A DISTÂNCIA

**VITÓRIA DA CONQUISTA
2022**

JOÃO PEDRO ALVES PASSOS

SISTEMA DE RECONHECIMENTO BIOMÉTRICO PALMAR A DISTÂNCIA

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Estadual do Sudoeste da Bahia (UESB), como requisito para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Roque Mendes Prado Trindade

VITÓRIA DA CONQUISTA

2022

JOÃO PEDRO ALVES PASSOS

SISTEMA DE RECONHECIMENTO BIOMÉTRICO PALMAR A DISTÂNCIA

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciência da Computação da Universidade Estadual do Sudoeste da Bahia (UESB), como requisito para obtenção do título de Bacharel em Ciência da Computação.

Este trabalho foi defendido e aprovado pela banca em 24/02/2023.

BANCA EXAMINADORA

Prof. Dr. Roque Mendes Prado Trindade – UESB

Orientador

Prof. Dr Hélio Lopes dos Santos – UESB

Avaliador

Prof^a. Dr Alexandra Oliveira Andrade – UESB

Avaliador

Dedico este trabalho aos meus pais,
que não pouparam esforços para que
eu pudesse concluir meus estudos.

AGRADECIMENTOS

Agradeço aos meus pais, por todo o zelo e dedicação que sempre despenderam comigo. Aos meus amigos que estiveram comigo durante todo este trajeto e me deram suporte para que pudesse chegar até aqui.

Algumas pessoas acham que foco significa dizer sim para a coisa em que você vai se focar. Mas não é nada disso. Significa não às centenas de outras boas idéias que existem. Você precisa selecionar cuidadosamente(Steve Jobs, 2008, revista Fortune).

RESUMO

O presente trabalho consiste em construir um produto capaz de fazer um reconhecimento biométrico palmar a distância. Tem como objetivo descrever as tecnologias que serão utilizadas, bem como o processo de seu desenvolvimento baseado no processamento digital de imagens, os resultados quanto a sua utilização e o seu possível impacto em sua produção. Utilizando como base de sua concepção doenças que têm como meio de propagação contato com superfícies que estão contaminadas porque alguém contaminado interagiu com essa superfície. Ao longo de cinco capítulos, será demonstrado toda a base teórica necessária para entender o objetivo, desenvolvimento, avaliação e conclusão do produto.

Palavras-chave: Reconhecimento biométrico; tecnologias; produto; processamento digital de imagens;.

ABSTRACT

The present work consists of building a product capable of doing remote palm biometric recognition. It aims to describe the technologies that will be used, as well as the process of its development based on digital image processing, the results regarding its use and its possible impact on its production. Using as a basis for its conception diseases that have as a means of propagation contact with surfaces that are contaminated because someone contaminated interacted with that surface. Over five chapters, all the theoretical basis necessary to understand the objective, development, evaluation and conclusion of the product will be demonstrated.

Keywords: Biometric recognition; technologies; product; digital image processing;.

LISTA DE ILUSTRAÇÕES

Figura 1 - Histograma de uma imagem em escala de cinza.....	18
Figura 2 - Imagem de um fluxo de automóveis.....	20
Figura 3 - Tipos de features haar.....	21
Tabela 1 - Matriz de uma imagem 8x8.....	22
Tabela 2 - Valores em 3 células de uma imagem integral.....	23
Figura 4 - Matriz de uma imagem 4x4 e sua integral.....	23
Figura 5 - Área aplicada por uma feature haar vertical.....	24
Figura 6- Valor da área no ponto A.....	24
Figura 7 - Valor da área no ponto B.....	25
Figura 8 - Valor da área no ponto C.....	25
Figura 9 - Valor da área no ponto D.....	25
Figura 10 - Imagem base Average Hashing.....	27
Figura 11 - Imagem base 8x8 escala de cinza.....	27
Figura 12 - Imagem base binarizada	28
Figura 13 - Diagrama de Contexto do sistema.....	32
Figura 14 - Diagrama de Caso de uso.....	33
Tabela 3 - Caso de Uso 1.....	34
Figura 15 - Declaração da câmera.....	35
Figura 16 - Laço que captura os frames da câmera.....	35
Figura 17 - Atribuindo o template no sistema.....	36
Figura 18 - Método de detecção da mão.....	36
Figura 19 - Método de validação da mão.....	37
Figura 20 - Classe de cadastro de mão.....	38

Figura 21 - Método de detecção da mão a ser registrada.....	39
Figura 22 - Tela principal do sistema.....	40
Figura 23 - Tela de cadastro 1.....	41
Figura 24 - Tela de cadastro 2.....	42

LISTA DE ABREVIATURAS E SIGLAS

UESB	Universidade Estadual do Sudoeste da Bahia
PDI	Processamento Digital de Imagens
DSR	Design Science Research
RGB	Red Green Blue
RGBA	Red Green Blue Alpha channel
GCI	Common Gateway Interface
GUI	Graphical User Interface
UX	User Experience

SUMÁRIO

1 Introdução	14
1.1 Objetivos	15
1.1.1 Objetivos Específicos	15
1.2 Metodologia	15
2 Revisão de literatura	17
2.1 Imagem Digital	17
2.1.1 Pixel	17
2.1.2 Resolução	18
2.2 Processamento Digital de Imagem	18
2.2.1 Histogramas e estatísticas de uma imagem	18
2.2.1.1 Histograma	19
2.2.1.2 Estatísticas	19
2.2.2 Cálculos com uma imagem	20
2.2.2.1 Modificação de intensidade	20
2.2.2.2 Operação de Threshold	20
2.3.3 Detecção de Objetos	21
2.3.4 Haar Cascade Classifier	22
2.3.4.1 Features	22
2.3.4.2 Integral image	23
2.3.4.3 Image Hash	28
2.3.4.3.5 Average Hashing	28
3 Ferramenta Desenvolvida	31
3.1 Definição de Escopo	31
3.2 Estudo das Ferramentas e Tecnologias	31
3.2.1 Python	31
3.2.1.1 Características	32
3.2.2 OpenCV	32
3.2.3 PyQt5	32
3.2.4 Raspberry Pi 3	32
3.3 Requisitos do Sistema	33
3.3.1 Diagrama de Contexto	33
3.3.2 Requisitos funcionais e não-funcionais	34
3.3.2.1 Requisitos Funcionais	34
3.3.2.2 Requisitos Não Funcionais	34
3.3.2.3 Diagramas de Caso de Uso	34
3.3.2.4 Fluxo dos Casos de Uso	35
3.4 Desenvolvimento Sistema	35
3.4.1 Reconhecer a câmera	35

3.4.2 Reconhecimento de Mão	36
3.4.3 Cadastro de mãos	39
3.4.4 Desenvolvimento Da GUI	40
4 Avaliação	44
4.1 Método de avaliação	44
5 Conclusão	45
5.1 Trabalhos Futuros	45

1 INTRODUÇÃO

A área de processamento de imagem é muito vasta, com aplicações diversas, seja em automação industrial envolvendo sensores visuais para robôs, pesquisas de recursos naturais a partir de dados capturados por satélites, entre outros. Contudo, não é comum ver aplicações em reconhecimento biométrico, de forma que seja algo democrático para o uso em atividades comuns.

Dentro desse cenário da utilização de PDI para o desenvolvimento de um reconhecedor biométrico, a ferramenta que mais vem se destacando para fins de desenvolvimento de ferramentas que tenham muita base matemática e lógica tem sido a linguagem Python.

Deste modo, portanto, tem como objetivo usar fundamentos de processamento de imagem e reconhecimento de padrões para desenvolver um sistema capaz de reconhecer uma mão a distância e devolver uma resposta caso seja reconhecida ou não. Desta forma podendo ser implementada em vários ambientes como trancas para portas, janelas ou em qualquer outro ambiente que exija algum tipo de permissão para entradas.

Ao observar que a forma mais comum de contágio de COVID-19 é feita por meio de contatos com superfícies ou objetos¹. É perceptível os limites de interação necessários para a prevenção e em pró da saúde das pessoas. Dessa forma, foi observado quais formas mais usuais seriam usadas no dia a dia para que influenciasse diretamente nas estatísticas de contaminação por contato, concluindo que objetos passem por algum tipo de toque com a mão seriam possivelmente grandes pontos focais para o vírus COVID-19.

¹ Disponível em:

<https://www.gov.br/saude/pt-br/coronavirus/como-e-transmitido#:~:text=A%20transmiss%C3%A3o%20da%20doen%C3%A7a%20pode,de%20pessoas%20sintom%C3%A1ticas%20para%20outras.> >. Acesso em: 15 nov. 2022.

1.1 OBJETIVOS

O objetivo geral deste trabalho é o desenvolvimento de um sistema capaz de reconhecer a palma de uma mão, utilizando conceitos da computação como Processamento digital de imagem, Machine Learning e Hashing.

1.1.1 Objetivos Específicos

- Reconhecer uma palma de mão em uma imagem.
- Utilizar informações biométricas para identificar a mão.
- Dar um feedback positivo caso a palma esteja cadastrada no sistema.

1.2 METODOLOGIA

Para a pesquisa e desenvolvimento desses sistemas de Reconhecimento Palmar foi utilizada a metodologia DSR que segundo Dresch et al (2015).

- Identificação do problema: é encontrado uma questão importante a ser resolvida ou a solução de um problema ou classes de problemas.
- Conscientização do problema: começar a buscar informações para entender o contexto e os requisitos do problema, podendo utilizar diversas abordagens.
- Revisão da Literatura: realizar um estudo na literatura onde se tenha estudos sobre situações parecidas ou conhecimentos já existentes.
- Identificação dos Artefatos e configuração das classes de problema: uma análise sistemática que permite desenvolver o nicho da pesquisa, sendo possível encontrar produtos que atendam às necessidades, nesse caso o novo artefato deve entregar soluções melhores dos que já existem.
- Proposição de Artefatos para resolver o problema: nela será vista a viabilidade do desenvolvimento do artefato.

- Desenvolvimento do artefato: trabalhar na implementação da abordagem que irá solucionar o problema, gerar melhoria ou trazer novas soluções.
- Avaliação do artefato: nessa fase é analisado os requisitos e os resultados para a verificação de falhas.
- Explicação das aprendizagens e conclusão: a pesquisa funcionará como referência na geração de conhecimento.
- Generalização para uma classe de problemas: o conhecimento deve ser compartilhado com o objetivo de ser aplicado em diversas situações.
- Comunicação dos resultados: a publicação do projeto, podendo ser em revistas, seminários, congressos dentre outros.

2 REVISÃO DE LITERATURA

Os primeiros registros de experiências fotográficas que se tem registro são datadas desde antes de Cristo, por meio de químicos e alquimistas. Contudo, o entendimento mais profundo a respeito do que era uma imagem e quais eram suas naturezas vieram a partir do século X veio por meio do árabe Alhaken de Basora, ao perceber as imagens que se projetavam em sua tenda ao se adentrar a luz. Os estudos a respeito da fotografia se intensificaram, até que no século XIX houve o primeiro registro de uma foto, feita pelo francês Joseph Niépce (1763-1828).[\(TODA MATERIA\)](#)

A tecnologia foi avançando junto com a popularização da fotografia, as máquinas fotográficas deixaram de ser grandes e a base de reações químicas, e começaram a ser cada vez menores e mais tecnológicas, aumentando a acessibilidade a uma câmera fotográfica de qualidade.

2.1 IMAGEM DIGITAL

A imagem digital, assim como a fotografia, é a representação de um ambiente (seja ele 3D ou não) em um plano. Contudo, diferente de uma imagem feita por uma câmera analógica que imprime em uma superfície (filme) que é extremamente sensível à luz, a imagem digital tem um processo diferente para ser criada.

Na área de computação existem várias abordagens em como uma imagem é feita. Uma imagem digital nada mais é do que uma matriz bidimensional com células que representam cores e cada célula de uma imagem é chamada de pixel.

2.1.1 Pixel

O pixel é o elemento mínimo de uma imagem feita com a técnica de rastreo. Em uma imagem bicolor um pixel pode assumir dois valores possíveis, 0 que representa o pixel desligado ou 1 que representa o pixel ligado.

Existem algumas abordagens de sistemas de cores que podem ser utilizadas em uma imagem. O sistema CMYK (cyan, magenta, yellow, black) é mais utilizado no mundo gráfico para impressão de cores com tinta, pois a combinação dessas quatro cores permite uma harmonia melhor e ressalta aspectos coloridos através da diminuição da luminosidade do fundo branco.

Ainda existe o sistema YUV que divide a representação de cores em duas partes, a luminância que vai indicar o brilho e o chroma que é a diferença entre vermelho e azul, essa abordagem traz a observação de cores como a distância em um espaço tridimensional para o ponto mais branco.

O sistema de cor mais comum para a representação de uma imagem em dispositivos digitais e a que será utilizada no sistema é a RGB ou RGBA, onde 3 matrizes, cada matriz sendo a base para uma das 3 cores primárias (o A seria o canal alfa que indica a transparência da imagem) e cada célula podendo receber um valor entre 0 e 255 indicando a intensidade da cor no ponto, são sobrepostas e o valor resultante da intersecção células de mesma coordenada irá sobressair.

2.1.2 Resolução

A qualidade de uma imagem de rastreio é dada pela quantidade de pixel que ela tem. Por exemplo, uma imagem que tem 200 pixels de largura e 50 pixels de altura tem uma qualidade menor do que uma imagem que é 480Lx480A, pois quanto mais células disponíveis para representar, mais detalhes poderão ser mostrados.

2.2 PROCESSAMENTO DIGITAL DE IMAGEM

O PDI é a área da computação que tem como meios para se atingir objetivos a capacidade de processar, analisar, editar, visualizar e armazenar imagens. É uma área ampla de desenvolvimento de soluções que passam por áreas de conhecimento da agronomia, computação, engenharia, bioquímica, entre outros. Um exemplo de utilização de PDI é a automação de veículos, onde por meio de uma

câmera a imagem é processada e filtrada, se tira as informações para que possa ser identificado obstáculos no caminho do veículo.

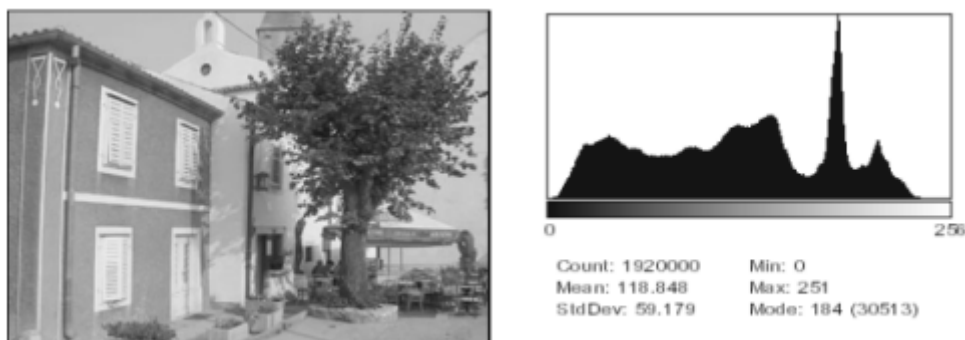
2.2.1 Histogramas e estatísticas de uma imagem

Como dito anteriormente, uma imagem pode ser representada em uma forma de matriz e muitas informações podem ser tiradas dessa representação, como gráficos para leitura de brilho, contraste, variação de cores, saturação, entre outros.

2.2.1.1 Histograma

Um histograma nada mais é do que um gráfico de distribuição de frequência, como pode ser visto na Figura 1, normalmente ele vai de uma escala que indica uma faixa de valor que cada pixel pode receber em uma imagem, de forma que esse valor pode variar entre 0 e 255 no sistema RGB por exemplo.

Figura 1 - Histograma de uma imagem em escala de cinza



Fonte: BURGER,WILLHELM; BURGE, MARK J;

Por meio do histograma é possível identificar o quão grande é o contraste de uma imagem, a partir do menor e do maior valor usado dentro da escala, quanto maior o contraste mais perceptível é as diferenças dos objetos dentro dela.

Histogramas feitos para imagens no sistema RGB são separados em três gráficos, um para cada cor representada, cada escala mostra intensidades diferentes e crescentes das respectivas cores.

2.2.1.2 Estatísticas

É possível também tirar informações estatísticas de histogramas de uma imagem, tais como: média de intensidade de cor e variância de intensidades.

2.2.2 Cálculos com uma imagem

Dados os conceitos de estatísticas e histograma de uma imagem, é possível utilizá-los para realizar manipulações da mesma. Existem softwares de edição de imagem que utilizam desses conceitos para transformar a imagem de forma mais harmoniosa dependendo do contexto.

2.2.2.1 Modificação de intensidade

Foi visto que no histograma, a representação de contraste é indicado pela distância entre o maior e menor valor mais usado da escala, para fazer manipulações como aumento ou diminuição das intensidades das cores, basta mudar proporcionalmente cada pixel da imagem caso o objetivo seja apenas mudar a intensidade da imagem.

2.2.2.2 Operação de Threshold

Fundamentalmente essa operação tem o objetivo de transformar em binário a intensidade de uma imagem, de forma que os pixels que existem nela assumam o valor de 0 ou 1, preto ou branco, ligado ou desligado.

Um parâmetro é passado para a função Threshold. Esse parâmetro existe para ser o valor limiar entre o 0 e o 1, ou seja, ele determina se o valor que está em determinado pixel irá assumir o valor 0 ou 1. Por exemplo, caso uma imagem que esteja em uma escala de cinza seja usada como base e o parâmetro usado para aplicar a função for 125, todo pixel que tiver o valor maior ou igual a 125 irá receber o valor 1 e todo pixel com valor menor que 125 receberá 0.

2.3.3 Detecção de Objetos

O principal objetivo da detecção de objetos é conseguir distinguir objetos específicos em uma imagem. Pensemos no seguinte exemplo, a prefeitura de uma determinada cidade decide fazer uma análise de fluxo de motos em determinada avenida em horários de pico.

Figura 2 - Imagem de um fluxo de automóveis



Fonte: Site doutor multas.²

Contudo, fazer uma contagem manual dessas motos não seria produtiva e nem eficiente, nesse momento a detecção de objetos se torna útil. Logo, essa prefeitura decide fazer a implementação de um sistema que consiga diferenciar carros e motos, além de conseguir fazer uma contagem da quantidade de motos que passaram pelos trechos escolhidos em um tempo determinado.

2.3.4 Machine Learning

Machine Learning é uma vertente da área de Inteligência Artificial (IA), que por sua vez tem o objetivo de por meio de algoritmos e dados, imitar a forma com que o ser humano aprende, de forma que seu aprendizado seja capaz de progredir rapidamente.

Essa é uma abordagem em que é necessário uma maior intervenção humana para o aprendizado, sendo preciso um conjunto de dados mais explícito e detalhado para que a máquina seja capaz de aprender.

O processo de aprendizado no machine learning é dividido comumente em 3 partes. Primeiramente, com base nos dados que foram passados para a máquina, uma decisão irá ser feita que inicialmente tendem a ser mais aleatórias. Logo após os resultados irão ser passados para uma função erro, que irá avaliar o quão assertivo foi a máquina. E por fim, o modelo de aprendizado da máquina será avaliado e ajustado para melhorar a taxa de acerto. Esse processo será realizado em ciclos até que se chegue ao limite de precisão do modelo.

Existem 3 métodos de machine learning, supervisionado, não supervisionado e o semi-supervisionado. O supervisionado é aquele cujo, os dados de entrada que são usados, são treinados para algoritmos que classificam dados e preveem resultados com precisão. O não supervisionado, utilizam de dados que não estão rotulados para descobrir padrões, sem a necessidade da intervenção humana, sua

² Disponível em:

<<https://doutormultas.jusbrasil.com.br/artigos/564316421/moto-no-corredor-pode-levar-multa-saiba-mais>>.

Acesso em: 15 nov. 2022.

capacidade de conseguir encontrar padrões é ideal para a utilização em análise exploratória de dados, imagem e reconhecimento de padrões, etc. O semi-supervisionado é o meio-termo entre as duas, utiliza tanto dados rotulados quanto não rotulados.

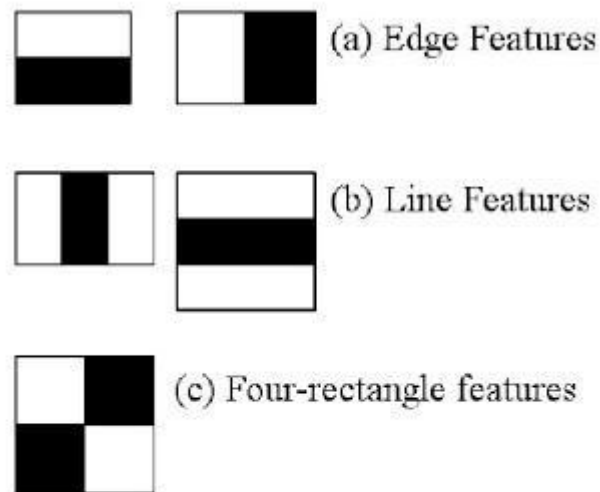
2.3.5 Haar Cascade Classifier

O Haar Cascade Classifier é um algoritmo que é usado para poder identificar objetos em uma imagem ou vídeo em tempo real. Criado por Paul Viola e Michael Jones com o objetivo de identificar faces, esse algoritmo tinha como desafio classificação binária e problema de localização. Contudo o seu funcionamento possui algumas restrições, como por exemplo, caso o treinamento do classificador seja para uma mão aberta, será difícil ou impossível detectar a mão caso esteja muito diferente do que ela foi projetada para detectar, uma mão fechada ou com um ângulo que não seja frontal não seriam reconhecidas. O contraponto dessa desvantagem é que, é possível utilizar diferentes templates de reconhecimento para serem usados em uma mesma imagem. ([WIKIPEDIA CONTRIBUTORS, 2019](#))

2.3.5.1 Features

Este algoritmo é baseado na detecção de bordas e linhas em uma imagem por meio de “features”. Cinco features são utilizadas como base para detectar os pontos de interesse da imagem.

Figura 3 - Tipos de features haar



Fonte: BURGER,WILLHELM; BURGE, MARK J;

Features do tipo A tem como objetivo detectar as bordas verticais ou horizontais. Features do tipo B com três retângulos para detectar áreas que possuam uma área mais clara em torno de uma área escura e vice-versa. E por fim a feature de 4 retângulos para detectar se existe mudanças muito grandes de intensidades em uma diagonal.

Contudo utilizar essas features pode ser bastante custoso quando usadas em imagens muito grandes. Para diminuir o custo desses cálculos, é utilizado “Integral Images”.

2.3.5.2 Integral image

Uma “Integral Image” é uma array formado pela soma de todos os pixels de uma determinada imagem base. Como visto anteriormente, uma imagem é um array bidimensional formado por pixels, e cada pixel representa uma intensidade de cor, por exemplo, uma imagem 8x8 de canal da cor cinza que vai de 0 até 255.

Tabela 1 - Matriz de uma imagem 8x8

0	6	3	10	2	9	2	23
1	4	2	12	7	2	3	15
0	2	0	4	7	2	5	23
4	4	2	14	11	5	4	18
7	22	19	6	14	11	2	3
9	2	6	7	21	30	1	1
2	3	2	4	23	12	4	5
20	12	0	0	0	0	1	13

Fonte: Autoria Própria

O cálculo para a imagem integral é feito a partir da soma do valor de todos os pixels acima e à esquerda do pixel que se está. Os valores das células de coordenadas (2,3),(4,4) e (1,6) serão:

Tabela 2 - Valores em 3 células de uma imagem integral

	13						
			68				
21							

Fonte: Autoria Própria

Com essas informações é possível encontrar os valores haar de cada feature utilizada, de forma mais performática utilizando um cálculo a partir de 4 pontos da matriz da imagem integral. Por exemplo, dada a matriz de uma imagem 4x4 cujo os valores dos pixels vão de 0 a 4 e a sua integral.

Figura 4 - Matriz de uma imagem 4x4 e sua integral

4	1	2	2	4	5	7	9
0	4	1	3	4	9	12	17
3	1	0	4	7	13	16	25
2	1	3	2	9	16	22	33

Fonte: Artigo no medium.³

Em uma feature de borda o cálculo é feito da seguinte forma, ao sobrepor a feature na imagem para calcular a área de interesse, teremos a seguinte situação.

Figura 5 - Área aplicada por uma feature haar vertical

4	5	7	9
4	9	12	17
7	13	16	25
9	16	22	33

Fonte: Autoria Própria

O cálculo do valor haar nesse ponto será determinado por valor da área mais clara menos o valor da área mais escura, e o valor de de cada uma dessas áreas é dado pelo cálculo de 4 áreas diferentes.

- A área A é dada pelo último ponto da seção de interesse, no caso da área branca seria o 16.

Figura 6- Valor da área no ponto A

³ Disponível em: <<https://towardsdatascience.com/how-much-faster-is-image-integral-90ba0008497a>> .Acesso em: 15 nov, 2022.

4	5	7	9
4	9	12	17
7	13	16	25
9	16	22	33

Fonte: Autoria Própria

- A área B é a área acima do ponto de interesse, logo o valor 5

Figura 7 - Valor da área no ponto B

4	5	7	9
4	9	12	17
7	13	16	25
9	16	22	33

Fonte: Autoria Própria

- A área C é o último ponto a esquerda da área clara

Figura 8 - Valor da área no ponto C

4	5	7	9
4	9	12	17
7	13	16	25
9	16	22	33

Fonte: Autoria Própria

- E a área D sendo a última a primeira célula da matriz

Figura 9 - Valor da área no ponto D

4	5	7	9
4	9	12	17
7	13	16	25
9	16	22	33

Fonte: Autoria Própria

O valor da área clara se é igual a:

$$\text{área clara} = A - B - C + D / \text{quantidade de pixels na área}$$

$$\text{área clara} = 16 - 5 - 9 + 4 / 3$$

$$\text{área clara} = 1.66$$

$$\text{área escura} = 22 - 7 - 16 + 5 / 3$$

$$\text{área escura} = 1.33$$

Após conseguir o valor de cada, é possível calcular o valor haar da área de interesse.

$$\text{valor haar} = \text{área clara} - \text{área escura}$$

$$\text{valor haar} = 1.66 - 1.33$$

$$\text{valor haar} = 0.33$$

Com esse valor o algoritmo consegue identificar se no quadrante da imagem existe alguma característica de interesse. O fluxo de testes são feitos divididos em 2 estágios, primeiramente são utilizados as features de bordas para verificar se existe algum objeto de interesse, caso tenha alguma resposta positiva o segundo estágio será iniciado com features linhas, ao encontrar um objeto de interesse no segundo é retornado positivo para objeto encontrado.

2.3.5.3 Image Hash

O Image Hash é um processo em uma imagem que tem como objetivo obter uma impressão digital "única" para cada imagem. Ele utiliza como base o Locality-sensitive hashing, que é um algoritmo que busca colocar inputs parecidos na mesma categoria, também é utilizado para data clustering e busca pelo vizinho mais próximo. Existem vários sub-algoritmos que derivam do Image Hash, mas o que daremos foco é o Average Hashing.

2.3.5.3.5 Average Hashing

O algoritmo do average hashing consiste em 5 passos simples. Primeiramente a imagem escolhida será reduzida a uma proporção de oito pixels de largura por oito pixels de altura, isso é feito para remover os detalhes e a frequência,

dessa forma, independente do “aspect ratio” da imagem o hash irá funcionar. Logo depois, é reduzida a escala de cores da imagem resultante para uma escala de cinza.

Figura 10 - Imagem base Average Hashing



Fonte: Artigo Content-Blockchain⁴

Figura 11 - Imagem base 8x8 escala de cinza



Fonte: Artigo Content-Blockchain⁵

⁴ Disponível em: <<https://content-blockchain.org/research/testing-different-image-hash-functions>> .Acesso em: 21 nov, 2022

⁵ Disponível em: <<https://content-blockchain.org/research/testing-different-image-hash-functions>> .Acesso em: 21 nov, 2022

Logo após a redução, o algoritmo irá ver a intensidade média dos pixels da imagem para poder encontrar o ponto que irá decidir se um pixel irá ser binarizado para 0 ou 1, e o ponto decisivo do algoritmo é justamente esse, a imagem binária será utilizada como base para poder gerar o código hash dela.

Figura 12 - Imagem base binarizada



Fonte: Artigo Content-Blockchain⁶

Para poder gerar o código hash é, cada pixel da matriz é percorrido e concatenando, gerando assim um número binário de 64 bits. Gerando dessa forma o hash: 00000000000100000000000000010000001000010100001101111111111111110.

O cálculo para descobrir a semelhança de uma imagem com outra é dado pela distância de Hamming entre elas, quanto mais próximo de zero, mais semelhante elas são.

A vantagem desse algoritmo é a simplicidade da sua implementação ele é bastante robusto contra mudanças na imagem, como mudanças de brilho, contraste, correção de gamma, escalamento, parcialmente para alterações como Corte de imagem e ajustes de cores e não funciona adequadamente em alterações da como rotação e cisalhamento.

⁶ Disponível em: <<https://content-blockchain.org/research/testing-different-image-hash-functions>> .Acesso em: 21 nov, 2022

3 FERRAMENTA DESENVOLVIDA

Esse Capítulo tem como objetivo descrever o desenvolvimento do sistema. Será aprofundada a tecnologia, técnicas e requisitos utilizados para a criação do mesmo.

3.1 DEFINIÇÃO DE ESCOPO

Para iniciar o desenvolvimento do projeto foi realizado um estudo para entender melhor o escopo do projeto e sua viabilidade. Logo, esse é o projeto de um produto que tem como objetivo realizar uma análise por meio de uma câmera para definir se existe uma mão no vídeo e se a mão apresentada está registrada no sistema

3.2 ESTUDO DAS FERRAMENTAS E TECNOLOGIAS

Nesta etapa do projeto foi definido quais ferramentas serão utilizadas para o desenvolvimento do sistema. O sistema será dividido em duas partes, um momento de registro de mãos e um momento de observação, ambas as partes serão feitas no mesmo ambiente.

3.2.1 Python

O Python é uma linguagem de programação criada em 1989 pelo matemático e programador Guido Van Rossum. É uma linguagem multiparadigma, ou seja, nela é possível programar utilizando vários paradigmas de programação, sendo elas a Programação Orientada a Objeto, Programação Imperativa, Programação Funcional e Programação Procedural. Essa linguagem se tornou bastante querida por alguns motivos interessantes.

3.2.1.1 Características

O python é uma linguagem de programação cujo a sintaxe se assemelha bastante à fala natural, visto que uma de suas características mais notáveis é a fácil leitura do código e a redução de linhas de código se comparada com outras linguagens. Atualmente a utilização do python tem se mostrado muito útil nas áreas de processamento de textos, processamento de imagens, dados científicos e CGIs para web.

3.2.2 OpenCV

OpenCV é uma biblioteca Open Source para o python que tem como objetivo trabalhar com visão computacional, foi desenvolvida pela Intel em 2000. Dentro dessa biblioteca é fornecido módulos para Processamento de Imagens e Vídeo, Estrutura de dados, Álgebra Linear, Interface Gráfica.

3.2.3 PyQt5

PyQt5 é um framework do python com objetivo de construir GUIs desenvolvida pela Riverbank Computing.

3.2.4 Raspberry Pi 3

Raspberry Pi é uma série de mini-computadores de placa única e tamanho reduzido. Começou a ser desenvolvido a partir de 2012 com o objetivo de promover o conhecimento em informática básica nas áreas de educação. Sua grande vantagem além de serem bastante pequenos é o preço reduzido em comparação com computadores mais comuns além de serem bastante "customizáveis". Atualmente, esses mini-computadores são bastantes usados como plataformas para

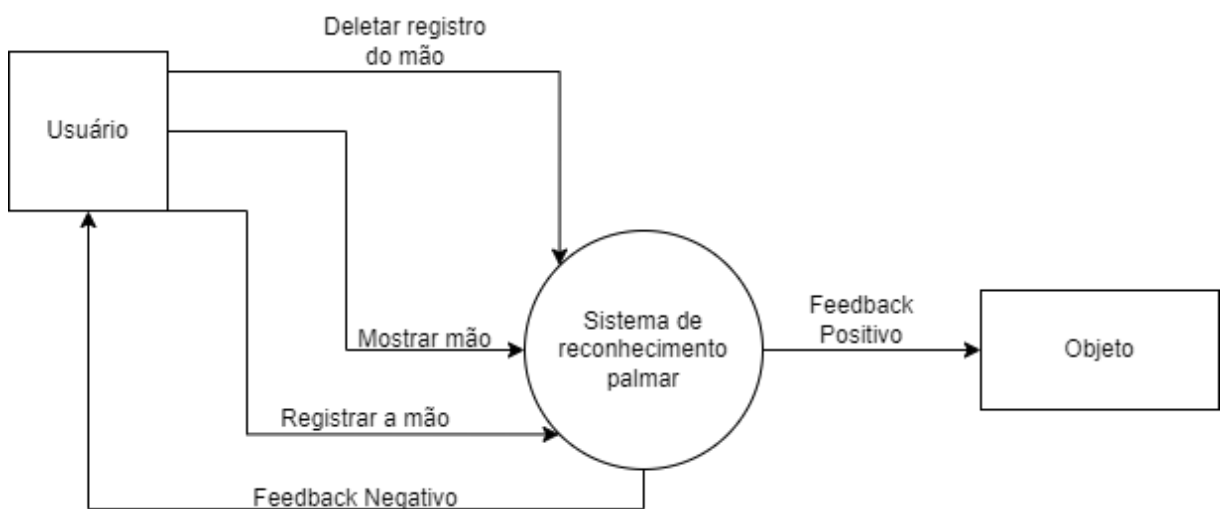
casas inteligentes e Internet das Coisas. Nesse projeto o raspberry será utilizado como a plataforma de funcionamento do sistema, utilizando como suporte a câmera.

3.3 REQUISITOS DO SISTEMA

Aqui será exposto os requisitos do sistema com relação ao diagrama de contexto, casos de uso do usuário, requisitos funcionais e requisitos não-funcionais.

3.3.1 Diagrama de Contexto

Figura 13 - Diagrama de Contexto do sistema



Fonte: Autoria Própria

3.3.2 Requisitos funcionais e não-funcionais

3.3.2.1 Requisitos Funcionais

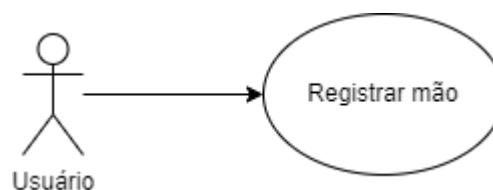
- A. O sistema deve estar monitorando em vídeo a todo momento, mas só fazer o processo de validação ao detectar uma mão.
- B. O sistema só dará feedback positivo quando a mão apresentada estiver cadastrada no sistema.
- C. O usuário deve ser capaz de fazer um cadastro de quantas mãos quiser.
- D. O usuário deve ser capaz de descadastrar qualquer mão cadastrada caso tenha a senha do sistema.
- E. O sistema deve dar um feedback após o processamento.

3.3.2.2 Requisitos Não Funcionais

- A. A resposta do reconhecimento palmar deve ser feita em até 3 segundos da detecção da mão.
- B. O cadastro do sistema deve ser simples e objetivo.

3.3.2.3 Diagramas de Caso de Uso

Figura 14 - Diagrama de Caso de uso



Fonte: Autoria Própria

3.3.2.4 Fluxo dos Casos de Uso

Tabela 3 - Caso de Uso 1

Nome do Caso de Uso: Cadastro de Mão
Autor(es)/(as): João Pedro
Atores envolvidos: Usuário
Versão: 1
Objetivo ou finalidade: Fazer o cadastro da mão para que o sistema a reconheça
Pré-condição: selecionar na tela a opção de cadastro
Fluxo principal 1- o usuário clicar no botão de cadastrar mão 2- estender a mão para a câmera até que todo o processo de cadastro seja feito
Fluxo alternativo: 1 - retornar um feedback caso demore muito a tentativa de cadastro
Pós-condição: retornar uma mensagem de sucesso para o usuário

Fonte: Autoria Própria

3.4 DESENVOLVIMENTO SISTEMA

Nesse ponto veremos o processo do desenvolvimento do sistema de reconhecimento biométrico da mão. Nessa seção será exposto o todo o processo de codificação do sistema e implementação até embarcar no raspberry pi.

3.4.1 Reconhecer a câmera

Primeiramente é preciso fazer com que o código reconheça a câmera que está acoplada no sistema. Para isso foi utilizado uma função do OpenCV chamada VideoCapture.

Figura 15 - Declaração da câmera

```
import cv2 as cv

vid = cv.VideoCapture(0)
```

Fonte: Autoria Própria

Logo após reconhecer a câmera, enquanto ela estiver em funcionamento é feito uma leitura do quadro pelo método `read`, e utilizando o método `imshow` o OpenCV uma janela no Sistema Operacional será aberta para mostrar o frame.

Figura 16 - Laço que captura os frames da câmera

```
while (vid.isOpened()):  
    ret, frame = vid.read()  
  
    if cv.waitKey(1) == ord('q'):  
        cv.destroyAllWindows()  
        break  
  
    cv.imshow("Frame", frame)
```

Fonte: Autoria Própria

3.4.2 Reconhecimento de Mão

Nesse sistema foi utilizado o Algoritmo Haar Cascade Classifier para reconhecer mãos em uma imagem, foi utilizado um detector já treinado para reconhecimento. O template foi carregado na variável `right`.

Figura 17 - Atribuindo o template no sistema

```
right = cv.CascadeClassifier("template.xml")
```

Fonte: Autoria Própria

Enquanto a câmera estiver funcionando a cada frame é executada a função `detectAndDisplay`, seu objetivo é detectar a mão em um frame e caso encontre a mão ele irá destacar e será possível visualizar em uma janela do SO. Para essa função é passado um argumento, `andimage` que tem como objetivo padronizar o tamanho do frame independente da resolução da câmera e é a junção do frame com uma imagem para delimitar a área de apresentação da mão.

Este método vai equalizar o `andframe` que está em escala de cinza, logo após ele é utilizado no template do cascade classifier para detectar a mão. O método `detectMultiScale` é uma função do `opencv` que detecta um objeto, para isso é passado uma imagem que é uma matriz do tipo `CV_8U`, o argumento `scaleFactor` que especifica o quanto a imagem vai ser reduzida ou aumentada para cada escala da imagem e `minNeighbors` determina quantos vizinhos cada retângulo candidato deve ter.

Figura 18 - Método de detecção da mão

```

119 def detectAndDisplay(andframe):
120     global threshold_type
121     global threshold_value
122     andframe = cv.equalizeHist(andframe)
123
124     righthand = right.detectMultiScale(andframe,1.2,7)
125
126     if righthand == ():
127         print("RightHand not available")
128     else:
129         for(x,y,w,h) in righthand:
130             # cv.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),3)
131             cropimage = cv.equalizeHist(andframe[y-50:y+250,x:x+200])
132
133             _, dst = cv.threshold(cropimage, threshold_value, max_binary_value, threshold_type )
134
135             # test = cv.Canny(cropimage,150,180)
136             handChecker(dst)
137             cv.imshow("hand",dst)

```

Fonte: Autoria Própria

Caso um a mão direita seja detectada a função `detectMultiScale` vai retornar um array com a localização da mão na imagem, ao receber esse array é passado para um laço que fica responsável por cortar o segmento da mão na imagem e fazer um `threshold` dela para que seja possível fazer uma checagem com as mãos cadastradas no sistema e mostra o objeto que foi detectado na tela.

Ao chegar no final da detecção ele vai para uma função chamada handChecker passando o threshold da mão e vendo se ela está cadastrada.

Figura 19 - Método de validação da mão

```
def handChecker(frame):  
    frame = Image.fromarray(frame)  
    frame_hash = imagehash.average_hash(frame)  
    for images in test_images:  
        for url in images:  
            template = cv.imread(url,0)  
            template = Image.fromarray(template)  
            template_hash = imagehash.average_hash(template)  
  
            result = frame_hash - template_hash  
  
            if(result <= 10):  
                return True  
    return False
```

Fonte: Autoria Própria

Essa função irá rodar um laço com todas as mãos cadastradas no sistema, cada usuário cadastrado tem três imagens de sua mão para que seja feita a checagem. Ao ler a imagem é utilizado a função average_hash da biblioteca ImageHash que faz uma análise da imagem com a imagem que foi passada para a função handChecker, e ao fim da análise irá retornar um número inteiro que indica o quão semelhantes essas imagens são, as três imagens feitas na hora do cadastro irão ser testadas, caso alguma delas tenha uma semelhança alta, será retornado positivo.

3.4.3 Cadastro de mãos

Foi utilizada a seguinte classe para organizar os cadastros de mãos.

Figura 20 - Classe de cadastro de mão

```
class Register():
    def __init__(self):
        self.handList = []
        self.img_counter = 0
        setTestImages()

    def reset(self):
        self.handList = []
        self.img_counter = 0

    def getHand(self, image):
        value, showimage, detected = registerHand(image)

        if(value == 1):
            self.handList.append(detected)
            self.img_counter+=1
        return cv.cvtColor(showimage, cv.COLOR_BGR2GRAY)

    def getCounter(self):
        return self.img_counter

    def createFolderUser(self, username):
        os.mkdir("./maos_reconheciveis/{}".format(username))

    def setImagesToRegister(self, username):
        for index, img in enumerate(self.handList):
            url = "./maos_reconheciveis/{}/img_{}.png".format(username, index)
            cv.imwrite(url, img)
        setTestImages()
```

Fonte: Aatoria Própria

Para o cadastro de uma nova mão a seguinte função é executada. Seu objetivo é identificar uma mão no vídeo e ao identificar irá pegar 3 imagens para que sejam usadas como identificadores do usuário, essas imagens estarão em formato

de threshold com o objetivo de tirar o máximo de informação biométrica do usuário possível.

Figura 21 - Método de detecção da mão a ser registrada

```
def registerHand(andidimage):
    grayframe = cv.cvtColor(andidimage, cv.COLOR_BGR2GRAY)
    grayframe = cv.equalizeHist(grayframe)

    righthand = detectHand(grayframe)

    if(isinstance(righthand, np.ndarray)):
        for (x,y,w,h) in righthand:
            cv.rectangle(andidimage, (x,y), (x+w,y+h), (0,255,0), 3)
            cropimage = cv.equalizeHist(grayframe[y:y+h,x:x+w])

            _, dst = cv.threshold(cropimage, 120, max_binary_value, 2 )

        return 1,andidimage,dst
    return 0,andidimage,grayframe
```

Fonte:Autoria Própria

Após conseguir as imagens, será criada uma pasta para que fique o cadastro do usuário.

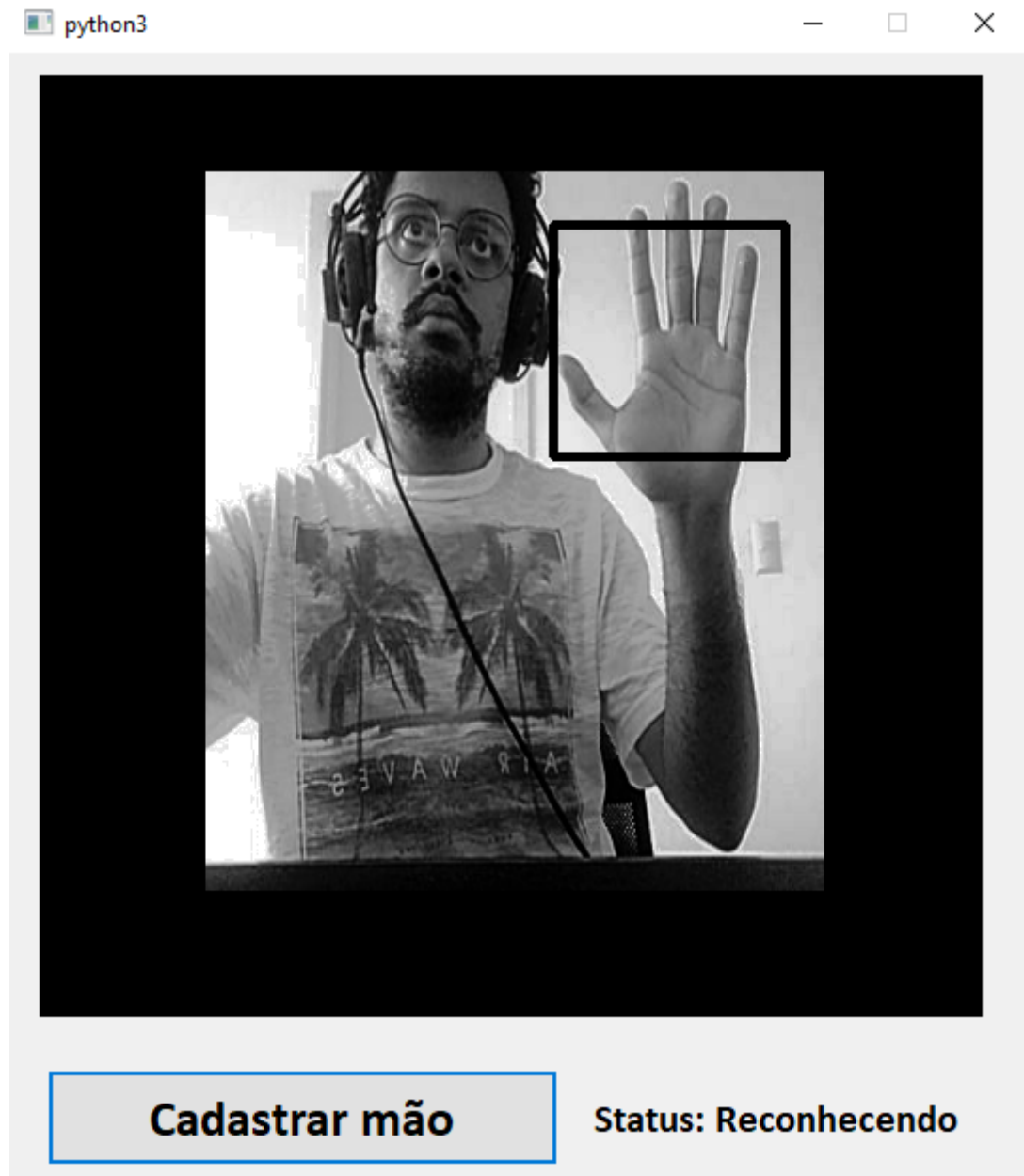
3.4.4 Desenvolvimento Da GUI

O objetivo da construção da GUI era de uma interface simples e objetiva do caminho que deveria ser seguido pelo usuário.

Logo ao abrir o sistema o usuário irá ver um retorno da câmera com a as informações de Cadastro de mão e o status do reconhecimento. Quando for apresentada a mão do usuário, caso o sistema comece o processo ao reconhecer uma mão sendo mostrada para a câmera, a mão é destacada como sinal de

feedback para o usuário. Caso a mão seja reconhecida no sistema, o reconhecedor irá dar um retorno positivo, mudando o status para sucesso.

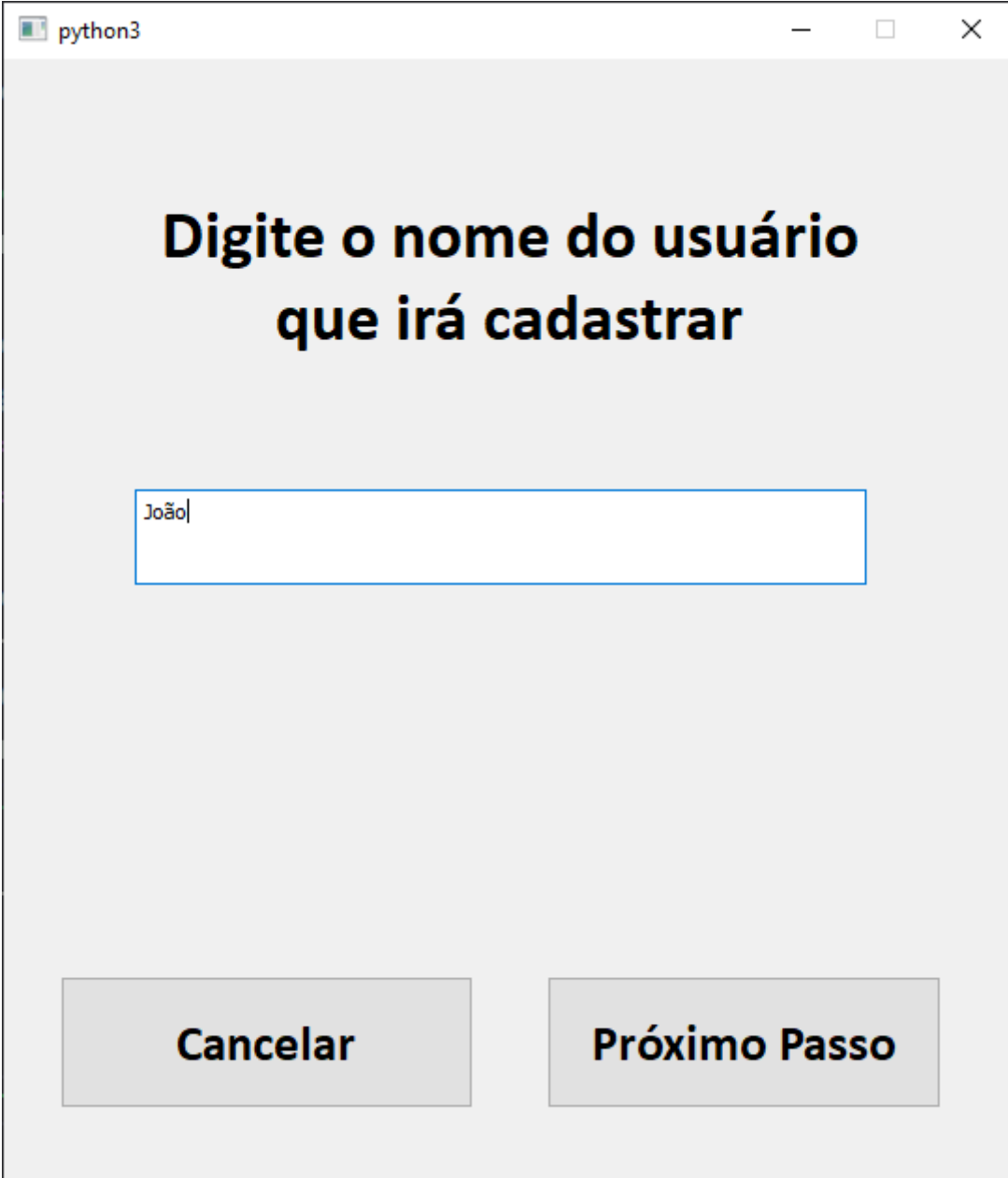
Figura 22 - Tela principal do sistema



Fonte: Autoria Própria

Ao clicar em “Cadastrar mão” o usuário será jogado para outra tela, onde pedido o nome do usuário que terá a mão cadastrada, nessa tela é possível Cancelar ou ir para o próximo passo.

Figura 23 - Tela de cadastro 1



The image shows a screenshot of a Python3 application window. The window title bar reads "python3" and includes standard minimize, maximize, and close buttons. The main content area has a light gray background. Centered on the screen is the instruction "Digite o nome do usuário que irá cadastrar" in a large, bold, black font. Below this instruction is a white text input field with a blue border, containing the text "João|". At the bottom of the window, there are two gray buttons with black text: "Cancelar" on the left and "Próximo Passo" on the right.

Fonte: Autoria Própria

Ao ir para o próximo passo, o usuário será solicitado para que estenda a mão. Ao estender ele deverá ficar dessa forma até que o cadastro finalize. O processo é

feito em 3 passos, ao estender a mão o sistema irá reconhecer a mão e tirar 3 fotos do da mão para que sejam usadas na hora da identificação.

Figura 24 - Tela de cadastro 2



Fonte: Aatoria Própria

4 AVALIAÇÃO

Neste capítulo será apresentado o método utilizado para a avaliação do sistema e seus resultados para que seja analisada a performance do sistema como um todo.

4.1 MÉTODO DE AVALIAÇÃO

Essa avaliação foi feita em 3 etapas, após o desenvolvimento do sistema para recolher informações de usabilidade.

- **Escolha de pessoas:** A fim de testes, 4 alunos do curso de Ciência da Computação da UESB foram escolhidos. Destaca-se que ambos estão em fases finais do curso.
- **Fluxo interno do sistema:** Os usuários de testes iriam fazer o fluxo de cadastro de mão e o fluxo de detecção.
- **Feedback da utilização:** Após a utilização do sistema será feita uma análise para destacar questões de funcionalidade e utilização do sistema.

Alguns pontos relevantes a respeito da utilização do sistema foram levantados, entre eles estão.

1. O algoritmo de detecção da mão no espaço não funciona muito bem com fundos mais escuros e “poluídos”.
2. A detecção de uma mão cadastrada não parece estar retornando de forma rápida e exata.
3. A interface não está polida, com erros de UX.

5 CONCLUSÃO

Esse trabalho foi desenvolvido com o objetivo de ser uma ferramenta multifuncional, podendo ser utilizada para vários fins, entre elas controle de acesso como principal foco. Projetado para reconhecer mãos a distância como medida protetiva a doenças que se contagiam por meio de contato com superfícies contaminadas.

Seu desenvolvimento foi simples devido ao forte suporte que a linguagem python oferece, disponibilizando no seu ecossistema todas as ferramentas que eram necessárias para o desenvolvimento do projeto.

Esse sistema foi projetado para que seja utilizado de forma simples e objetiva, com poucas interações diretas no sistema.

E esperado, o resultado foi bastante promissor, dado o contexto e seus objetivos. Tendo como pontos de ressalva ajustes com relação ao desempenho, alternativas na forma de iniciar o processo de análise e melhoria na interface para o usuário, pensando em métodos de desenvolvimento que priorizem a experiência do mesmo na utilização do sistema.

5.1 TRABALHOS FUTUROS

Nesta seção serão expostas ideias para possivelmente serem utilizadas como o objetivo de melhorar a experiência de utilização e aprimorar o sistema.

- Melhorar a GUI utilizando padrões de design melhores.
- Melhorar performance na detecção e registro de mãos
- Diminuir carga nos equipamentos que executam o sistema

REFERÊNCIAS

WIKIPEDIA CONTRIBUTORS. Viola–Jones object detection framework. Disponível em: <https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework> .

TOWARDS DATA SCIENCE. **How much faster is image integral?**: Compare brute force to image integral using Google Colab. 17 out. 2020. Disponível em: <<https://towardsdatascience.com/how-much-faster-is-image-integral-90ba0008497a>> . Acesso em: 15 nov. 2022.

MAUSS, B. **Of Card Tricks and Integral Images**: An Explanation of Integral Images., 22 Jan. 2021. Disponível em: <<https://benmauss.medium.com/of-card-tricks-and-integral-images-e1d4aa869216>> . Acesso em: 15 nov. 2022.

DRESCH, A.; LACERDA, D. P.; JÚNIOR, J. A. V. A. **Design science research**: método de pesquisa para avanço da ciência e tecnologia. Bookman Editora, 2015.

Python. Disponível em: <<https://pt.wikipedia.org/wiki/Python>> .

TODA MATERIA. **História da fotografia: origem e evolução**. Disponível em: <<https://www.todamateria.com.br/historia-da-fotografia/>> .

WILHELM BURGER; BURGE, M. **Digital image processing** : an algorithmic introduction using Java. New York: Springer, 2008.

Testing different image hash functions – Content Blockchain. Disponível em: <<https://content-blockchain.org/research/testing-different-image-hash-functions/>> . Acesso em: 21 nov. 2022.

Looks Like It - The Hacker Factor Blog. Disponível em: <<https://www.hackerfactor.com/blog/index.php?/archives/432-Looks-Like-It.html>> .

O que é machine learning? Disponível em:
<<https://www.ibm.com/br-pt/cloud/learn/machine-learning>>.

Locality-sensitive hashing. Disponível em:
<https://en.wikipedia.org/wiki/Locality-sensitive_hashing>.