



UNIVERSIDADE FEDERAL DO AMAZONAS - UFAM
INSTITUTO DE COMPUTAÇÃO - ICOMP
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA - PPGI

Sistema de Teleoperação Baseado em
Reconhecimento Visual de Gestos para
Robótica Assistiva

Heron Jeovalino Biváqua de Moura

MANAUS-AM

2025



PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM
INFORMÁTICA



Sistema de Teleoperação Baseado em
Reconhecimento Visual de Gestos para
Robótica Assistiva

Heron Jeovalino Biváqua de Moura

Dissertação apresentada ao Programa de Pós-Graduação em Informática, da Universidade Federal do Amazonas, como parte dos requisitos necessários à obtenção do título de Mestre em Informática, na área de Engenharia de Software e Sistemas Embarcados.

Orientador: Raimundo da Silva Barreto, D.Sc.

MANAUS-AM

2025

Ficha Catalográfica

Elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

M929s Moura, Heron Jeovalino Biváqua de
 Sistema de Teleoperação Baseado em Reconhecimento Visual de Gestos
 para Robótica / Heron Jeovalino Biváqua de Moura. - 2025.
 144 f. : il., color. ; 31 cm.

 Orientador(a): Raimundo da Silva Barreto.
 Dissertação (mestrado) - Universidade Federal do Amazonas, Programa
 de Pós-Graduação em Informática, Manaus, 2025.

 1. Robótica assistiva. 2. LIBRAS. 3. Reconhecimento de gestos. 4. Visão
 computacional. 5. Controle biomimético. I. Barreto, Raimundo da Silva. II.
 Universidade Federal do Amazonas. Programa de Pós-Graduação em
 Informática. III. Título



Ministério da Educação
Universidade Federal do Amazonas
Coordenação do Programa de Pós-Graduação em Informática

FOLHA DE APROVAÇÃO

"SISTEMA DE TELEOPERAÇÃO BASEADO EM RECONHECIMENTO VISUAL DE GESTOS PARA ROBÓTICA ASSISTIVA"

HERON JEOVALINO BIVÁQUA DE MOURA

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos Professores:

Prof. Dr. Raimundo da Silva Barreto - **Presidente**

Prof. Dr. José Reginaldo Hughes de Carvalho - **Membro Interno**

Prof. Dr. Ruiter Braga Caldas - **Membro Externo**

Manaus, 12 de setembro de 2025.



Documento assinado eletronicamente por **Raimundo da Silva Barreto, Professor do Magistério Superior**, em 12/09/2025, às 15:36, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **José Reginaldo Hughes Carvalho, Professor do Magistério Superior**, em 16/09/2025, às 11:37, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Ruiter Braga Caldas, Coordenador de Curso**, em 16/09/2025, às 23:56, conforme horário oficial de Manaus, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufam.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **2764017** e o código CRC **1E84BC65**.

Avenida General Rodrigo Octávio, 6200 - Bairro Coroados I Campus Universitário Senador Arthur Virgílio Filho, Setor Norte - Telefone: (92) 3305-1181 / Ramal 1193
CEP 69080-900, Manaus/AM, coordenadorppgi@icomp.ufam.edu.br

Referência: Processo nº 23105.037961/2025-63

SEI nº 2764017

*O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES-PROEX) - Código de Financiamento 001, sendo parcialmente financiado pela Fundação de Amparo à Pesquisa do Estado do Amazonas – FAPEAM – por meio do projeto **POSGRAD 24-25**.*

Agradecimentos

A minha esposa Marta, por seu apoio e amor sustentador em todos os projetos e sonhos que abracei.

A minha família, André, Amy, Gabriel, Jessica, Isabela e Analice por me inspirarem a não desistir.

A Jackson e Amorim, colegas de profissão que colaboraram decisivamente nos processos de confecção e montagem de diversas partes da estrutura dos protótipos de mãos e na soldagem e preparação dos seus circuitos.

Ao sr. Moacir Freitas pelo compartilhamento de décadas de experiência de eletrônica e seu apoio prático nas montagens dos circuitos eletrônicos envolvidos nos experimentos deste trabalho

Ao meu professor e orientador Raimundo Barreto, por seu benéfico e surpreendente dom em saber compreender e "ler" pessoas, tais como eu e meus colegas, sempre apresentando novas ideias, caminhos e sugestões para poder ajudá-las e estimulá-las a seguir em frente. Sua orientação e apoio sempre me impulsionaram, me conduziram e me fizeram chegar até aqui.

"Form follows function"

Louis Sullivan, MIT

*"Robots seem like bridges between
technology and human life"*

B. J. Gupta

Resumo

Esta dissertação investiga uma arquitetura de robótica assistiva voltada à reprodução biomimética de gestos manuais por meio de uma mão robótica antropomórfica de cinco dedos. A proposta configura-se como uma prova de conceito que integra captura visual por câmera, processamento e reconhecimento de gestos, mapeamento cinemático e controle de atuadores, de modo a converter sinais visuais em movimentos físicos correspondentes. O foco central recai sobre o desenho e a validação de uma cadeia de processamento “visão-ação” capaz de realizar gestos estáticos de forma fiel à morfologia humana, contribuindo para o avanço de soluções generalizáveis em interação humano-robô e acessibilidade tecnológica. Para demonstrar a viabilidade da arquitetura, conduz-se um estudo de caso com as cinco vogais da Língua Brasileira de Sinais (Libras)—A, E, I, O e U—utilizadas aqui como conjunto representativo de configurações manuais. Nessa aplicação exemplar, a mão robótica executa os gestos identificados a partir das imagens, evidenciando o potencial do sistema para traduzir entradas visuais em saídas motoras de maneira coerente e reproduzível. Embora a ênfase não esteja na Libras como objeto principal, os resultados do estudo de caso ilustram como a abordagem pode apoiar estratégias de acessibilidade comunicacional para pessoas surdas e, de forma mais ampla, servir de base para outras instâncias de mediação gesto-ação em contextos educacionais, assistivos e de teleoperação. A arquitetura é composta por dois servidores web conectados via rede sem fio. O primeiro realiza a detecção e classificação dos gestos utilizando bibliotecas de visão computacional e aprendizado profundo. O segundo converte os sinais interpretados em comandos motores enviados a um conjunto de servomotores embarcados. O sistema é projetado para funcionar sem a necessidade de dispositivos vestíveis ou sensores mecânicos. A metodologia proposta abrange quatro etapas principais: (1) rastreamento e reconhecimento visual das vogais em LIBRAS; (2) integração dos servidores de interpretação e controle; (3) comunicação eficiente com a camada IoT; e (4) execução precisa dos gestos por atuadores robóticos. Os resultados esperados incluem a demonstração da viabilidade funcional do sistema, sua aplicabilidade como ferramenta de comunicação assistiva, e a redução da complexidade das interfaces físicas voltadas à tradução automatizada de sinais.

Palavras-chave: Robótica assistiva; LIBRAS; reconhecimento de gestos; visão computacional; controle biomimético; Internet das Coisas; acessibilidade.

Abstract

This dissertation investigates an assistive robotics architecture designed for the biomimetic reproduction of hand gestures through an anthropomorphic five-finger robotic hand. The proposal is conceived as a proof of concept that integrates visual capture via camera, gesture processing and recognition, kinematic mapping, and actuator control, in order to convert visual signals into corresponding physical movements. The central focus lies in the design and validation of a “vision–action” processing chain capable of performing static gestures faithfully to human morphology, thereby contributing to the advancement of generalizable solutions in human–robot interaction and technological accessibility. To demonstrate the feasibility of the architecture, a case study is conducted with the five vowels of the Brazilian Sign Language (Libras)—A, E, I, O, and U—used here as a representative set of manual configurations. In this exemplary application, the robotic hand executes the gestures identified from the images, highlighting the system’s potential to translate visual inputs into motor outputs in a coherent and reproducible manner. Although Libras is not the primary focus of the research, the case study results illustrate how the approach may support communication accessibility strategies for deaf individuals and, more broadly, serve as a basis for other instances of gesture–action mediation in educational, assistive, and teleoperation contexts. The architecture is composed of two web servers connected via wireless network. The first performs gesture detection and classification using computer vision and deep learning libraries. The second converts the interpreted signals into motor commands sent to an embedded set of servomotors. The system is designed to operate without the need for wearable devices or mechanical sensors. The proposed methodology encompasses four main stages: (1) visual tracking and recognition of vowels in Libras; (2) integration of interpretation and control servers; (3) efficient communication with the IoT layer; and (4) precise execution of gestures by robotic actuators. The expected outcomes include demonstrating the functional feasibility of the system, its applicability as an assistive communication tool, and the reduction of physical interface complexity for automated sign translation.

Keywords: Assistive robotics; Libras; gesture recognition; computer vision; biomimetic control; Internet of Things; accessibility.

Lista de Figuras

1.1	<i>Robô-cachorro policial teleoperado pela SWAT de Los Angeles.</i>	2
1.2	<i>Diagrama de Componentes UML do sistema proposto</i>	5
1.3	<i>Diagrama de Sequência UML</i>	6
2.1	<i>Diagrama SysML de State Machine modelando um sistema de controle de um automóvel</i>	18
2.2	<i>Taxonomia dos Diagramas SysML</i>	21
4.1	<i>Captura de dados por meio do registro de movimentos gestuais</i>	45
4.2	<i>Processamento do sinal de vídeo por algoritmo classificador de movimento gestual</i>	46
4.3	<i>Transmissão de dados processados pelo algoritmo de reconhecimento gestual</i>	47
4.4	<i>Recepção de dados de movimento por meio do sistema de captura</i>	48
4.5	<i>Execução do movimento mecânico da mão robótica comandado pelo servomotor</i>	49
4.6	<i>Sinal PWM ordenando ao mecanismo IoT movimento de rotação suave de 15°.</i>	51
4.7	<i>Diagrama de Atividade: Visão Ampla</i>	53
4.8	<i>Diagrama de Atividade: Etapa 1 - Captura da Imagem</i>	54
4.9	<i>Diagrama de Atividade: Etapa 2 - Controle Mecânico</i>	55
4.10	<i>Diagrama de Definição de Blocos: Visão de Alta Abstração</i>	57
4.11	<i>Diagrama de Definição de Blocos: Visão Detalhada</i>	58
5.1	<i>Etapas do desenvolvimento do sistema de reconhecimento gestual.</i>	76
5.2	<i>Fluxo de processamento e integração entre bibliotecas na aplicação de reconhecimento gestual.</i>	79
5.3	<i>Módulo IoT ESP32_WROOM</i>	82
5.4	<i>Ilustração de diagrama apresentando um módulo IoT ESP32 e servo motor</i>	83
5.5	<i>Pinagem de Módulo IoT ESP32-WROOM 30 pinos</i>	84
5.6	<i>Classificação de gestos com Fingerpose</i>	87
5.7	<i>Modelo de landmarks do MediaPipe</i>	88
5.8	<i>Testes de captura de imagens e landmarks, visando movimentos reflexos em plataforma experimental</i>	89

6.1	<i>Os cinco sinais de LIBRAS referentes às vogais do alfabeto e escopo deste estudo.</i>	93
6.2	<i>Testes com Servo Motores conectados a potenciômetros e a Servo Shield V5 Arduino</i>	94
6.3	<i>Diagrama apresentando fluxo e sucessão de experimentos planejados</i>	95
6.4	<i>Experimentos de FASE 1: confecção de dispositivos primitivos para simulação biomimética</i>	96
6.5	<i>FASE 2: testes com protótipo de mão robótica acoplado a console de aeromodelo.</i>	97
6.6	<i>Experimentos de flexionamentos e retratibilidade na FASE 2.</i>	98
6.7	<i>Placas de circuito Buck Converter DC-DC Step Down XL4015 para filtragem de ruídos.</i>	99
6.8	<i>Diagrama elétrico do circuito de alimentação dos servo motores e do Módulo IoT ESP32</i>	100
6.9	<i>Soldagem e acoplamentos de componentes para o Protótipo 3.</i>	102
6.10	<i>Protótipo 3: Mão confeccionada em DIPROM, anexada a uma estrutura MDF</i>	102
6.11	<i>Subsistemas integrantes do Protótipo 3 (FASE 3) de Mão Robótica.</i>	103
6.12	<i>Power adapters DC para INPUT de alimentação</i>	103
6.13	<i>Eixo rotacional usinado responsável pelo controle de giro do pulso da mão robô.</i>	104
6.14	<i>Ilustrações de flexão de dedos e pulso em resposta a pulsos PWM.</i>	108
7.1	<i>URL e QR-Code para video de apresentação do trabalho</i>	114

Lista de Tabelas

2.1	Áreas de Pesquisa em Visão Computacional e suas Aplicações	24
2.2	Comparação entre Métodos de Visão Robótica	26
2.3	Classificações e Descrição do Controle de Manipulação	27
3.1	Matriz comparativa dos trabalhos correlatos analisados	40
4.1	<i>Matriz comparativa entre algoritmos de aprendizado de máquina para reconhecimento de gestos manuais. Legenda: (*****) Excelente; (****) Muito bom; (***) Moderada; (**) Baixa; (*) Muito baixa; (x) Não aplicável.</i>	64
4.2	<i>Matriz comparativa entre as abordagens de tratamento de imagem para reconhecimento gestual em LIBRAS.</i>	67
4.3	<i>Matriz comparativa considerando o quesito do uso ou não de algoritmos classificadores associados aos cenários promovidos pelas abordagens estudadas no tratamento de imagem para reconhecimento gestual.</i>	68
4.4	<i>Matriz comparativa entre algoritmos de aprendizado de máquina quanto à performance e características técnicas aplicadas ao reconhecimento de gestos manuais. Legenda indicando graus de performance: (*****) – Excelente; (****) – Muito bom; (***) - Moderada; (**) - Baixa; (*) - Muito baixa; (x) - Não aplicável ou péssima performance.</i>	68
6.1	Parâmetros de configuração experimental	105
6.2	Métricas de desempenho do sistema	106
6.3	Matriz de confusão (n=50 testes)	107
6.4	Matriz de movimentação sistêmica angular dos 6 servomotores para comandar ações tanto sobre os 5 dedos como sobre giro da base do punho no eixo longitudinal.	108
6.5	Latência por estágio (ms)	109
6.6	Análise SWOT do sistema	109

Lista de Abreviaturas e Siglas

2D	Bidimensional
3D	Tridimensional
AC	Alternating Current
AI	Artificial Intelligence
CNN	Convolutional Neural Networks
CPU	Central Processing Unit
DC	Direct Current
DBD	Block Definition Diagram
ESP32	Espressif Systems Microcontroller
GB	Gigabyte
GND	Ground (Reference Potential)
GPU	Graphics Processing Unit
GRT	Gesture Recognition Toolkit
HGR	Hand Gesture Recognition
HRI-C	Human Robot Interaction-Colaboration
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
Hz	Hertz
IBD	Internal Block Diagram
INCOSE	International Council on Systems Engineering
IoT	Internet of Things
JS	JavaScript
JSON	JavaScript Object Notation
LCD	Liquid Crystal Display
LSTM	Long Short-Term Memory
MB	Megabyte
ME	Movimento de Epêntese

ML	Machine Learning
OMG	Object Management Group
PWM	Pulse Width Modulation
QUGV	Quadruped Unmanned Ground Vehicle
RGB	Red, Green, Blue
RGB-D	Red, Green, Blue, Depth
RNN	Recurrent Neural Networks
SoC	System on a Chip
SPD	Symetric and Positive Defined Neural Network
SVM	Support Vector Machine
SWAT	Special Weapons and Tactics
SysML	Systems Modeling Language
TTS	Text to Speech
UML	Unified Modeling Language
URI	Uniform Resource Identifier
USB	Universal Serial Bus
V8	Google's JavaScript Engine
Vcc	Voltage at the Common Collector
VR	Virtual Reality
WiFi	Wireless Fidelity
YOLO	You Only Look Once

Sumário

1	Introdução	1
1.1	Contexto	1
1.2	Definição do Problema	3
1.3	Motivação	3
1.4	Objetivos	3
1.4.1	Objetivo Geral	3
1.4.2	Objetivos Específicos	3
1.5	Visão Geral do Método Proposto	5
1.6	Organização da Proposta	5
2	Fundamentação Teórica	7
2.1	Visão Computacional: Conceitos e Aplicações	7
2.1.1	Fundamentos da Robótica	7
2.1.2	Visão Robótica	8
2.2	Reconhecimento de Gestos	11
2.2.1	Definição e Classificação de Gestos	11
2.2.2	Reconhecimento de Gestos Manuais	11
2.2.3	Aplicações em Robótica Manipuladora	13
2.3	Interação Humano-Máquina através de Visão Computacional	15
2.3.1	Tecnologias de Sensoriamento Visual	15
2.3.2	Técnicas de Processamento Visual	15
2.3.3	Análise Visual de Comportamento Humano	16
2.4	A Linguagem SysML para Modelagem de Sistemas	17
2.4.1	Definição e Contexto	17
2.4.2	Vantagens e Aplicações da SysML	17
2.4.3	Diferenças entre SysML e UML	18
2.4.4	Taxonomia dos Diagramas SysML	20

2.4.5	Funcionalidades Específicas dos Diagramas	22
2.5	Mapeamento Sistemático da Literatura	23
2.5.1	Análise Taxonômica de Metodologias	23
2.5.2	Análise Comparativa de Métodos em Visão Robótica	25
2.5.3	Controle de Robôs Manipuladores	27
2.5.4	Reconhecimento de Gestos Baseado em Esqueletos	29
2.6	Resumo do Capítulo	30
3	Trabalhos Correlatos	31
3.1	Reconhecimento de Gestos através de Dispositivos de Automação Residencial	31
3.1.1	Fundamentação Teórica e Metodológica	31
3.1.2	Implementação e Características Técnicas	32
3.1.3	Metodologia e Resultados	32
3.1.4	Análise do Estado da Arte Relacionado	32
3.1.5	Contribuições e Limitações	32
3.2	Dispositivo Virtual para Aprendizado de Habilidades com Assistência Robótica	33
3.2.1	Arquitetura do Sistema Proposto	33
3.2.2	Metodologia de Aprendizado e Transferência	33
3.2.3	Vantagens e Aplicações	34
3.2.4	Características Técnicas Relevantes	34
3.2.5	Relevância para a Pesquisa Proposta	34
3.3	Braço Robótico Controlado por Reconhecimento de Gestos Manuais	35
3.3.1	Fundamentação Teórica	35
3.3.2	Arquitetura Neural e Transferência de Aprendizado	35
3.3.3	Sistema de Classificação e Controle	35
3.3.4	Vantagens Operacionais	36
3.3.5	Aplicações e Resultados	36
3.3.6	Contribuições Metodológicas	36
3.4	Sistema Interativo de Robô Controlado Remotamente por Gestos Manuais para Pessoas com Deficiências	37
3.4.1	Arquitetura do Sistema IoT	37
3.4.2	Fluxo Operacional	37
3.4.3	Capacidades Robóticas Implementadas	37
3.4.4	Métricas de Performance	38
3.4.5	Inovações Tecnológicas	38
3.4.6	Aplicações e Impacto Social	39
3.4.7	Relevância para a Pesquisa Proposta	39

3.5	Análise Comparativa e Posicionamento da Pesquisa	40
3.5.1	Matriz Comparativa de Abordagens	40
3.5.2	Lacunas Identificadas	40
3.5.3	Contribuições Distintivas da Pesquisa Proposta	40
3.5.4	Posicionamento no Estado da Arte	41
3.6	Resumo do Capítulo	41
4	Método Proposto	43
4.1	Arquitetura do Sistema	43
4.2	Metodologia do Sistema Proposto	44
4.3	Modelagem SysML do Sistema Proposto	52
4.3.1	Diagrama de Atividade	52
4.3.2	Diagrama de Definição de Blocos (<i>Block Definition Diagram</i> - BDD)	56
4.3.3	Diagrama de Definição de Blocos - Visão de Alta Abstração	57
4.3.4	Diagrama de Blocos Internos	59
4.4	Estratégias de Captura e Processamento de Imagem	59
4.4.1	Abordagem 1: Medição de Distância entre Landmarks	60
4.4.2	Abordagem 2: Mapeamento de Flexão de Dedos	62
4.4.3	Abordagem 3: MediaPipe e Scikit-Learn com Classificador Random Forest	63
4.5	Resumo do Capítulo	69
5	Ferramentas Implementacionais	70
5.1	Configuração do Ambiente de Desenvolvimento	70
5.2	Implementação do Método Proposto	71
5.3	Ferramentas de Visão Computacional e Frameworks Utilizados	72
5.3.1	Etapas da Construção do Classificador Random Forest	72
5.4	Engenharia de Software para o Reconhecimento Gestual	73
5.4.1	Objetivos e Funcionalidades	73
5.4.2	Frameworks e Tecnologias Empregadas	73
5.4.3	Entradas e Saídas do Sistema	73
5.4.4	Arquitetura e Pipeline do Sistema	74
5.4.5	Pacotes Python e dependências utilizadas na aplicação de reconhecimento gestual	76
5.4.6	Bibliotecas e Classificador utilizados	80
5.4.7	Aplicação e uso do conceito de “Vision Task”	80
5.5	Ferramentas de Backend para Implementação da Camada HTTP WebServer	81
5.5.1	Ambiente Server-Side com Node.js e WebServer HTTP	81
5.6	Dispositivos IoT e Engenharia de Sistemas Embarcados utilizada no método	82

5.6.1	Módulo ESP32 WROOM	82
5.6.2	Controle de Servomotores via ESP32	84
5.6.3	WebServer Embarcado no ESP32	85
5.7	Implementação da Captura de Imagem	86
5.7.1	Abordagem 1: Distância entre Landmarks	86
5.7.2	Abordagem 2: Classificação de Flexão (Implementada via biblioteca Fingerpose.js)	86
5.7.3	Integração com Machine Learning	88
5.8	Resumo do Capítulo	90
6	Avaliação Experimental	92
6.1	Definição do Escopo da Prova de Conceito	92
6.2	Evolução de Experimentos com elementos anatômicos	94
6.2.1	Fases de testes para obtenção de modelo de protótipo anatômico satisfatório	94
6.2.2	Desenvolvimento de primeiros protótipos mecânicos de anatomia manual (FASE 1): mãos e dedos	96
6.2.3	Concepção de Protótipo 2 (FASE 2) de Mão Robótica confeccionada com madeira e com servos controlados por console de aeromodelo.	97
6.2.4	Circuito padrão de suprimento de alimentação dos Servo Motores e do módulo IoT ESP32	98
6.2.5	Concepção de Protótipo 3 (FASE 3) de Mão Robótica confeccionada com material DI- PROM e desenvolvimento de rede distribuída de circuitos de alimentação de servos)	100
6.3	Resultados Obtidos	105
6.3.1	Metodologia de Testes	105
6.3.2	Configuração Experimental	105
6.3.3	Análise Quantitativa	105
6.3.4	Análise Temporal Detalhada	108
6.3.5	Limitações e Melhorias	109
6.4	Resumo do Capítulo	110
7	Conclusão e Trabalhos Futuros	111
7.1	Limitações	111
7.2	Contribuições alcançadas	112
7.3	Trabalhos futuros	113
	Apêndices	114
	Glossário	115
	Referências Bibliográficas	120

Capítulo 1

Introdução

No domínio da teleoperação de sistemas robóticos com suporte de plataformas embarcadas, o controle remoto por meio de gestos manuais, capturados especialmente por câmeras de vídeo, configura-se como uma abordagem emergente de elevada relevância. Tal paradigma de interação homem-máquina viabiliza a execução de tarefas remotas por meio de interfaces não intrusivas, oferecendo uma solução eficiente e adaptável em ambientes onde a presença física do operador é restrita, inviável ou apresenta riscos operacionais significativos. (Khan *et al.* 2023).

As aplicações da telerrobótica vêm se expandindo de forma significativa. No setor industrial, robôs teleoperados reduzem a exposição humana em ambientes de produção (Barker e Jewitt 2022). Na área médica, viabilizam procedimentos cirúrgicos complexos realizados remotamente (Huang *et al.* 2011). Na exploração espacial, permitem o controle de sondas e rovers em outros planetas (Fong *et al.* 2013). Robôs humanoides também vêm sendo projetados para interações mais naturais com humanos (Cascone *et al.* 2021). Já em ambientes subaquáticos ou de risco extremo, são empregados para exploração, resgate e inspeção de estruturas (Lehman *et al.* 2008), (Yuh *et al.* 2011).

Independentemente do contexto, uma característica comum entre esses sistemas é sua capacidade de realizar tarefas em ambientes perigosos ou inacessíveis, ampliando as possibilidades humanas (Bekey 2005).

Entre as interfaces de controle, o reconhecimento de gestos se destaca como uma solução prática e intuitiva. Essa técnica inclui quatro componentes fundamentais: sensores, identificação, rastreamento e classificação de gestos (Liu e Wang 2018).

1.1 Contexto

A teleoperação refere-se ao controle remoto de robôs, permitindo que operadores humanos desempenhem tarefas em locais perigosos ou de difícil acesso. Em novembro de 2023, por exemplo, a SWAT de Los Angeles utilizou um cão-robô teleoperado por um membro da equipe de policiais (imagem do dispositivo em

ação na Figura 1.1) para abordar com sucesso um suspeito armado em um ônibus, demonstrando a eficácia desses sistemas em situações reais e úteis em contextos e motivos que justifiquem ou exijam simultaneamente distanciamento e continuidade de abordagem com perícia humana¹.

A operação desses robôs geralmente envolve uma interface de controle baseada em visão computacional e aprendizado de máquina. Essas ferramentas permitem ao sistema replicar os movimentos do operador com precisão, muitas vezes com feedback visual ou tátil em tempo real, o que melhora a qualidade da interação.



Figura 1.1: *Robô-cachorro policial teleoperado pela SWAT de Los Angeles.*

Os principais benefícios da teleoperação incluem:

1. **Telecooperação:** controle remoto de atuadores a partir de locais seguros;
2. **Flexibilidade e mobilidade:** capacidade de operar em espaços apertados ou hostis;
3. **Redução de riscos:** menor exposição dos operadores a situações perigosas.

Esse modelo já apresenta impactos significativos em áreas como saúde, manutenção remota, missões espaciais e industriais.

¹Relato divulgado por telejornais em novembro de 2023 sobre a atuação da SWAT de Los Angeles.

1.2 Definição do Problema

Com o avanço das tecnologias de teleoperação, tornaram-se, nos últimos anos, mais disponíveis, dentro do estado da arte, soluções que integram visão computacional com manipulação robótica para reprodução precisa de sinais manuais complexos, tais como aquelas utilizadas na Língua Brasileira de Sinais (Libras). Em contrapartida, a ausência de interfaces intuitivas e acessíveis limita o uso dessas tecnologias em contextos assistivos, educacionais e de acessibilidade comunicacional.

Diante desse cenário, surge a seguinte questão de pesquisa:

É possível desenvolver e validar um sistema de teleoperação que utilize visão computacional para reprodução biomimética de gestos manuais por meio de uma mão robótica antropomórfica de cinco dedos em tempo real e reproduzi-los, de forma remota e sincronizada?

1.3 Motivação

A motivação deste trabalho está associada ao desenvolvimento de soluções que promovam acessibilidade comunicacional e inclusão, especialmente para pessoas surdas ou com deficiências motoras. O uso de uma interface visual para reconhecimento de gestos em Libras pode abrir caminhos para:

1. **Inclusão digital e comunicacional:** viabilizando que intérpretes robóticos atuem em ambientes remotos;
2. **Autonomia e acessibilidade:** permitindo que usuários com mobilidade reduzida operem dispositivos a distância por meio de gestos;
3. **Inovação em robótica assistiva:** propondo uma abordagem que dispense o uso de sensores vestíveis;
4. **Contribuição científica e tecnológica:** explorando o potencial da teleoperação aliada à visão computacional para replicação de sinais linguísticos.

1.4 Objetivos

1.4.1 Objetivo Geral

Investigar e validar a viabilidade de teleoperar uma mão robótica por meio do reconhecimento visual de gestos da Libras capturados por câmera, sem a necessidade de dispositivos vestíveis.

1.4.2 Objetivos Específicos

1. Desenvolver uma interface visual que reconheça, em tempo real, movimentos dos dedos correspondentes a sinais da Libras;

-
2. Projetar um algoritmo de mapeamento dos gestos capturados para comandos motores;
 3. Implementar a comunicação entre o sistema de visão computacional e os servomotores da mão robótica via rede local;
 4. Avaliar a responsividade e a precisão do sistema em replicar os sinais com fidelidade;
 5. Demonstrar a viabilidade da proposta em ambiente de simulação controlada.

1.5 Visão Geral do Método Proposto

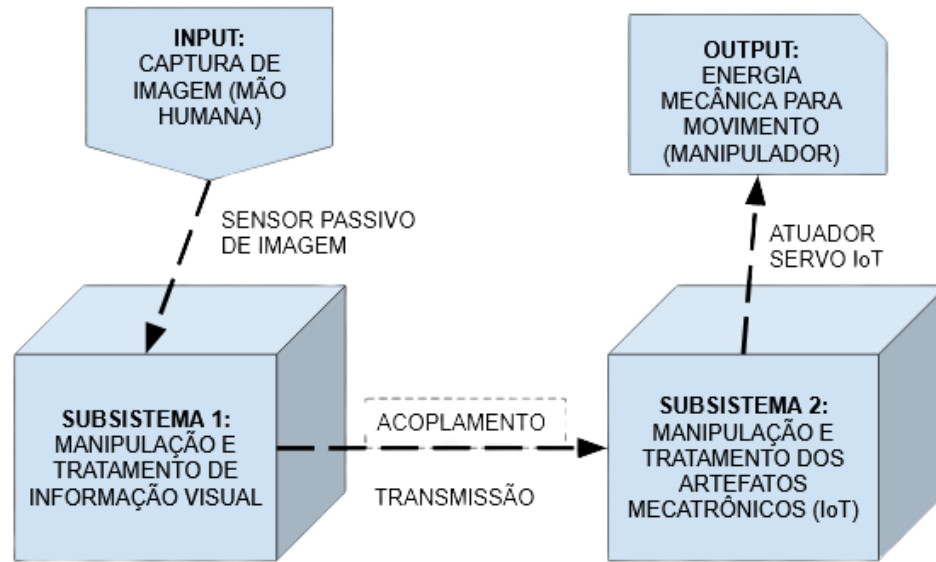


Figura 1.2: *Diagrama de Componentes UML do sistema proposto*

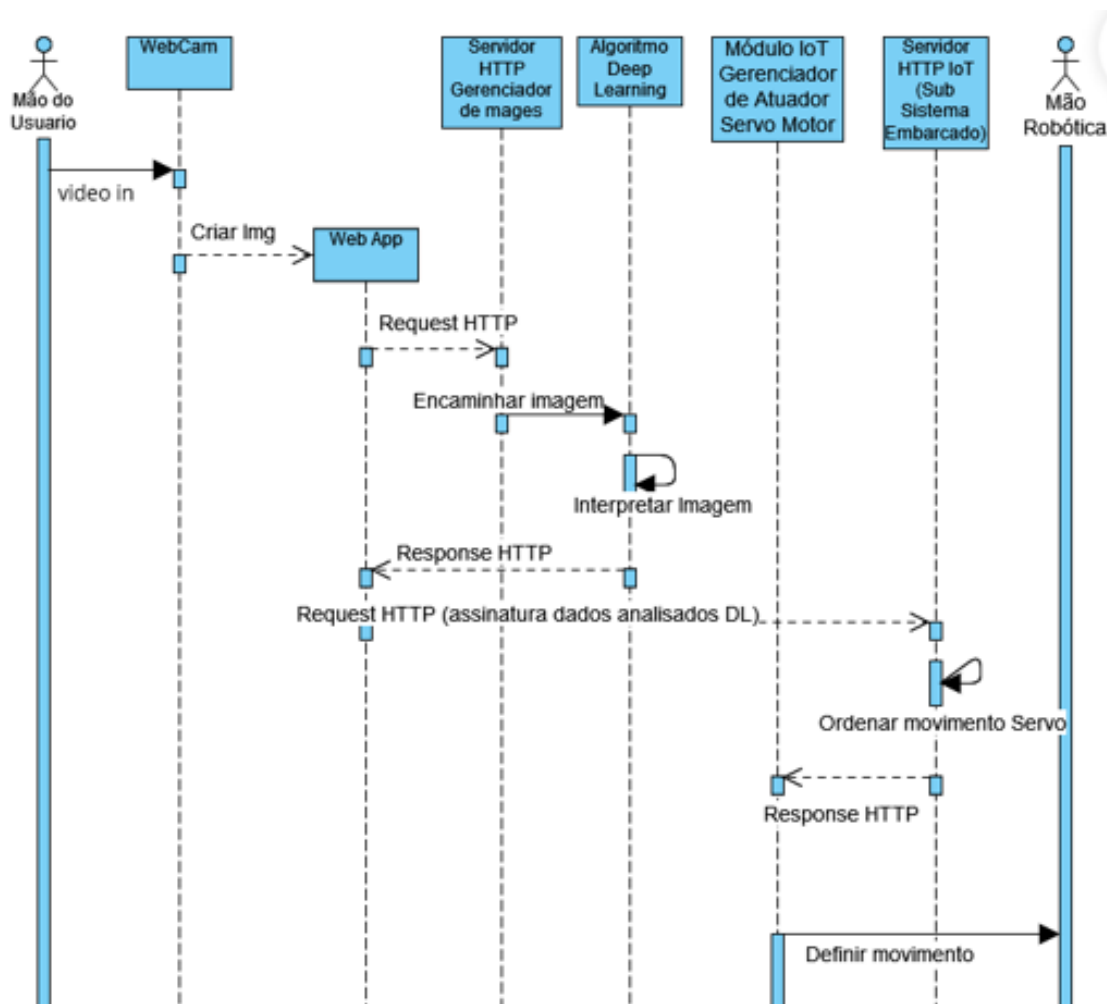
O sistema proposto compreende dois módulos principais: entrada (captura de imagem) e saída (dispositivo robótico). A Figura 1.2 apresenta essa arquitetura.

A Figura 1.3 ilustra o fluxo das operações por meio de um Diagrama de Sequência UML. O sistema realiza:

- Captura e análise de imagem;
- Processamento dos dados por algoritmo de aprendizado de máquina;
- Transmissão das informações ao servidor de controle;
- Geração de comandos para acionamento dos servos motores.

1.6 Organização da Proposta

A dissertação apresenta-se estruturada da seguinte forma: O Capítulo 2 apresenta os fundamentos teóricos relacionados ao reconhecimento de gestos e à visão computacional. Em seguida, o Capítulo 3 descreve a arquitetura do sistema e a metodologia proposta para o desenvolvimento da solução. O Capítulo 4 discute os trabalhos correlatos, estabelecendo comparações com abordagens existentes na literatura. O Capítulo 5 trata do detalhamento da implementação técnica, enquanto o Capítulo 6 aborda a preparação do ambiente utilizado para os testes experimentais. Por fim, o Capítulo 7 expõe os resultados obtidos, as conclusões do estudo e as propostas para trabalhos futuros.

Figura 1.3: *Diagrama de Sequência UML*

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta os conceitos fundamentais necessários para a compreensão deste trabalho, fornecendo base teórica sobre visão computacional, reconhecimento de gestos, interação humano-máquina e modelagem de sistemas. Os temas abordados incluem segmentação de imagem, segmentação interativa, reconhecimento de gestos manuais, detecção de pontos de referência da mão, incorporação de imagem e classificação de imagem, além de metodologias de modelagem usando SysML.

2.1 Visão Computacional: Conceitos e Aplicações

2.1.1 Fundamentos da Robótica

A robótica é o campo de estudo dedicado ao desenvolvimento de agentes físicos autônomos, denominados robôs, capazes de executar tarefas através da manipulação do ambiente físico (Russell e Norvig 2003). Estes sistemas são equipados com componentes especializados que lhes permitem interagir efetivamente com o mundo real:

Sistemas de Atuação: Os robôs utilizam efetadores como pernas, rodas, articulações e garras para exercer forças físicas sobre o ambiente. Estes componentes são responsáveis pela execução das ações planejadas pelo sistema de controle.

Sistemas de Percepção: Para compreender e navegar em seus ambientes, os robôs empregam diversos tipos de sensores. Os sensores passivos, como câmeras RGB e RGB-D, capturam informações visuais do ambiente. Os sensores ativos, incluindo ultrassom e LIDAR, emitem sinais para medir distâncias e mapear o espaço circundante. Adicionalmente, sensores proprioceptivos, como giroscópios e acelerômetros, fornecem informações sobre o movimento e orientação do próprio robô.

A classificação dos robôs pode ser organizada em três categorias principais, cada uma adequada a

diferentes tipos de aplicações:

1. **Manipuladores:** Também conhecidos como braços robóticos, são sistemas fisicamente fixos em uma posição específica. Estes robôs são amplamente utilizados em aplicações industriais de montagem, soldagem e manipulação de materiais, onde precisão e repetibilidade são fundamentais.
2. **Robôs Móveis:** Caracterizados pela capacidade de locomoção em seus ambientes operacionais, utilizam diferentes mecanismos de propulsão como rodas, esteiras, pernas ou hélices, dependendo do meio em que operam (terrestre, aquático ou aéreo).
3. **Manipuladores Móveis:** Representam a convergência das duas categorias anteriores, combinando capacidades de locomoção com habilidades de manipulação. Os robôs humanoides são exemplos notáveis desta categoria, oferecendo maior versatilidade para executar tarefas complexas em ambientes projetados para humanos ([Russell e Norvig 2003](#)).

2.1.2 Visão Robótica

A visão robótica é um campo interdisciplinar que combina princípios da robótica e visão computacional para permitir que robôs percebam, interpretem e respondam ao ambiente visual ([Robinson et al. 2023](#)). Este campo representa um componente crítico para o desenvolvimento de sistemas robóticos autônomos capazes de interagir efetivamente com humanos e ambientes complexos.

2.1.2.1 Componentes Fundamentais

A implementação da visão robótica requer a integração de componentes de hardware e software especializados, capazes de captar, processar e interpretar informações visuais do ambiente. Os principais elementos incluem:

- **Sensores Visuais:** Incluem câmeras RGB para captura de imagens coloridas e sensores RGB-D que adicionam informações de profundidade, proporcionando percepção tridimensional do ambiente.
- **Algoritmos de Processamento:** Responsáveis por traduzir dados visuais brutos em informações úteis para tomada de decisões e geração de comandos de controle para o robô.
- **Sistemas de Interpretação:** Permitem que o robô compreenda o contexto visual, identifique objetos, pessoas e suas ações, possibilitando respostas apropriadas às situações observadas.

2.1.2.2 Vantagens Estratégicas

A visão robótica oferece capacidades únicas que melhoram significativamente o desempenho robótico. Como alguns elementos de aprimoramento oferecidos por ela, temos:

1. **Controle Adaptativo de Câmera:** Permite ajustar automaticamente parâmetros como foco, zoom e orientação para otimizar a qualidade das informações visuais capturadas.

2. **Mobilidade Inteligente:** Facilita a navegação autônoma através da análise visual do ambiente, detecção de obstáculos e planejamento de trajetórias.
3. **Coleta Dinâmica de Informações:** Possibilita a adaptação do ponto de vista do robô para obter informações visuais adicionais conforme necessário (Arenas *et al.* 2017), (Choudhary e Chethan 2015), (Luo *et al.* 2019), (Ikai *et al.* 2016).

2.1.2.3 Avanços Recentes e Integração Multimodal

O campo tem experimentado progressos significativos no processamento de dados multimodais, integrando informações visuais com outros tipos de sensores para tomada de decisões mais robustas (Pierson e Gashler 2017), (Sünderhauf *et al.* 2018). Esta abordagem multimodal permite que robôs desenvolvam uma compreensão mais completa e contextual de seus ambientes.

A visão robótica também facilita formas mais naturais de interação humano-robô, eliminando a necessidade de interfaces complexas ou dispositivos especiais para comunicação. Esta capacidade é particularmente valiosa para aplicações assistivas e colaborativas, onde a naturalidade da interação é fundamental (Schmidt e Fox 2020).

2.1.2.4 Aplicações e Desenvolvimentos Contemporâneos

A visão computacional tem desempenhado um papel estratégico na evolução da robótica inteligente, passando a ser assumida como uma interface crítica entre a percepção sensorial e a ação autônoma. A crescente sofisticação na capacidade de extração e interpretação de informações visuais em tempo real tem se concretizado a partir da convergência do uso de algoritmos de aprendizado profundo, sensores de alta resolução e arquiteturas computacionais otimizadas.

Esse avanço tem contribuído decisivamente para o desenvolvimento de sistemas robóticos mais responsivos, adaptativos e contextualmente conscientes, com potencial de aplicação em ambientes complexos, dinâmicos e não estruturados. Paralelamente, observa-se uma tendência consolidada em direção à integração de múltiplos modos de entrada perceptiva, articulando visão, linguagem, som e tato em sistemas colaborativos centrados no ser humano. Tal abordagem tem favorecido o desenvolvimento de interfaces homem-máquina mais naturais e eficientes, especialmente no contexto de robótica assistiva, social e industrial.

Pesquisas recentes têm ampliado significativamente o escopo da visão robótica, impulsionando seu uso em contextos cada vez mais complexos e interativos. Essas investigações vêm consolidando a visão computacional como um componente essencial em sistemas robóticos inteligentes, promovendo avanços tanto em ambientes controlados quanto em cenários dinâmicos e não estruturados.

Ao incorporar capacidades de interpretação visual refinadas, os sistemas robóticos contemporâneos seguem progredindo no sentido de uma cognição operacional situada, na qual a percepção visual não apenas

detecta estímulos, mas também subsidia a inferência de intenções e a tomada de decisões adaptativas em tempo real. Dentre exemplos metodológicos de tais refinamentos, encontram-se:

- **Interação Baseada em Gestos:** Um dos campos mais promissores é o desenvolvimento de interfaces naturais que permitem a comunicação entre humanos e máquinas por meio de gestos manuais ou corporais. Esses sistemas são capazes de interpretar movimentos em tempo real, possibilitando, por exemplo, o controle de robôs colaborativos ou assistivos sem a necessidade de dispositivos físicos intermediários (Xia *et al.* 2019). Essa abordagem tem aplicações diretas em robótica assistiva, educação inclusiva e interfaces acessíveis para pessoas com deficiência.
- **Colaboração Multimodal:** A integração de múltiplas fontes sensoriais — como visão, áudio, sensores táteis e informações contextuais — tem sido essencial para enriquecer a capacidade de interpretação do ambiente por parte dos robôs. A colaboração multimodal possibilita uma interação mais fluida e contextualizada entre humanos e sistemas autônomos, especialmente em tarefas que exigem alto grau de percepção e tomada de decisão adaptativa (Jaimes e Sebe 2007). Esse paradigma é crucial em aplicações como teleoperação, robôs domésticos e assistentes sociais.
- **Reconhecimento de Ações Humanas:** O reconhecimento automático de ações humanas é outra vertente fundamental, voltada para a identificação de padrões de movimento e a compreensão da intenção do usuário. Sistemas capazes de reconhecer atividades — como caminhar, sentar, acenar ou manipular objetos — têm sido aplicados em vigilância inteligente, monitoramento de idosos e apoio à segurança ocupacional, oferecendo respostas contextualmente relevantes e, muitas vezes, preditivas (Zhang *et al.* 2019).
- **Aplicações Específicas:** A visão robótica também tem encontrado espaço em domínios especializados, como a navegação autônoma de veículos aéreos não tripulados (UAVs), onde algoritmos de visão são empregados para localização, mapeamento e rastreamento de alvos em tempo real (Bonin-Font *et al.* 2008). Além disso, destaca-se o desenvolvimento de sistemas de aprendizado adaptativo, que capacitam robôs a melhorar sua percepção visual com base em experiências prévias, tornando-se progressivamente mais eficientes na execução de tarefas complexas em ambientes variáveis (Schmidt e Fox 2020).

Apesar dos avanços significativos, ainda existe uma lacuna na literatura quanto à integração abrangente de técnicas de visão robótica para colaboração humano-robô em cenários do mundo real, especialmente no que se refere à avaliação de desempenho em ambientes não estruturados (Schmidt e Fox 2020).

2.2 Reconhecimento de Gestos

2.2.1 Definição e Classificação de Gestos

Gestos podem ser definidos como movimentos intencionais de partes do corpo humano, especialmente mãos, braços, cabeça ou torso, realizados com o propósito de transmitir informações, expressar emoções ou comandar ações (Robinson *et al.* 2023). O reconhecimento de gestos representa a capacidade de sistemas computacionais interpretarem e compreenderem o significado desses movimentos corporais.

Entre os diversos tipos de gestos, aqueles realizados com as mãos são os mais amplamente utilizados em aplicações de interface humano-computador devido às suas características naturais e intuitivas. Os gestos manuais podem ser classificados em duas categorias principais:

1. **Gestos Estáticos:** Caracterizados por posturas fixas da mão, onde a informação relevante está contida na configuração espacial dos dedos e na posição da mão, sem envolver movimento temporal significativo.
2. **Gestos Dinâmicos:** Envolvem movimento temporal das mãos ou dedos, podendo incluir movimento global (toda a mão se move através do espaço) ou movimento local (apenas os dedos se movem relativamente à palma) (Ahmed *et al.* 2008).

2.2.2 Reconhecimento de Gestos Manuais

O Reconhecimento de Gestos Manuais (HGR - *Hand Gesture Recognition*) constitui um subcampo especializado da visão computacional focado na interpretação de movimentos e posturas das mãos humanas. Este campo tem como objetivo desenvolver sistemas que proporcionem interfaces naturais e eficientes entre humanos e máquinas.

2.2.2.1 Motivação e Aplicações

Os sistemas de HGR oferecem vantagens significativas em termos de conveniência, velocidade e economia para usuários. Principais aplicações incluem:

- **Interfaces de Controle Remoto:** Permitem operação de dispositivos complexos através de gestos simples, sem necessidade de contato físico.
- **Linguagem de Sinais:** Facilitam a comunicação para pessoas com deficiência auditiva através da interpretação automática de sinais.
- **Controle Robótico:** Possibilitam comandar robôs de forma intuitiva através de gestos naturais.

- **Interfaces de Jogos:** Criam experiências imersivas onde movimentos corporais controlam elementos do jogo.
- **Aplicações Assistivas:** Desenvolvem interfaces adaptadas para pessoas com limitações de mobilidade.

2.2.2.2 Abordagens Metodológicas

O reconhecimento de gestos manuais pode ser implementado através de duas abordagens principais:

1. **HGR Dinâmico:** Focado no reconhecimento de ações através da análise de sequências de vídeo, capturando a evolução temporal dos movimentos. Esta abordagem é particularmente adequada para interpretar gestos complexos que envolvem trajetórias específicas.
2. **HGR Estático:** Concentra-se no reconhecimento de posturas através de imagens individuais. Esta abordagem oferece vantagens em termos de eficiência computacional e é adequada para implementação em dispositivos com recursos limitados, como *smartphones* e sistemas embarcados.

2.2.2.3 Integração com Estimativa de Pose

Na robótica e na visão computacional, o termo POSE refere-se à descrição completa da configuração de um corpo físico em um espaço tridimensional, englobando posição (localização no espaço, geralmente representada por coordenadas x, y, z) e orientação (a direção ou rotação desse corpo, expressa em ângulos de Euler, quaternions ou matrizes de rotação). Diferentemente de uma simples coordenada espacial, a **pose** fornece uma representação holística do objeto ou do agente robótico, sendo crucial para tarefas de navegação, manipulação e interação com o ambiente. Em outras palavras, conhecer a **pose** significa compreender onde o corpo está e como ele está orientado, o que constitui a sua configuração total.

No contexto da estimativa de **pose** aplicada ao reconhecimento de gestos, esse conceito é expandido para a detecção de estruturas corporais, como articulações das mãos, dedos, braços e outras partes relevantes. Através da análise dessas articulações, é possível modelar um “esqueleto virtual” que descreve tanto a posição relativa das partes quanto suas orientações dinâmicas, fornecendo informações estruturais detalhadas sobre o movimento humano.

Desenvolvimentos recentes em estimativa de **pose** abriram novas possibilidades para o campo do reconhecimento gestual, permitindo recriar e classificar gestos com maior precisão e robustez. Frameworks modernos de aprendizado profundo, como PyTorch e MediaPipe, disponibilizam modelos pré-treinados capazes de gerar representações esqueléticas das mãos em tempo real. Esses modelos, treinados em grandes volumes de dados, conseguem capturar padrões complexos associados a diferentes configurações manuais e expressões gestuais, traduzindo-os em vetores de pose que podem ser utilizados em sistemas de classificação, controle robótico ou interfaces homem-máquina.

Assim, a integração entre reconhecimento de gestos e estimativa de **pose** não apenas melhora a acurácia dos sistemas, mas também aproxima o comportamento artificial da percepção humana, viabilizando aplicações avançadas em áreas como robótica assistiva, realidade aumentada e interação natural baseada em visão.

Frameworks modernos de aprendizado profundo, como *PyTorch* e *MediaPipe*, incorporam modelos pré-treinados capazes de gerar representações esqueléticas das mãos em tempo real. Estes modelos são treinados em grandes conjuntos de dados e podem aprender padrões complexos associados a diferentes tipos gestuais e configurações manuais.

2.2.3 Aplicações em Robótica Manipuladora

A integração entre técnicas de reconhecimento de gestos e sistemas de robótica manipuladora tem se consolidado como uma das vertentes mais promissoras na interação homem-máquina. Por meio da interpretação de gestos manuais capturados por sensores visuais, é possível estabelecer uma interface de controle não intrusiva, intuitiva e altamente responsiva, capaz de traduzir ações humanas naturais em comandos para manipulação robótica precisa. Tal abordagem é particularmente relevante em contextos onde o uso de dispositivos físicos é impraticável ou indesejado, como em ambientes cirúrgicos, industriais ou assistivos.

2.2.3.1 Comandos Gestuais Básicos

Neste subgrupo de comandos, os gestos manuais são utilizados de forma direta para controlar ações fundamentais de um braço robótico. Cada gesto é associado a uma instrução específica, geralmente de baixo nível, permitindo controle imediato e reativo das articulações ou atuadores. Esses comandos são a base da interação gestual e normalmente envolvem configurações estáticas da mão (posturas) ou movimentos simples que podem ser reconhecidos com alta confiabilidade.

- **Palma Aberta:** Este gesto, facilmente reconhecido por sistemas de visão computacional, é comumente utilizado como um comando de ativação, pausa ou elevação do sistema robótico. Ele pode sinalizar um estado receptivo, em que o robô aguarda instruções subsequentes, funcionando como um gesto de prontidão (Fareed *et al.* 2015), (Martin e Moutarde 2019).
- **Movimento de Rotação:** A rotação da mão ou punho pode ser interpretada como uma solicitação de giro das juntas do robô, sendo especialmente útil para tarefas que exigem reorientação do end-effector ou manipulação angular de objetos (Martin e Moutarde 2019).
- **Gestos de Pinça:** A simulação de um movimento de pinça com os dedos indicador e polegar é tipicamente utilizada para controlar a abertura e o fechamento de garras robóticas. Esse comando é essencial para operações de preensão e soltura, com aplicações em manipulação fina e montagem (Fareed *et al.* 2015), (Lima *et al.* 2019).

- **Apontar:** O gesto de apontar, com um dedo estendido, permite indicar objetos ou regiões de interesse dentro do campo de visão do robô. Ele é amplamente utilizado para guiar ações subsequentes, como alcançar, posicionar ou manipular elementos específicos no ambiente (Quintero *et al.* 2015), (Rautaray e Agrawal 2015), (Song *et al.* 2017).

2.2.3.2 Controle Multimodal

O controle multimodal amplia a expressividade e a precisão do sistema ao combinar gestos manuais com outros sinais corporais, como movimentos de cabeça, postura ou orientação do tronco. Esta abordagem permite a criação de sistemas de controle hierárquico, em que diferentes modalidades sensoriais colaboram para interpretar comandos compostos. O uso simultâneo de múltiplos canais de entrada favorece a redundância e melhora a robustez do sistema em cenários dinâmicos ou ruidosos, promovendo uma interação mais natural e fluida (Lima *et al.* 2019), (Martin e Moutarde 2019).

2.2.3.3 Aplicações Colaborativas

As aplicações colaborativas envolvem cenários em que humanos e robôs compartilham o mesmo espaço físico e trabalham em conjunto para alcançar objetivos comuns. Nessas situações, os gestos não apenas servem como comandos, mas também como sinais de intenção e coordenação, fundamentais para a fluidez da cooperação. Abaixo, destacam-se exemplos representativos de como gestos manuais têm sido integrados a diferentes contextos colaborativos:

1. **Assistência Culinária:** Robôs assistivos podem interpretar gestos humanos para identificar ingredientes, seguir etapas de receitas e auxiliar em tarefas como misturar, cortar ou servir, contribuindo para a autonomia de usuários em ambientes domésticos ou profissionais (Quintero *et al.* 2015).
2. **Manipulação Coordenada:** Em sistemas com múltiplos braços robóticos, os gestos permitem a seleção de objetos e a coordenação da entrega ou recepção de componentes, promovendo manipulação sincronizada e inteligente com base na percepção visual dos gestos (Ikai *et al.* 2016).
3. **Ambientes Industriais:** Em linhas de montagem, gestos manuais são utilizados para iniciar, ajustar ou interromper operações sem a necessidade de interfaces físicas. Esta forma de controle reduz o tempo de resposta e aumenta a segurança ao eliminar pontos de contato diretos com máquinas em operação.

2.3 Interação Humano-Máquina através de Visão Computacional

A visão computacional desempenha papel fundamental no desenvolvimento de sistemas capazes de compreender e interagir efetivamente com o mundo real. Esta capacidade é especialmente importante para criar interfaces naturais entre humanos e máquinas, estabelecendo uma linguagem comum baseada em informações visuais. Ao permitir que os sistemas interpretem sinais visuais — como gestos, expressões faciais e movimento corporal — a visão computacional contribui de forma decisiva para o avanço da interação humano-máquina (IHM), promovendo acessibilidade, automação inteligente e respostas adaptativas em tempo real.

2.3.1 Tecnologias de Sensoriamento Visual

A aquisição de dados visuais de qualidade é um pré-requisito essencial para o sucesso de qualquer sistema de visão computacional. Para isso, diferentes tecnologias de sensoriamento são utilizadas, cada uma com características específicas quanto à resolução, profundidade, fidelidade espectral e capacidade de operar sob diferentes condições ambientais. A seguir, descrevem-se os principais tipos de sensores empregados nesses sistemas e suas respectivas funcionalidades:

Câmeras RGB: Capturam informações visuais em três canais de cor (vermelho, verde e azul), fornecendo representações detalhadas e coloridas do ambiente. Estas câmeras são fundamentais para tarefas de reconhecimento de objetos, análise de cenas e interpretação de atividades humanas. Sua ampla disponibilidade e baixo custo as tornam ideais para aplicações em larga escala, como vigilância, reconhecimento de padrões e controle visual de robôs.

Sensores RGB-D: Combinam informações de cor com dados de profundidade (depth), fornecendo uma percepção tridimensional do ambiente em tempo real. Essa tecnologia é especialmente útil em contextos onde a compreensão espacial precisa ser acurada, como em navegação robótica, reconstrução 3D e manipulação de objetos em ambientes complexos ou desestruturados.

2.3.2 Técnicas de Processamento Visual

Uma vez adquiridos os dados visuais, é necessário aplicar técnicas de processamento para extrair informações significativas e interpretáveis. Essas técnicas compõem o núcleo das operações da visão computacional e são projetadas para detectar padrões, reconhecer entidades e inferir relações espaciais e temporais. As abordagens a seguir são amplamente empregadas em aplicações práticas de interação homem-máquina:

1. **Detecção de Objetos:** Refere-se à capacidade de localizar e identificar automaticamente objetos específicos dentro de imagens estáticas ou sequências de vídeo. Esta técnica fornece coordenadas

espaciais (caixas delimitadoras) e rótulos de classe, permitindo que o sistema reconheça múltiplos elementos do ambiente e reaja de forma seletiva a diferentes estímulos visuais.

2. **Classificação de Imagens:** Consiste em atribuir a uma imagem inteira uma categoria semântica dentre um conjunto de classes predefinidas. Essa técnica é utilizada para entender o conteúdo global da imagem, o que pode servir de base para decisões contextuais, como a ativação de modos operacionais diferentes em um robô assistivo, dependendo do ambiente identificado.
3. **Segmentação Semântica:** Vai além da classificação de imagens ao realizar uma análise pixel a pixel, associando cada ponto da imagem a uma categoria semântica específica. Isso permite que o sistema tenha uma compreensão rica e estruturada da cena, sendo especialmente útil para navegação autônoma, detecção de obstáculos e manipulação baseada em contexto espacial (Pierson e Gashler 2017), (Viola e Jones 2004).

2.3.3 Análise Visual de Comportamento Humano

Uma das áreas mais desafiadoras e relevantes da visão computacional é a análise do comportamento humano com base em dados visuais. Esta área visa compreender, em tempo real, as intenções, emoções e ações dos usuários, criando as condições para sistemas interativos mais responsivos, adaptáveis e empáticos. O monitoramento visual do comportamento humano é particularmente valioso em aplicações assistivas, segurança, entretenimento interativo e robótica social. A seguir, destacam-se as técnicas fundamentais empregadas neste domínio:

Detecção Facial: Envolve a identificação e localização de rostos humanos em imagens, funcionando como etapa inicial para tarefas subsequentes como reconhecimento de identidade, análise emocional e verificação de atenção. Algoritmos modernos permitem detecção robusta mesmo em ambientes com iluminação variável ou múltiplos indivíduos (Viola e Jones 2004).

Estimativa de Pose Corporal: Trata-se da inferência da posição tridimensional das articulações do corpo humano com base em imagens bidimensionais. Essa técnica permite reconstruir posturas e acompanhar gestos corporais, viabilizando aplicações como controle gestual, análise biomecânica e monitoramento de ergonomia em ambientes industriais (Cao *et al.* 2017).

Rastreamento de Movimento: Refere-se ao acompanhamento contínuo da posição e orientação de objetos ou indivíduos ao longo do tempo. Essa funcionalidade é crucial para reconhecer padrões de movimento, detectar comportamentos anômalos e prever trajetórias em ambientes dinâmicos, como em sistemas de vigilância ou interação com robôs móveis (Henriques *et al.* 2015).

Estas capacidades de análise visual criam oportunidades para o desenvolvimento de sistemas que compreendem ações humanas de maneira contextual e podem responder de forma apropriada às necessidades do usuário. A integração dessas técnicas forma a base de uma interação homem-máquina mais natural, efí-

ente e centrada no comportamento, aproximando o desempenho das máquinas da complexidade da percepção humana.

2.4 A Linguagem SysML para Modelagem de Sistemas

2.4.1 Definição e Contexto

A *Systems Modeling Language* (SysML) é uma linguagem gráfica de modelagem desenvolvida especificamente para especificação, análise, design e verificação de sistemas complexos. Como extensão especializada da *Unified Modeling Language* (UML), a SysML incorpora conceitos específicos para modelagem de sistemas físicos, expandindo as capacidades da UML além do domínio puramente computacional.

A SysML foi desenvolvida conjuntamente pelo *Object Management Group* (OMG) e pelo *International Council on Systems Engineering* (INCOSE)¹, organizações responsáveis pela padronização e formalização de linguagens de modelagem em engenharia de sistemas.

2.4.2 Vantagens e Aplicações da SysML

A utilização da SysML oferece benefícios significativos para o desenvolvimento de sistemas complexos:

1. **Compreensão Estrutural:** Através de diagramas de blocos e diagramas de contexto, a SysML permite representar claramente a arquitetura do sistema e os relacionamentos entre seus componentes, facilitando o entendimento da organização geral.
2. **Gestão de Requisitos:** Fornece mecanismos robustos para documentar, organizar e rastrear requisitos do sistema, garantindo que todas as especificações sejam capturadas e mantidas ao longo do ciclo de vida do projeto.
3. **Modelagem Comportamental:** Permite representar o comportamento dinâmico do sistema através de diagramas de sequência, diagramas de estados e outros diagramas comportamentais, mostrando como o sistema responde a diferentes estímulos e eventos.
4. **Suporte a Sistemas Complexos:** Adequada para modelagem de sistemas que integram componentes físicos e computacionais, incluindo aspectos de arquitetura, interfaces e alocação de recursos.
5. **Comunicação Padronizada:** Estabelece uma linguagem comum entre diferentes equipes e disciplinas envolvidas no projeto, reduzindo ambiguidades e mal-entendidos.
6. **Rastreabilidade:** Permite acompanhar como requisitos são implementados no sistema, proporcionando controle efetivo sobre o atendimento às especificações.

¹<https://www.omgsysml.org/what-is-sysml.htm>

2.4.3 Diferenças entre SysML e UML

Embora baseada na UML (Unified Modeling Language), a SysML (Systems Modeling Language) apresenta características distintivas que a tornam mais adequada para modelagem de sistemas multidisciplinares e físicos. A UML, tradicionalmente voltada ao desenvolvimento de software, tem como foco principal estruturas e comportamentos típicos de sistemas orientados a objetos. Por sua vez, a SysML expande e adapta esses conceitos, introduzindo novos elementos e diagramas que permitem capturar aspectos estruturais, funcionais, de requisitos e de análise paramétrica de sistemas complexos, tal como ilustra a Figura 2.1. Essa diferenciação torna a SysML especialmente valiosa em áreas como engenharia mecatrônica, aeroespacial, automotiva e robótica.

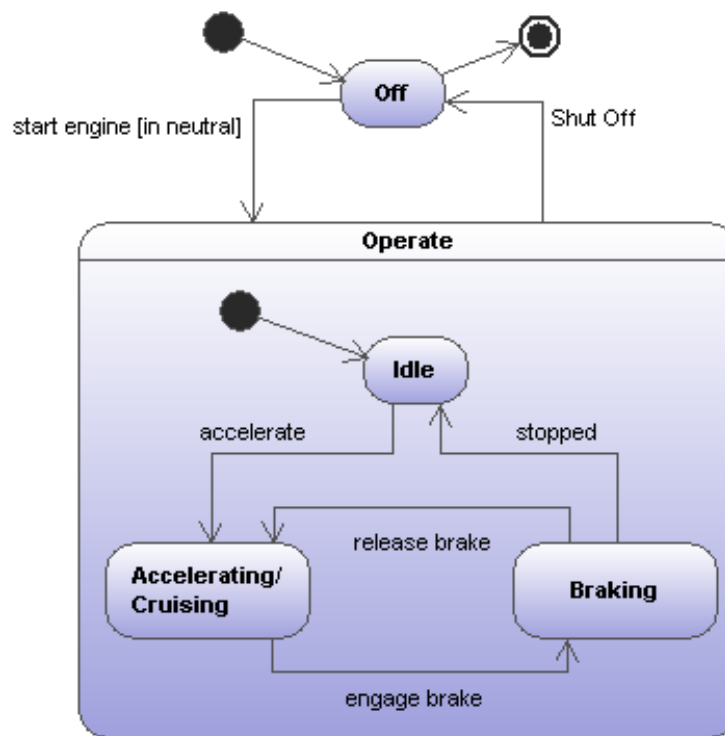


Figura 2.1: Diagrama SysML de State Machine modelando um sistema de controle de um automóvel

2.4.3.1 Objetos de Modelagem

A principal diferença entre as linguagens pode ser observada no conjunto de elementos disponíveis para a construção de modelos. Enquanto a UML é centrada em conceitos como classes, objetos e atores tipicamente associados a software, a SysML oferece uma gramática de modelagem mais ampla, capaz de representar componentes físicos, fluxos de energia, variáveis de desempenho e interações entre disciplinas de engenharia.

A seguir, destacam-se os principais objetos de modelagem em ambas as linguagens:

- **UML:** Focada principalmente em classes, objetos, casos de uso e atores relacionados ao desenvolvimento de software. Esses elementos permitem representar entidades lógicas do sistema, suas interações e comportamentos esperados sob diferentes cenários funcionais.
- **SysML:** Incorpora elementos adicionais como *blocos*, *partes*, *portas*, *requisitos*, *parâmetros* e *restrições*, projetados para capturar a natureza física e quantitativa dos sistemas. Os *blocos*, por exemplo, representam unidades modulares que podem ser hardware, software, pessoas, subsistemas ou qualquer entidade do domínio de engenharia. As *portas* e *conectores* facilitam a modelagem das interfaces, enquanto os *diagramas de requisitos* e *paramétricos* integram diretamente a modelagem funcional com a análise de desempenho.

2.4.3.2 Diagramas Específicos

A SysML introduz três tipos de diagramas que não estão presentes na UML, permitindo a representação de aspectos fundamentais de engenharia de sistemas, como composição estrutural hierárquica, conexões internas e análise quantitativa. Estes diagramas complementam os já existentes na UML, fortalecendo a capacidade da linguagem de capturar especificações completas de sistemas físico-cibernéticos. A seguir, são descritos os três principais diagramas exclusivos da SysML:

1. **Diagrama de Definição de Bloco (BDD):** Representa elementos estruturais chamados *blocos* e suas relações hierárquicas (como composição, especialização e associação). Os blocos constituem a unidade fundamental da SysML, podendo modelar entidades físicas (como sensores, atuadores, módulos eletrônicos), componentes de software, operadores humanos ou funções abstratas do sistema. Esse diagrama fornece uma visão de alto nível da arquitetura estrutural.
2. **Diagrama Interno de Bloco (IBD):** Detalha as conexões internas entre as partes de um bloco definido no BDD. Permite representar graficamente as interações entre subcomponentes por meio de conectores e interfaces. É essencial para descrever como as partes do sistema colaboram para realizar comportamentos desejados, sendo útil em engenharia elétrica, hidráulica, mecânica e de controle.
3. **Diagrama Paramétrico (PD):** Permite a modelagem de restrições matemáticas sobre as propriedades do sistema, utilizando equações e relações paramétricas. Essa capacidade torna possível realizar análises de engenharia como estimativas de desempenho, consumo energético, confiabilidade, massa, custo, ou qualquer variável de interesse mensurável. Ele estabelece a ponte entre modelagem funcional e simulações quantitativas.

2.4.4 Taxonomia dos Diagramas SysML

A SysML oferece um conjunto abrangente de nove tipos de diagramas que podem ser agrupados segundo sua finalidade: estrutural, comportamental ou analítica. Esta taxonomia (cuja árvore sistêmica é apresentada na Figura 2.2) fornece aos engenheiros uma base metodológica consistente para decompor, analisar e validar sistemas complexos em diferentes estágios do ciclo de vida. Diferente da UML, cuja ênfase está em estruturas de software, a SysML busca capturar a totalidade do sistema, permitindo uma abordagem transdisciplinar. A seguir, apresenta-se a classificação desses diagramas segundo suas categorias funcionais:

2.4.4.1 Diagramas Estruturais

Estes diagramas descrevem a organização estática do sistema, suas partes constituintes, interconexões e hierarquias. São utilizados para representar arquiteturas físicas e lógicas, bem como a estrutura modular do sistema.

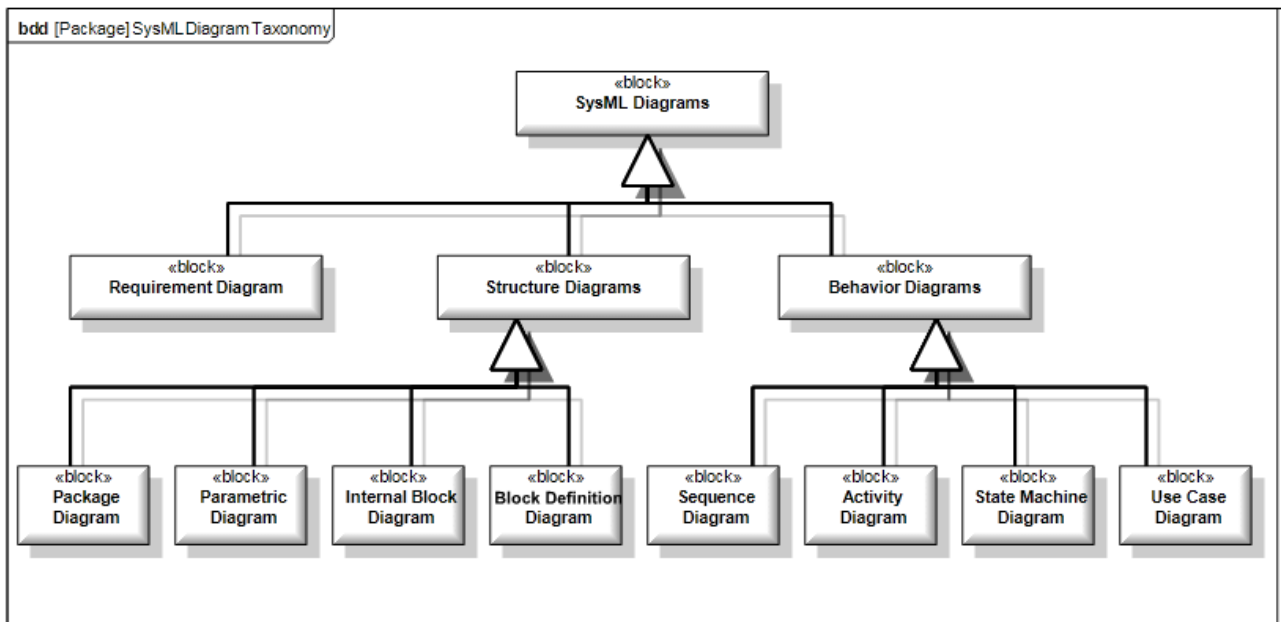


Figura 2.2: *Taxonomia dos Diagramas SysML*

- Diagrama de Definição de Bloco (BDD)
- Diagrama de Bloco Interno (IBD)
- Diagrama de Pacotes

2.4.4.2 Diagramas Comportamentais:

Capturam o comportamento dinâmico do sistema, incluindo fluxos de atividades, interações entre componentes, estados internos e respostas a estímulos externos. São fundamentais para especificar como o sistema opera ao longo do tempo.

- Diagrama de Caso de Uso
- Diagrama de Atividade
- Diagrama de Sequência
- Diagrama de Máquina de Estados

2.4.4.3 Diagramas de Requisitos e Análise

Esses diagramas estendem a capacidade da modelagem estrutural e comportamental ao integrar diretamente os requisitos de engenharia e os parâmetros quantitativos do sistema. São essenciais para garantir rastreabilidade, validação e suporte à análise multidisciplinar.

- Diagrama de Requisitos
- Diagrama Paramétrico

2.4.5 Funcionalidades Específicas dos Diagramas

Cada tipo de diagrama SysML atende a propósitos específicos na modelagem de sistemas [Delligatti \(2013\)](#):

- **Diagrama de Definição de Bloco:** Utilizado para representar hierarquias do sistema e classificações de componentes, mostrando as relações estruturais entre diferentes elementos.
- **Diagrama de Bloco Interno:** Especifica a estrutura interna de blocos individuais, detalhando conexões entre partes internas e suas interfaces.
- **Diagrama de Caso de Uso:** Fornece visão de alto nível da funcionalidade do sistema através da interação com atores externos.
- **Diagrama de Atividade:** Representa fluxos de controle e transformação de dados entre atividades, sendo útil para análise de processos.
- **Diagrama de Sequência:** Especifica interações temporais entre partes do sistema através de trocas de mensagens.
- **Diagrama de Máquina de Estados:** Modela comportamento baseado em estados e transições em resposta a eventos específicos.
- **Diagrama Paramétrico:** Expressa restrições matemáticas sobre propriedades do sistema, suportando análises quantitativas de engenharia.
- **Diagrama de Pacotes:** Organiza o modelo em hierarquias de contêineres, facilitando a gestão de modelos complexos.
- **Diagrama de Requisitos:** Documenta requisitos textuais e suas relações, integrando gestão de requisitos com modelagem do sistema.

A SysML também incorpora relações de alocação que permitem representar diferentes tipos de mapeamento, incluindo alocação de funções para componentes, componentes lógicos para físicos, e software para hardware, facilitando a rastreabilidade em projetos complexos.

2.5 Mapeamento Sistemático da Literatura

Esta seção apresenta uma análise sistemática e comparativa de tecnologias e métodos de visão computacional e visão robótica, oferecendo uma visão abrangente das metodologias identificadas no estado da arte. O objetivo é fornecer um panorama estruturado das descobertas, evoluções e características relevantes registradas na literatura especializada em reconhecimento de gestos manuais.

2.5.1 Análise Taxonômica de Metodologias

A sistematização das técnicas desenvolvidas recentemente no estado da arte revela características importantes sobre o campo da visão computacional aplicada à interação humana. A análise da Tabela 2.1 evidencia três aspectos fundamentais:

Diversidade Multidisciplinar: O campo da visão computacional evidencia características marcadamente multidisciplinares, abrangendo aplicações desde vigilância de segurança até assistência médica. Esta diversidade reflete a versatilidade das técnicas de visão computacional e sua capacidade de adaptação a diferentes domínios de aplicação.

Predominância do Aprendizado Profundo: A análise revela uma tendência clara em direção ao uso de técnicas de aprendizado profundo, particularmente Redes Neurais Convolucionais (CNNs) e Long Short-Term Memory (LSTM). Esta preferência metodológica reflete a eficácia comprovada dessas abordagens na classificação, segmentação e reconhecimento de padrões complexos em dados visuais.

Foco em Aplicações em Tempo Real: Métodos como YOLO (You Only Look Once), OpenPose e técnicas específicas para robôs aéreos demonstram a importância crescente de sistemas capazes de processar informações visuais em tempo real. Esta característica é fundamental para aplicações autônomas que requerem decisões rápidas e precisas baseadas em informações visuais.

O mapeamento elaborado por Robinson *et al.* (2023) oferece uma visão comparativa abrangente de métodos, finalidades e campos de aplicação, cobrindo áreas diversas como:

- Detecção de Objetos: Redmon *et al.* (2016)
- Classificação de Imagem: LeCun *et al.* (1998)
- Segmentação Semântica: He *et al.* (2018)
- Detecção de Humanos: Cao *et al.* (2017)
- Interação Gestual: Song *et al.* (2019)
- Colaboração Multimodal: Chen, Wang, Liang, Baltrušaitis, Zadeh e Morency (2017)
- Reconhecimento de Ações: Hochreiter e Schmidhuber (1997)
- Aplicações Robóticas Especializadas: Khoshelham e Elberink (2012), Lalonde *et al.* (2006)

Tabela 2.1: Áreas de Pesquisa em Visão Computacional e suas Aplicações

Área de Pesquisa	Técnica e Método	Utilidade e Finalidades	Vantagens	Estratégias	Campos de Aplicação
Deteção de Objetos	YOLO (You Only Look Once)	Deteção de objetos em tempo real	Alta velocidade e precisão	Alta velocidade e precisão	Vigilância de segurança, veículos autônomos
Classificação de Imagem	Redes Neurais Convolucionais (CNNs)	Classificação precisa de imagens	Aprendizado profundo, escalabilidade	pro-fundo, escalabilidade	Reconhecimento de objetos em fotos
Segmentação Semântica	Segmentação por Instância	Segmentação precisa de objetos	Deteção de objetos individuais	Deteção de objetos individuais	Identificação de objetos em imagens médicas
Deteção de Humanos	OpenPose	Rastreamento preciso de movimento	Rastreamento de pontos-chave	Rastreamento de pontos-chave	Análise de postura humana e interação
Interação Gestual	Reconhecimento de Gestos	Interação intuitiva e natural	Identificação de gestos, comunicação	Identificação de gestos, comunicação	Controle de dispositivos e jogos
Colaboração Multimodal	Fusão de Dados Multimodais	Compreensão holística do ambiente	Integração de informações sensoriais	Integração de informações sensoriais	Robótica, assistência médica, veículos autônomos
Reconhecimento de Ação	LSTM (Long Short-Term Memory)	Modelagem de sequências de ações	Reconhecimento de padrões temporais	Reconhecimento de padrões temporais	Análise de movimentos e atividades humanas
Reconhecimento de Gestos Manuais	CNNs Especializadas	Interação precisa com gestos	Deteção de gestos específicos	Deteção de gestos específicos	Controle de interfaces e robôs
Visão Robótica	Sensor RGB-D	Percepção tridimensional	Sensor versátil com profundidade	Sensor versátil com profundidade	Robótica, mapeamento, navegação
Visão Robótica	Algoritmos de Suporte	Tradução de imagens em sinais de controle	Adaptação a diferentes cenários	Adaptação a diferentes cenários	Orientação e controle de robôs
Visão para Robôs	Deteção de Objetos	Identificação para interação	Precisão na detecção, velocidade	Precisão na detecção, velocidade	Manipulação de objetos, segurança
Visão para Robôs	Aprendizado Adaptativo	Melhoria contínua de capacidades	Aprendizado contínuo, adaptação	Aprendizado contínuo, adaptação	Tarefas robóticas em evolução
Robôs Aéreos	Visão Computacional Aérea	Navegação autônoma aérea	Sensoriamento aéreo, mapeamento	Sensoriamento aéreo, mapeamento	Exploração, monitoramento aéreo

2.5.2 Análise Comparativa de Métodos em Visão Robótica

A análise especializada dos métodos aplicados em visão robótica revela tendências específicas que caracterizam o desenvolvimento atual desta área. O mapeamento apresentado na Tabela 2.2 evidencia três direções principais:

Integração Multimodal e Adaptação Contínua: A tendência dominante envolve a fusão de informações de múltiplas fontes sensoriais para tomada de decisões mais robusta. Esta abordagem é evidente nas categorias de "Tratamento de Dados Multimodais" e "Avanços na Detecção Robótica", demonstrando a importância de combinar dados visuais com outras fontes de informação para obter interpretações mais precisas do ambiente.

Aprendizado Adaptativo: O foco crescente em "Aprendizado para Visão Robótica" indica uma evolução em direção a sistemas capazes de melhorar continuamente suas capacidades através da experiência. Esta característica é fundamental para aplicações robóticas que devem operar em ambientes dinâmicos e não estruturados.

Diversificação de Aplicações: A presença de métodos específicos para "Robôs Aéreos" exemplifica a expansão da visão robótica para novos domínios de aplicação. Esta diversificação reflete a crescente maturidade das técnicas básicas e sua capacidade de adaptação a diferentes plataformas e ambientes operacionais.

Considerações sobre Sistemas Embarcados: Um aspecto notável na análise do estado da arte é o uso frequente de sensores RGB-D, como o Microsoft Kinect, em sistemas embarcados robóticos. Dispositivos como o módulo ESP32 podem ser integrados com estes sensores para criar sistemas robóticos autônomos capazes de percepção tridimensional, demonstrando a viabilidade de implementações compactas e eficientes.

Tabela 2.2: Comparação entre Métodos de Visão Robótica

Categoria	Técnica e Método	Utilidade e Finalidades	Vantagens	Estratégias	Campos de Aplicação
Visão Robótica	Sensor RGB-D	Percepção tridimensional do ambiente	Sensor versátil com profundidade	com	Robótica, mapeamento, navegação
Visão Robótica	Algoritmos de Suporte	Tradução de imagens em sinais de controle	Adaptação a diferentes cenários		Orientação e controle de robôs
Visão para Robôs	Deteção de Objetos	Identificação de objetos para interação	Precisão na detecção, velocidade		Manipulação de objetos, segurança
Visão para Robôs	Aprendizado Adaptativo	Melhoria contínua da capacidade	Aprendizado contínuo, adaptação		Tarefas robóticas em evolução
Robôs Aéreos	Visão Computacional Aérea	Navegação autônoma em ambientes aéreos	Sensoriamento aéreo, mapeamento		Exploração, monitoramento aéreo
Processamento Multimodal	Fusão de Dados Sensoriais	Tomada de decisões baseada em múltiplas fontes	Robustez através de redundância		Robótica autônoma, interação humana
Deteção Avançada	Processamento Visual Inteligente	Interpretação contextual de cenas	Compreensão semântica avançada		Robótica colaborativa, assistência

2.5.3 Controle de Robôs Manipuladores

Os robôs manipuladores representam uma categoria específica de sistemas robóticos projetados para tarefas de manipulação física, caracterizados por serem fisicamente fixos em suas posições operacionais (Russell e Norvig 2016). Estes sistemas são amplamente utilizados em aplicações industriais, logísticas e de montagem onde precisão e repetibilidade são fundamentais.

A análise da Tabela 2.3 revela três tendências significativas no desenvolvimento de sistemas de controle para manipuladores:

Interfaces Naturais de Controle: Observa-se uma tendência crescente em direção ao uso de gestos manuais e movimentos corporais para controle de robôs manipuladores. Esta abordagem torna a interação humano-robô mais intuitiva e reduz a necessidade de treinamento especializado para operadores.

Diversificação de Aplicações: Os casos de uso identificados abrangem desde aplicações industriais tradicionais até colaboração em ambientes domésticos e assistência em tarefas cotidianas, demonstrando a versatilidade crescente destes sistemas.

Integração Sensorial Avançada: A incorporação de informações visuais e táteis nos sistemas de controle permite operações mais precisas e adaptáveis, melhorando a capacidade dos robôs de interagir com ambientes não estruturados.

Tabela 2.3: Classificações e Descrição do Controle de Manipulação

Tipo de Controle	Aplicações Práticas	Fonte Referencial
Palma Aberta	Classificação de objetos em ambiente industrial	Arenas <i>et al.</i> (2017)
Comandos de Braço	Navegação visual em ambientes desconhecidos	Choudhary e Chethan (2015)
Rotação do Braço	Estimação de pose para rastreamento de pessoas	Chen (2011); Song <i>et al.</i> (2017)
Controle de Garra	Controle robótico em linhas de montagem; Cirurgia colaborativa	Fareed <i>et al.</i> (2015); (Lima <i>et al.</i> 2019); Martin e Moutarde (2019)
Posicionamento de Objetos	Posicionamento preciso usando reconhecimento gestual	Arenas <i>et al.</i> (2017)
Retorno à Posição Base	Controle de posição para manipulação 3D	Arenas <i>et al.</i> (2017)
Continua		

Tabela 2.3 – *Classificações do Controle de Manipulação (continuação)*

Tipo de Controle	Aplicações Práticas	Fonte Referencial
Seleção por Apon- tamento	Rastreamento de alta veloci- dade usando filtros kerneliza- dos	Fontalis et al. (2023) , (Fareed et al. 2015), Luo et al. (2019)
Controle Multimo- dal	Rastreamento de objetos para aplicações visuais	Lima et al. (2019) ; Martin e Moutarde (2019)
Ações Colaborati- vas	Controle baseado em instru- ções naturais visuais e táteis	Quintero et al. (2015)
Manipulação Ges- tual	Interação multimodal em am- bientes de visão computacio- nal	Moh et al. (2019) ; Quintero et al. (2015) ; Valle et al. (2019)
Manipulação Coordena- da	Captação e transferência de objetos entre humano e robô	Ikai et al. (2016)

2.5.4 Reconhecimento de Gestos Baseado em Esqueletos

No contexto do reconhecimento de gestos manuais, os "esqueletos" referem-se a representações estruturais simplificadas que capturam a configuração espacial das mãos e dedos. Estas representações descrevem graficamente a posição relativa e orientação dos elementos anatômicos relevantes, incluindo articulações e segmentos das mãos.

A análise comparativa dos métodos baseados em esqueletos revela uma evolução nas abordagens técnicas, cada uma com características específicas, tais como:

Abordagens baseadas em Descritores Manuais: De Smedt *et al.* (2016) propõem o uso de descritores artesanais combinados com codificação do vetor de Fisher e máquinas de vetor de suporte (SVM) para classificação. Esta abordagem evidencia eficácia na representação geométrica das mãos, mas apresenta limitações na captura de padrões temporais complexos.

Métodos de Aprendizado Temporal: Chen, Guo, Wang e Zhang (2017) exploram Redes Neurais Recorrentes (RNN) para aprender características de movimento, incluindo tanto movimento global quanto movimento específico dos dedos. O uso de RNN bidirecionais evidencia eficácia na captura de dinâmicas temporais, sendo adequado para reconhecimento contínuo de gestos, embora requiera volumes substanciais de dados para treinamento.

Representação 3D Avançada: Weng *et al.* (2018) desenvolvem uma Rede Convolutacional Transversal de Pose Deformável para representação 3D da mão, utilizando LSTM para reconhecimento gestual. Esta abordagem evidencia robustez contra ruído nas articulações e eficácia na representação de poses tridimensionais, apesar da complexidade computacional adicional.

Mecanismos de Atenção: Hou *et al.* (2018) apresentam Redes de Convolução Temporal Residual com Atenção Espacial-Temporal, focando em características relevantes para aplicações em tempo real. Esta abordagem requer grandes conjuntos de dados rotulados para treinamento efetivo.

Processamento Paralelo: Devineau *et al.* (2018) utilizam CNNs paralelas para articulações esqueléticas com MLP para classificação. A vantagem está no processamento paralelo de sequências de articulações, embora com compreensão limitada de dependências temporais.

Integração CNN-LSTM: Nunez *et al.* (2018) combinam CNN e LSTM em abordagem de aprendizado profundo, permitindo extração de recursos complexos adequada para conjuntos de dados em grande escala, requerendo recursos computacionais substanciais.

Abordagens baseadas em Grafos: Métodos recentes exploram representações baseadas em grafos com autoatenção para considerar relações espaciais entre articulações, sendo eficazes na captura de relações espaciais, embora com complexidade crescente para estruturas maiores.

Aprendizado de Variedades: Nguyen *et al.* (2019) propõem redes neurais baseadas em aprendizado de variedades SPD (Simétricas e Definidas Positivas), utilizando matrizes SPD para representação gestual que captura aspectos espaciais e temporais, requerendo recursos computacionais substanciais.

Arquiteturas de Dois Fluxos: [Liu et al. \(2020\)](#) desenvolvem redes de dois fluxos separando efetivamente postura e movimento, utilizando CNNs 3D e 2D para extração de recursos, resultando em arquitetura mais complexa para treinamento e implementação.

A diversidade de abordagens reflete a evolução contínua do campo, com cada método oferecendo vantagens específicas dependendo dos requisitos da aplicação. A escolha da abordagem adequada deve considerar fatores como disponibilidade de recursos computacionais, requisitos de tempo real e características específicas dos gestos a serem reconhecidos.

2.6 Resumo do Capítulo

Este capítulo apresentou uma fundamentação teórica abrangente sobre os principais conceitos que sustentam este trabalho. Inicialmente, foram discutidos os princípios da visão computacional e robótica, abordando desde os componentes básicos dos sistemas robóticos (sensores e atuadores) até as categorias de robôs (manipuladores, móveis e manipuladores móveis).

Em seguida, explorou-se o campo do reconhecimento de gestos, com ênfase nos gestos manuais, suas classificações (estáticos e dinâmicos) e aplicações em robótica manipuladora. Foram apresentados diversos métodos de controle gestual, incluindo comandos básicos (como palma aberta e gestos de pinça) e abordagens mais complexas (controle multimodal e ações colaborativas).

A seção sobre interação humano-máquina através de visão computacional detalhou as principais tecnologias de sensoriamento visual (câmeras RGB, sensores RGB-D) e técnicas de processamento (detecção de objetos, classificação de imagens e segmentação semântica), além de abordar a análise visual de comportamento humano.

Por fim, foi introduzida a linguagem SysML para modelagem de sistemas, com destaque para seus diagramas estruturais, comportamentais e de análise, bem como suas diferenças em relação à UML. O capítulo também incluiu um mapeamento sistemático da literatura, comparando métodos em visão robótica e abordagens para reconhecimento de gestos baseados em esqueletos.

Esta fundamentação teórica fornece as bases necessárias para o desenvolvimento do sistema proposto, integrando conceitos de visão computacional, reconhecimento de gestos, interação humano-máquina e modelagem de sistemas.

Capítulo 3

Trabalhos Correlatos

Este capítulo analisa trabalhos correlatos à pesquisa, selecionados por similaridade técnica e metodológica. Foram examinadas quatro abordagens com convergência paradigmática, destacando-se: (1) uso de visão computacional para reconhecimento gestual; (2) aplicação em teleoperação robótica; (3) interfaces homem-máquina naturais; e (4) soluções em tempo real. A comparação identifica lacunas, valida a originalidade da proposta e estabelece parâmetros para justificar a escolha de uma orientação metodológica, fortalecendo o embasamento teórico, posicionando a contribuição no cenário científico atual.

3.1 Reconhecimento de Gestos através de Dispositivos de Automação Residencial

O trabalho desenvolvido por [Liu et al. \(2021\)](#) propõe o reconhecimento de gestos em tempo real para controle de braço robótico em ambientes de automação residencial, apresentando significativa convergência com a abordagem metodológica aqui proposta.

3.1.1 Fundamentação Teórica e Metodológica

A pesquisa considera a detecção de gestos manuais como uma linguagem simbólica não vocal aplicável à interação homem-computador. O estudo utiliza a arquitetura Inception v3, uma Rede Neural Convolutacional (CNN) desenvolvida pela equipe do Google Research, aplicando técnicas de transferência de aprendizado para otimizar o treinamento.

A arquitetura Inception v3 representa uma evolução das versões anteriores (Inception v1 e v2/GoogLeNet), tendo sido projetada especificamente para o *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC)

de 2015. Esta arquitetura evidencia uma capacidade superior para classificação de imagens em milhares de categorias, característica essencial para aplicações de reconhecimento gestual.

3.1.2 Implementação e Características Técnicas

O sistema implementado realiza a classificação de gestos capturados por câmera em seis categorias distintas, cada uma com probabilidades correspondentes. A integração com sistemas de automação residencial e controle de braço robótico simplifica significativamente as operações domésticas e de manipulação robótica.

Principais vantagens identificadas:

- Facilidade de uso superior aos comandos manuais convencionais;
- Capacidade de automatização de ações domésticas complexas;
- Integração eficiente com sistemas robóticos;
- Processamento em tempo real com alta precisão.

3.1.3 Metodologia e Resultados

A metodologia emprega retreinamento do modelo Inception v3 com transferência de aprendizado, reduzindo significativamente os requisitos computacionais, de dados e tempo de processamento. A classificação gestual baseia-se em análise probabilística, garantindo robustez na identificação de padrões.

3.1.4 Análise do Estado da Arte Relacionado

O estudo referencia múltiplas abordagens para reconhecimento gestual, incluindo redes neurais profundas, redes convolucionais e métodos de atenção espacial-temporal. Estas abordagens variam em complexidade e eficácia para detecção gestual em ambientes humano-robóticos, destacando-se a importância da precisão e robustez em aplicações de tempo real.

A comparação com tecnologias como MediaPipe e FingerPose identifica métodos baseados em representação de dados de esqueleto, descritores de mão, Redes Neurais Recorrentes (RNN), CNNs, e técnicas de aprendizado profundo. Particular atenção é dada ao processamento de dados de esqueleto em redes neurais baseadas em grafos, como a Rede Convolutiva de Grafos Espaço-Temporais (ST-GCN).

3.1.5 Contribuições e Limitações

Contribuições principais:

- Demonstração da viabilidade de gestos manuais para automação residencial;
- Integração eficiente de CNN com sistemas robóticos;

- Validação experimental da transferência de aprendizado em reconhecimento gestual.

Limitações identificadas:

- Dependência de condições controladas de iluminação;
- Limitação a seis categorias gestuais predefinidas;
- Necessidade de calibração para diferentes usuários.

3.2 Dispositivo Virtual para Aprendizado de Habilidades com Assistência Robótica

A pesquisa conduzida por [Luo *et al.* \(2023\)](#) propõe um dispositivo auxiliar virtual para treinamento humano utilizando aprendizado robótico via teleoperação, focalizando no desenvolvimento de sistemas de teleoperação para tarefas semiautomáticas em ambientes remotos de alta complexidade.

3.2.1 Arquitetura do Sistema Proposto

O sistema baseia-se em visão computacional e aprendizado robótico para capturar habilidades de manipulação de operadores especializados, gerando trajetórias especializadas para treinamento de operadores inexperientes. Um dispositivo virtual (*fixture*) baseado em visão fornece orientações de força, melhorando significativamente a precisão do controle.

3.2.2 Metodologia de Aprendizado e Transferência

A metodologia envolve:

1. **Captura de habilidades especializadas:** Registro de trajetórias e padrões de manipulação de operadores experientes;
2. **Geração de trajetórias de referência:** Criação de modelos de movimento otimizados para treinamento;
3. **Orientação de força virtual:** Implementação de feedback háptico baseado em visão computacional;
4. **Transferência de conhecimento:** Aplicação de técnicas de aprendizado para acelerar o treinamento de novatos.

3.2.3 Vantagens e Aplicações

Principais vantagens:

- Capacidade de treinar rapidamente operadores inexperientes para nível especializado;
- Melhoria significativa na precisão de manipulação robótica;
- Redução do tempo de aprendizado para tarefas complexas;
- Validação experimental comprovada da eficácia do sistema.

Aplicações práticas:

- Treinamento de teleoperação em ambientes perigosos;
- Trabalhos remotos de alta precisão;
- Cirurgias robóticas assistidas;
- Manipulação de materiais em ambientes hostis.

3.2.4 Características Técnicas Relevantes

O sistema integra informações de interação multimodal para otimizar o desempenho do controle robótico. A simplicidade e conveniência da interface tornam o sistema aplicável tanto para treinamento quanto para operações produtivas em trabalhos remotos.

A abordagem baseada em visão elimina a necessidade de sensores hápticos complexos, reduzindo custos e aumentando a robustez do sistema em ambientes industriais diversos.

3.2.5 Relevância para a Pesquisa Proposta

Esta pesquisa oferece contribuições significativas para o entendimento de como sistemas de visão computacional podem ser aplicados em teleoperação robótica. Os conceitos de transferência de habilidades e orientação virtual são particularmente relevantes para o desenvolvimento de interfaces gestuais intuitivas.

3.3 Braço Robótico Controlado por Reconhecimento de Gestos Manuais

O trabalho desenvolvido por [Varshini *et al.* \(2021\)](#) apresenta uma abordagem focada no reconhecimento em tempo real de gestos manuais para controle de braço robótico e automação residencial, demonstrando convergência metodológica significativa com a pesquisa proposta.

3.3.1 Fundamentação Teórica

A pesquisa estabelece gestos manuais como linguagem simbólica não vocal eficiente para comunicação homem-computador. A aplicação de visão computacional para detecção gestual resulta em interfaces naturais na área de interação humano-computador, eliminando barreiras tradicionais de operação.

3.3.2 Arquitetura Neural e Transferência de Aprendizado

O sistema utiliza a arquitetura Inception v3 como base neural, aplicando técnicas de transferência de aprendizado para otimizar:

- **Redução de recursos computacionais:** Menor demanda de processamento para treinamento;
- **Minimização de dados de treino:** Aproveitamento de conhecimento pré-existente;
- **Otimização temporal:** Redução significativa do tempo de treinamento;
- **Melhoria da generalização:** Maior robustez em cenários diversos.

3.3.3 Sistema de Classificação e Controle

O sistema classifica gestos capturados por câmera em seis categorias distintas, cada uma associada a uma probabilidade de confiança. Esta abordagem probabilística favorece a robustez na tomada de decisões e permite implementação de limiares de confiança adaptativos.

Fluxo de processamento:

1. Captura de imagem via câmera;
2. Pré-processamento e normalização;
3. Classificação neural com Inception v3;
4. Análise probabilística dos resultados;
5. Transmissão de comandos para microcontrolador;
6. Execução de ações em dispositivos ou braço robótico.

3.3.4 Vantagens Operacionais

Simplicidade de uso:

- Interface intuitiva baseada em gestos naturais;
- Eliminação de comandos manuais complexos;
- Redução da curva de aprendizado para usuários;
- Operação sem contato físico com dispositivos.

Eficiência sistêmica:

- Automação residencial otimizada;
- Controle robótico simplificado;
- Redução de erros operacionais;
- Melhoria na acessibilidade para usuários com limitações físicas.

3.3.5 Aplicações e Resultados

O sistema evidencia aplicabilidade em:

- **Automação residencial:** Controle de dispositivos domésticos inteligentes;
- **Manipulação robótica:** Operação de braços robóticos para tarefas diversas;
- **Assistência domiciliar:** Suporte para pessoas com mobilidade reduzida;
- **Educação tecnológica:** Plataforma para ensino de robótica e IA.

3.3.6 Contribuições Metodológicas

A pesquisa contribui significativamente para o estado da arte ao:

- Demonstrar a viabilidade de transferência de aprendizado em reconhecimento gestual;
- Validar a integração de técnicas de aprendizado profundo com sistemas robóticos;
- Estabelecer protocolos eficientes de comunicação entre visão computacional e controle robótico;
- Provar a eficácia de interfaces gestuais para automação residencial.

3.4 Sistema Interativo de Robô Controlado Remotamente por Gestos Manuais para Pessoas com Deficiências

A pesquisa desenvolvida por [Karim *et al.* \(2023\)](#) apresenta um Sistema Interativo Humano-Robô inovador, permitindo controle robótico remoto via gestos manuais através de framework IoT baseado em smartphone, especificamente projetado para auxiliar pessoas com deficiências vocais e auditivas.

3.4.1 Arquitetura do Sistema IoT

O sistema implementa uma arquitetura distribuída composta por:

1. **Camada de captura:** Aplicativo móvel para registro gestual;
2. **Camada de comunicação:** Transmissão segura de dados via internet;
3. **Camada de processamento:** Servidor centralizado para reconhecimento gestual;
4. **Camada de controle:** Sistema robótico para execução de tarefas;
5. **Camada de feedback:** Retorno visual e tátil para o usuário.

3.4.2 Fluxo Operacional

Processo de interação:

1. **Captura gestual:** Usuário executa gestos diante do smartphone;
2. **Registro e pré-processamento:** Aplicativo móvel processa e envia dados;
3. **Reconhecimento centralizado:** Servidor analisa gestos e gera comandos;
4. **Transmissão de comandos:** Envio de instruções para sistema robótico;
5. **Execução robótica:** Robô realiza tarefa correspondente ao gesto;
6. **Confirmação:** Sistema retorna status de execução ao usuário.

3.4.3 Capacidades Robóticas Implementadas

O sistema robótico foi configurado para realizar operações de **pegar e colocar** (*pick-and-place*) baseadas em comandos gestuais. Esta funcionalidade fundamental permite:

- Manipulação de objetos diversos;
- Assistência em tarefas domésticas;
- Suporte para pessoas com limitações físicas;
- Automação de processos repetitivos.

3.4.4 Métricas de Performance

A validação experimental demonstrou resultados excepcionais:

Precisão de reconhecimento gestual: 100%

- Classificação correta de todos os gestos testados;
- Robustez em diferentes condições de iluminação;
- Estabilidade com múltiplos usuários;
- Consistência em sessões prolongadas de uso.

Precisão de manipulação robótica: 100%

- Execução bem-sucedida de todas as tarefas de captura;
- Posicionamento preciso de objetos;
- Adaptação a objetos de diferentes tamanhos e formas;
- Manutenção da precisão em operações consecutivas.

Tempo médio de execução: 78 segundos

- Inclui todo o ciclo: captura → processamento → execução;
- Tempo competitivo para aplicações assistivas;
- Margem para otimização em versões futuras;
- Adequado para tarefas não críticas temporalmente.

3.4.5 Inovações Tecnológicas

Acessibilidade aprimorada:

- Interface gestual elimina barreiras vocais e auditivas;
- Operação remota amplia alcance de ação;
- Framework móvel permite portabilidade;
- Baixo custo de implementação.

Robustez do sistema:

- Arquitetura distribuída oferece maior confiabilidade;
- Processamento centralizado otimiza recursos;
- Comunicação IoT permite escalabilidade;
- Interface intuitiva reduz curva de aprendizado.

3.4.6 Aplicações e Impacto Social

Aplicações diretas:

- Assistência para pessoas com deficiências motoras;
- Suporte domiciliar para idosos;
- Reabilitação robótica;
- Automação residencial acessível.

Impacto social:

- Promoção da inclusão digital;
- Melhoria da qualidade de vida;
- Redução da dependência de cuidadores;
- Democratização do acesso à tecnologia robótica.

3.4.7 Relevância para a Pesquisa Proposta

Esta pesquisa oferece contribuições valiosas para o desenvolvimento de sistemas gestuais robóticos:

- **Validação de viabilidade:** Evidencia a eficácia de sistemas gestuais em aplicações reais;
- **Métricas de benchmark:** Estabelece padrões de performance para comparação;
- **Arquitetura de referência:** Fornece modelo para sistemas distribuídos;
- **Foco em acessibilidade:** Inspira desenvolvimento de soluções inclusivas.

3.5 Análise Comparativa e Posicionamento da Pesquisa

A análise dos trabalhos correlatos revela um panorama diversificado de abordagens para reconhecimento gestual e controle robótico, cada uma contribuindo com perspectivas únicas para o estado da arte.

3.5.1 Matriz Comparativa de Abordagens

Critério	Liu et al.	Luo et al.	Varshini et al.	Karim et al.
Arquitetura Neural	Inception v3	Visão + ML	Inception v3	CNN personalizada
Aplicação Principal	Automação residencial	Teleoperação	Braço robótico	Assistência a deficientes
Tempo Real	Sim	Sim	Sim	Sim
Precisão Reportada	Não especificada	Experimental	Não especificada	100%
Gestos Suportados	6 categorias	Trajetórias	6 categorias	Múltiplos
Inovação Principal	Transfer learning	Dispositivo virtual	CNN otimizada	Framework IoT

Tabela 3.1: Matriz comparativa dos trabalhos correlatos analisados

3.5.2 Lacunas Identificadas

A análise dos trabalhos correlatos revela lacunas importantes que justificam a originalidade da presente pesquisa:

- **Limitação a gestos estáticos:** Maioria dos trabalhos foca em gestos discretos, não explorando linguagens gestuais complexas como LIBRAS;
- **Dependência de hardware específico:** Sistemas frequentemente requerem câmeras especializadas ou sensores adicionais;
- **Escalabilidade limitada:** Dificuldade de expansão para novos gestos sem retreinamento extensivo;
- **Falta de padronização:** Ausência de protocolos comuns para integração com diferentes sistemas robóticos.

3.5.3 Contribuições Distintivas da Pesquisa Proposta

A presente pesquisa diferencia-se dos trabalhos correlatos através de:

- **Foco em LIBRAS:** Aplicação específica para linguagem de sinais brasileira;
- **Arquitetura híbrida:** Combinação de MediaPipe com Random Forest;

- **Sistema embarcado:** Implementação em ESP32 para IoT;
- **Validação sistemática:** Comparação rigorosa entre múltiplas abordagens;
- **Aplicação social:** Contribuição para inclusão e acessibilidade.

3.5.4 Posicionamento no Estado da Arte

Esta pesquisa posiciona-se como uma evolução natural dos trabalhos analisados, incorporando suas principais contribuições enquanto endereça limitações identificadas. A abordagem proposta representa um avanço significativo ao:

- Combinar eficiência computacional com alta precisão;
- Focar em aplicação social específica (LIBRAS);
- Implementar arquitetura escalável e de baixo custo;
- Validar metodologia através de comparação sistemática.

O conjunto de trabalhos correlatos analisados estabelece um embasamento sólido para a contribuição científica proposta, validando a relevância e originalidade da pesquisa desenvolvida.

3.6 Resumo do Capítulo

Este capítulo apresentou uma análise abrangente de quatro trabalhos fundamentais que colaboram com a pesquisa proposta, evidenciando tanto aspectos positivos quanto lacunas e limitações peculiares a cada proposta. Por conseguinte, as soluções analisadas até aqui, tanto sumarizam quanto descrevem certos elementos científicos significativos e substancialmente destacáveis no estado da arte, tais como:

- **Excelência técnica:** Todos os trabalhos demonstram arquiteturas robustas (como Inception v3 e CNNs personalizadas) com comprovada eficácia em reconhecimento gestual, oferecendo benchmarks de performance relevantes;
- **Inovação aplicada:** As pesquisas apresentam soluções práticas para problemas reais, desde automação residencial até assistência a pessoas com deficiências, validando o potencial transformador da tecnologia;
- **Eficiência computacional:** As técnicas de transferência de aprendizado e otimização neural desenvolvidas pelos correlatos estabelecem paradigmas valiosos para implementação em sistemas embarcados;
- **Integração sistêmica:** Os trabalhos demonstram casos de sucesso na conexão entre visão computacional, processamento em tempo real e controle robótico, superando desafios técnicos significativos.

A análise revelou ainda como esta pesquisa se beneficia das conquistas anteriores enquanto avança o estado da arte em três dimensões principais:

1. **Especificidade aplicada:** Ao focar na interpretação de LIBRAS, a proposta amplia o escopo das soluções existentes, tradicionalmente limitadas a gestos discretos;
2. **Acessibilidade ampliada:** A combinação de MediaPipe com Random Forest e implementação em ESP32 supera limitações de hardware especializado, tornando a solução mais versátil e acessível em termos de implementação;
3. **Validação rigorosa:** A metodologia comparativa adotada nesta pesquisa oferece um quadro de avaliação mais abrangente que os estudos individuais analisados.

Capítulo 4

Método Proposto

*Este capítulo apresenta a arquitetura do método proposto, considerando os aspectos mais relevantes de seu desenvolvimento, bem como a exposição de sua concepção e compreensão lógica por meio de diagramas e linguagens descritivas como SysML e UML. São apresentadas as opções adotadas para a arquitetura e metodologia do sistema de teleoperação da mão robótica via reconhecimento gestual por visão computacional. A solução combina hardware (câmera, módulo **ESP32-WROOM**, servomotores) e software (processamento em PC, comunicação cliente-servidor). O uso da modelagem empregando a SysML, destacou o Diagrama de Blocos Internos (IBD) para representar a estrutura e interconexões dos módulos. O fluxo operacional envolve captura de imagens, processamento e transmissão em tempo real para reprodução dos movimentos. Delinearam-se ainda as estratégias para reconhecimento das vogais em LIBRAS, comparando três abordagens: (1) distâncias euclidianas entre landmarks (MediaPipe Hands); (2) classificação de flexão digital (Fingerpose); e (3) integração de 'landmarks' com Random Forest (Scikit-Learn). Conclusivamente, a avaliação levou em consideração elementos como acurácia, robustez e controle de latência, incluindo extensões de testes envolvendo outros algoritmos de Machine Learning objetivando otimização.*

4.1 Arquitetura do Sistema

O método proposto baseia-se em um sistema embarcado de teleoperação e computação visual, composto pelos seguintes componentes:

- Dispositivo de captura de imagens (webcam) para aquisição do fluxo de vídeo focado na mão humana;
- Computador pessoal conectado ao dispositivo de captura;
- Software de reconhecimento gestual implementado no computador;
- Camada cliente de uma solução web cliente-servidor para processamento dos gestos;

- Servidor web backend para gerenciamento da recepção e tratamento de imagens;
- Módulo servidor embarcado ESP32 WROOM integrado via protocolo Wi-Fi 802.11;
- Módulo step down para fornecimento de tensão auxiliar aos servomotores;
- Servomotores 9G SG90 para controle de movimentos e flexão dos dedos da mão robótica;
- Mecanismo de prototipação que simula o movimento mecânico do dedo robótico conectado ao servomotor.

A partir desta arquitetura eletrônica e eletromecânica, o sistema foi modelado utilizando a linguagem SysML e seus respectivos artefatos.

A pesquisa investigou três abordagens complementares para a captura e o mapeamento dos movimentos dos dedos, com o objetivo de viabilizar o processamento de imagem e a transmissão de dados em tempo real ao sistema de controle do servomotor. A primeira técnica consistiu na medição das distâncias euclidianas entre pontos de referência (landmarks) dos dedos, fornecendo um método direto de inferência posicional. A segunda abordagem envolveu o mapeamento dos graus de flexão digital por meio da biblioteca Hand Pose, permitindo a identificação do nível de curvatura de cada dedo. Por fim, a terceira estratégia incorporou técnicas de aprendizado de máquina, combinando a biblioteca MediaPipe com um classificador Random Forest implementado com o pacote Scikit-Learn, de modo a aprimorar a acurácia da identificação gestual.

4.2 Metodologia do Sistema Proposto

O processo metodológico adotado foi estruturado em cinco etapas sequenciais e integradas. Inicialmente, o sistema realiza a captura do movimento gestual por meio de uma webcam, como ilustrado na Figura 4.1. Em seguida, ocorre a identificação, o mapeamento e a quantificação do movimento capturado, etapa essencial para a preparação dos dados para transmissão, conforme representado na Figura 4.2. Na terceira etapa, os dados processados são transmitidos pela rede até o servidor de gerenciamento de imagens, como mostra a Figura 4.3. Posteriormente, os dados são recebidos e encaminhados ao servidor IoT responsável pelo controle dos atuadores físicos (Figura 4.4). Finalmente, ocorre o acionamento do mecanismo robótico que reproduz, de forma biomimética, o movimento gestual capturado, conforme ilustrado na Figura 4.5.

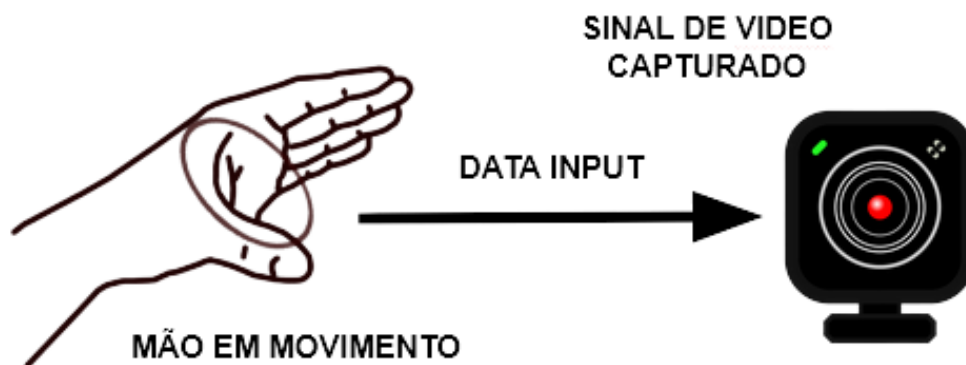


Figura 4.1: Captura de dados por meio do registro de movimentos gestuais

4.2.0.1 Etapa 1: Captura do Movimento Gestual

Na primeira etapa (Figura 4.1), o sistema realiza a aquisição do sinal visual. Uma mão gesticula diante da webcam, e o sensor passivo de imagem captura o fluxo de vídeo para processamento posterior.

Esta etapa estabelece a base de entrada do sistema ao converter o movimento físico da mão em dados digitais através do sensor de vídeo. A webcam atua como ponto de coleta primária dos sinais visuais, sendo responsável por registrar, em tempo real, cada variação posicional dos dedos e articulações da mão. A fidelidade e a taxa de quadros da câmera impactam diretamente a qualidade da detecção, uma vez que movimentos rápidos ou pouco iluminados podem comprometer a acurácia da segmentação gestual nas etapas seguintes.

Além de sua função como sensor visual, a webcam tem papel fundamental na sincronização temporal dos dados. O fluxo contínuo de imagens gerado é responsável por manter a coerência dos gestos capturados, permitindo que o sistema reconheça padrões dinâmicos e transições sutis entre diferentes configurações manuais. Isso é crucial para garantir que o sistema de reconhecimento seja responsivo, minimizando atrasos na identificação dos comandos e otimizando a interação entre o usuário e o sistema robótico assistivo.

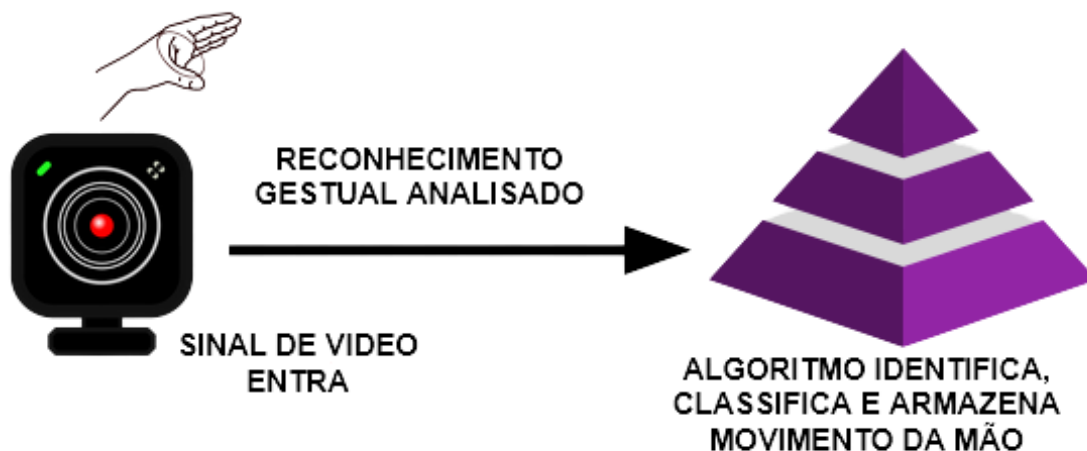


Figura 4.2: Processamento do sinal de vídeo por algoritmo classificador de movimento gestual

4.2.0.2 Etapa 2: Processamento e Interpretação dos Dados

Conforme ilustrado na Figura 4.2, o fluxo de imagens proveniente da webcam é processado por um algoritmo classificador previamente treinado, que identifica, interpreta e mapeia o movimento gestual.

Nesta etapa, os quadros extraídos do fluxo de vídeo passam por uma série de transformações computacionais que visam a extração de características relevantes, tais como posições angulares dos dedos, distâncias entre landmarks e variações de curvatura. Tais características são então normalizadas e organizadas em vetores de entrada compatíveis com os modelos classificadores, que operam com base em aprendizado supervisionado. O sistema aplica filtros temporais e espaciais para minimizar ruídos visuais e garantir uma maior robustez na segmentação do gesto, mesmo sob condições de iluminação adversa ou com oclusões parciais da mão.

A utilização de um algoritmo classificador permite que o sistema reconheça diferentes configurações gestuais a partir de padrões aprendidos durante a fase de treinamento. Isso viabiliza a associação direta entre os dados visuais capturados e comandos interpretáveis para o sistema de controle. A eficiência desta etapa é determinante para a acurácia do sistema como um todo, uma vez que eventuais falhas na interpretação gestual podem comprometer a resposta motora do atuador robótico. Dessa forma, o processamento em tempo real e a capacidade do classificador de generalizar os gestos são elementos centrais para o sucesso da aplicação em contextos interativos.

4.2.0.3 Etapa 3: Transmissão de Dados via Rede

A terceira etapa (Figura 4.3) descreve a transmissão dos dados gestuais processados para o microcontrolador IoT, por meio de uma rede local estabelecida via ponto de acesso Wi-Fi físico baseado no protocolo IEEE 802.11. Nesta configuração, a comunicação ocorre diretamente entre o dispositivo que executa a aplicação de visão computacional e o microprocessador embarcado, que hospeda um servidor HTTP leve responsável por receber os comandos gestuais e acionar os elementos mecânicos da mão robótica.



Figura 4.3: Transmissão de dados processados pelo algoritmo de reconhecimento gestual

Ao final do processamento visual, os dados extraídos do gesto reconhecido são codificados na forma de parâmetros inseridos diretamente na URL de uma requisição HTTP do tipo GET. Essa requisição é enviada através da rede local para o endereço IP do microcontrolador, que interpreta os valores recebidos a partir da query string. Cada parâmetro representa valores numéricos que indicam as posições desejadas dos servomotores correspondentes aos diferentes dedos do protótipo robótico.

O servidor HTTP embarcado no microprocessador é responsável por processar os parâmetros da URL, traduzindo-os em comandos elétricos que acionam os servomotores com base nos ângulos especificados. Essa abordagem permite um controle direto, rápido e eficiente sobre os atuadores, favorecendo sistemas embarcados com restrições de recursos computacionais. Além disso, a simplicidade do protocolo adotado facilita a integração e a depuração durante o desenvolvimento e a validação funcional do sistema.

Embora esta implementação não incorpore mecanismos de autenticação, validação de entrada ou criptografia, a arquitetura se mostra adequada para ambientes de prototipagem e testes locais. Em aplicações futuras, pode-se considerar a ampliação da robustez do sistema com o uso de protocolos seguros, autenticação por token e filtros de verificação para entrada de dados, de modo a garantir a integridade e a confiabilidade da comunicação em cenários mais sensíveis.

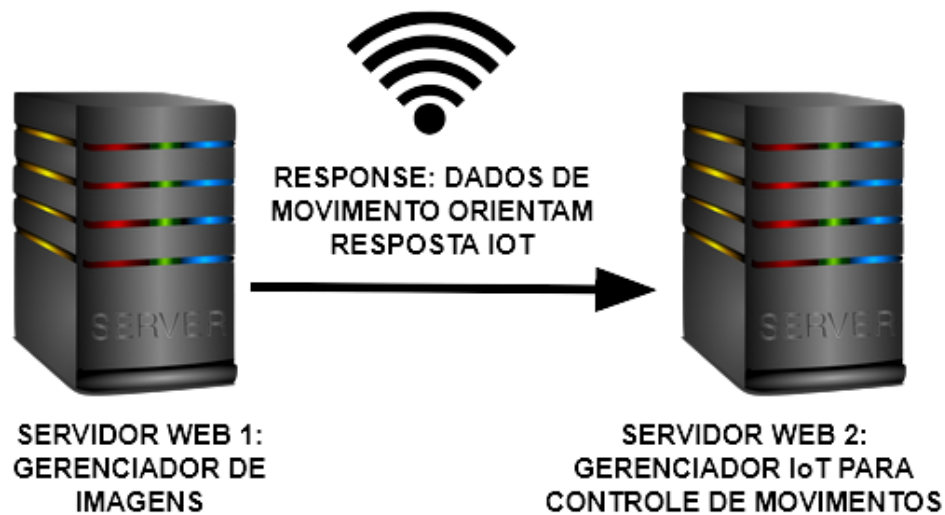


Figura 4.4: Recepção de dados de movimento por meio do sistema de captura

4.2.0.4 Etapa 4: Recepção e Decodificação dos Dados

Na Etapa 4 (Figura 4.4), dá-se início ao processo de recepção e decodificação dos dados gestuais pelo microprocessador IoT, que atua como servidor web embarcado. Após a transmissão via protocolo HTTP, os pacotes contendo os parâmetros codificados na query string são recebidos pela interface de rede Wi-Fi do microcontrolador, o qual deve garantir a integridade e a correta sincronização das requisições para assegurar a continuidade do controle gestual.

O servidor HTTP embarcado interpreta os parâmetros extraídos da URL da requisição, efetuando a decodificação dos valores numéricos que representam os ângulos ou posições desejadas para os servomotores. Essa etapa inclui a validação dos dados recebidos para evitar comandos inválidos ou fora da faixa operacional dos atuadores. Após a decodificação, o microprocessador converte os valores em sinais PWM (Pulse Width Modulation), adequados para o acionamento dos motores que controlam os movimentos dos dedos do protótipo robótico.

A eficiência e a precisão deste processo são fundamentais para que o sistema responda de forma rápida e coerente aos comandos capturados na fase inicial. A sincronização entre a recepção dos dados e a execução dos movimentos permite a reprodução fluida e biomimética dos gestos, minimizando atrasos perceptíveis pelo usuário. Além disso, esta arquitetura distribuída, com a lógica de controle concentrada no microprocessador IoT, possibilita maior autonomia e robustez operacional, reduzindo dependências externas e facilitando o escalonamento para múltiplos dispositivos atuadores.

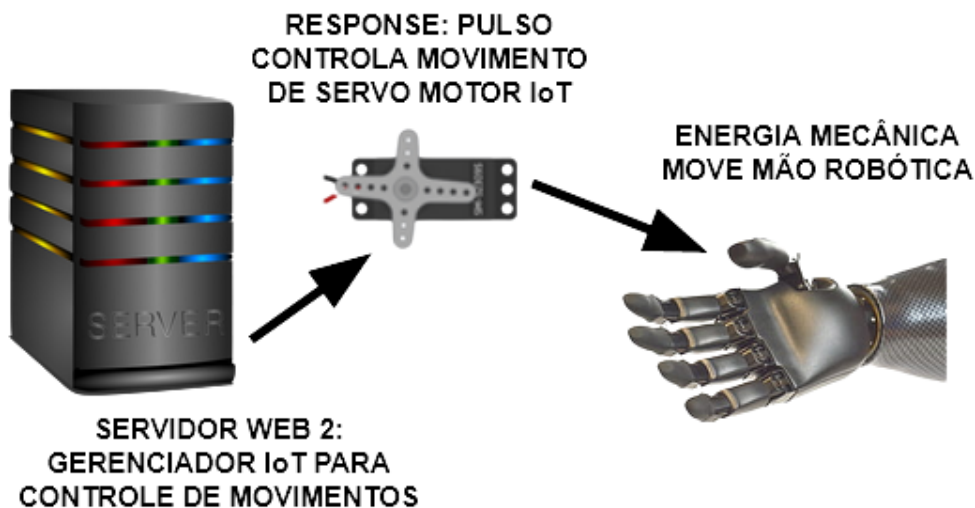


Figura 4.5: Execução do movimento mecânico da mão robótica comandado pelo servomotor

4.2.0.5 Etapa 5: Acionamento do Sistema Robótico

A Figura 4.5 ilustra a etapa final do processo, na qual ocorre a execução do movimento mecânico da mão robótica, controlado pelos servomotores em resposta aos comandos recebidos via comunicação HTTP.

Nesta etapa, os valores de ângulo para a flexão ou retração dos dedos são transmitidos ao microcontrolador por meio da query string da requisição HTTP, representando diretamente a posição desejada de cada servo motor responsável pelo movimento dos dedos. Devido à simplicidade dos servomotores empregados, que não dispõem de controle de torque nem de parametrização de velocidade e precisão, a movimentação se baseia unicamente na definição do ângulo final a ser alcançado pelo atuador.

O microcontrolador converte os valores angulares recebidos em sinais PWM padrão, acionando os servos para executar a movimentação correspondente. Essa abordagem direta e simplificada atende ao propósito experimental, permitindo a reprodução básica dos gestos capturados sem a necessidade de controle fino sobre características dinâmicas do movimento.

Assim, o acionamento da mão robótica representa a etapa final do ciclo de controle, na qual os dados processados são convertidos em movimentos físicos efetivos, demonstrando a viabilidade do sistema para experimentos de reconhecimento e reprodução gestual, mesmo com limitações inerentes ao hardware utilizado.

A imagem apresentada pela Figura 4.6 ilustra um sistema de controle de movimento para uma mão robótica, em que o segmento de reta A-B representa uma força de tração.

O protótipo buscado neste trabalho baseia-se primariamente em um mecanismo de alavanca. Nele, um conjunto de falanges funciona como uma alavanca de primeira classe ou, dependendo do ponto de aplicação da força, de terceira classe. O ponto de pivô, ou junta, da falange é o fulcro. A força aplicada pelo servomotor atua no sistema para gerar um torque, que é a grandeza responsável por causar a rotação.

O servomotor, representado na imagem, é o atuador principal, fornecendo a força necessária para movimentar a falange. Essa força é transmitida através de um cabo, corda ou haste, que se estende do servomotor até o ponto de aplicação na falange.

A flexão da falange ocorre quando o servomotor traciona o cabo, puxando o ponto A para baixo, causando a rotação da falange em torno da junta. A extensão é o movimento oposto, que pode ser causado por um segundo servomotor ou por um sistema de mola antagonista que puxa a falange para a posição de extensão quando a força de tração é liberada.

A mesma Figura 4.6 ainda apresenta um ângulo α entre a linha de ação da força de tração (representada pela linha tracejada) e o eixo da falange é crucial. A componente do vetor de força (segmento A-B) que realmente contribui para o torque é dada por $F \cdot \sin(\alpha)$, onde F é a magnitude da força de tração.

O torque (T) exercido sobre a junta da falange é calculado pela fórmula:

$$T = r \cdot F \cdot \sin(\alpha)$$

onde:

- T é o torque.
- r é a distância do ponto de rotação (a junta da falange) até o ponto de aplicação da força (ponto A).
- F é a magnitude da força de tração.
- α é o ângulo entre o vetor de força e o segmento de falange.

Concluindo o exemplo apresentado, a imagem indica um ângulo de 15° entre duas linhas pontilhadas, que representam ali a faixa de movimento ou o desvio angular ideal da força. O controle preciso do servomotor é fundamental para variar o ângulo de tração e a intensidade da força, permitindo o controle fino do movimento de flexão e extensão das falanges.

Em síntese: o segmento de reta A-B representa a força de tração aplicada por um servomotor. Essa força é decomposta em um vetor, cuja componente perpendicular ao braço de alavanca da falange gera o torque necessário para controlar o movimento rotacional de flexão e extensão.

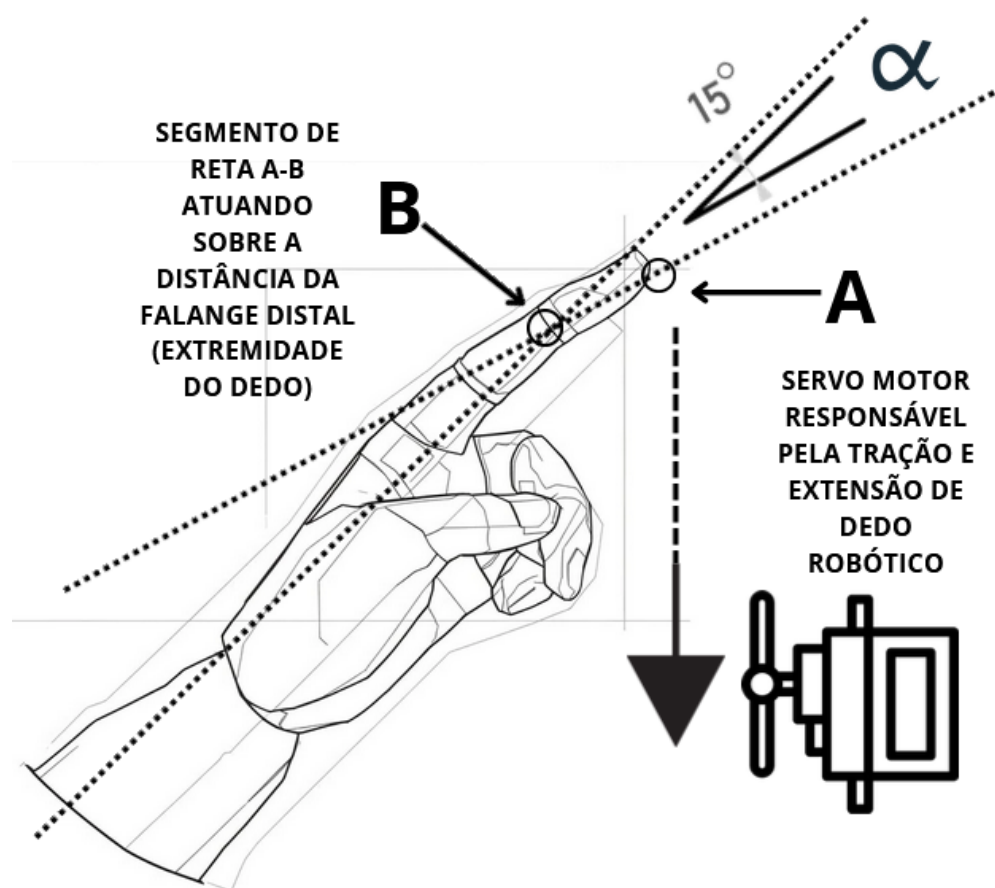


Figura 4.6: Sinal PWM ordenando ao mecanismo IoT movimento de rotação suave de 15° .

4.3 Modelagem SysML do Sistema Proposto

Para modelar o sistema eletrônico e eletromecânico proposto, foram utilizados artefatos específicos da linguagem SysML.

4.3.1 Diagrama de Atividade

O Diagrama de Atividade auxilia na compreensão do comportamento dinâmico do sistema, identificando interações entre diferentes componentes. Este recurso é particularmente útil para identificar estados do sistema e as transições entre eles.

A Figura 4.7 utiliza o Diagrama de Atividade para especificar o comportamento do sistema, focando no fluxo de controle e na transformação de entradas em saídas através de uma sequência de ações. O artefato mostra-se adequado para compreender e expressar o comportamento desejado do sistema proposto e as interações de seus componentes internos.

Para melhor visualização e percepção dos detalhes, o diagrama principal foi dividido em duas seções que contemplam o mesmo conteúdo da imagem original.

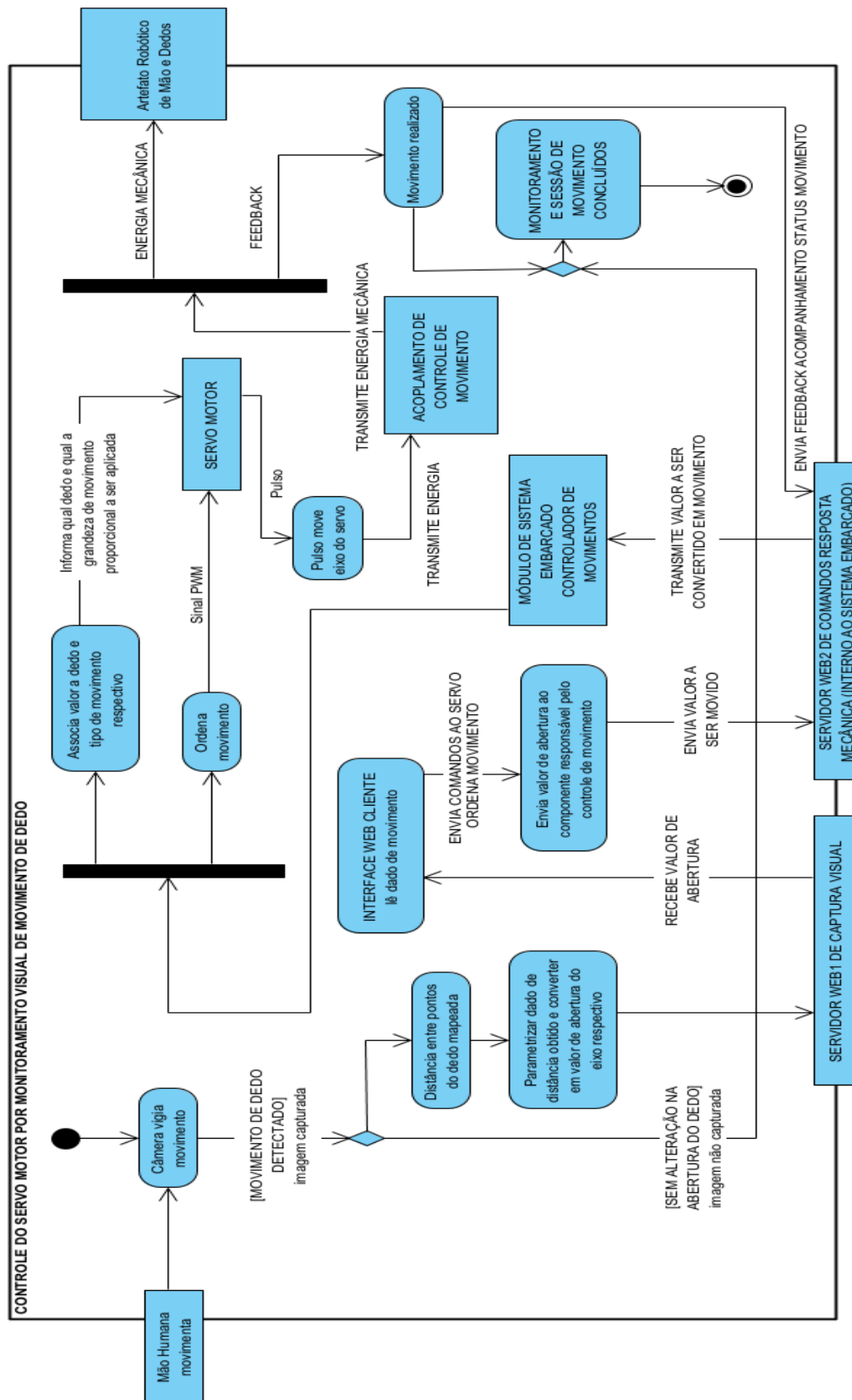


Figura 4.7: Diagrama de Atividade: Visão Ampla

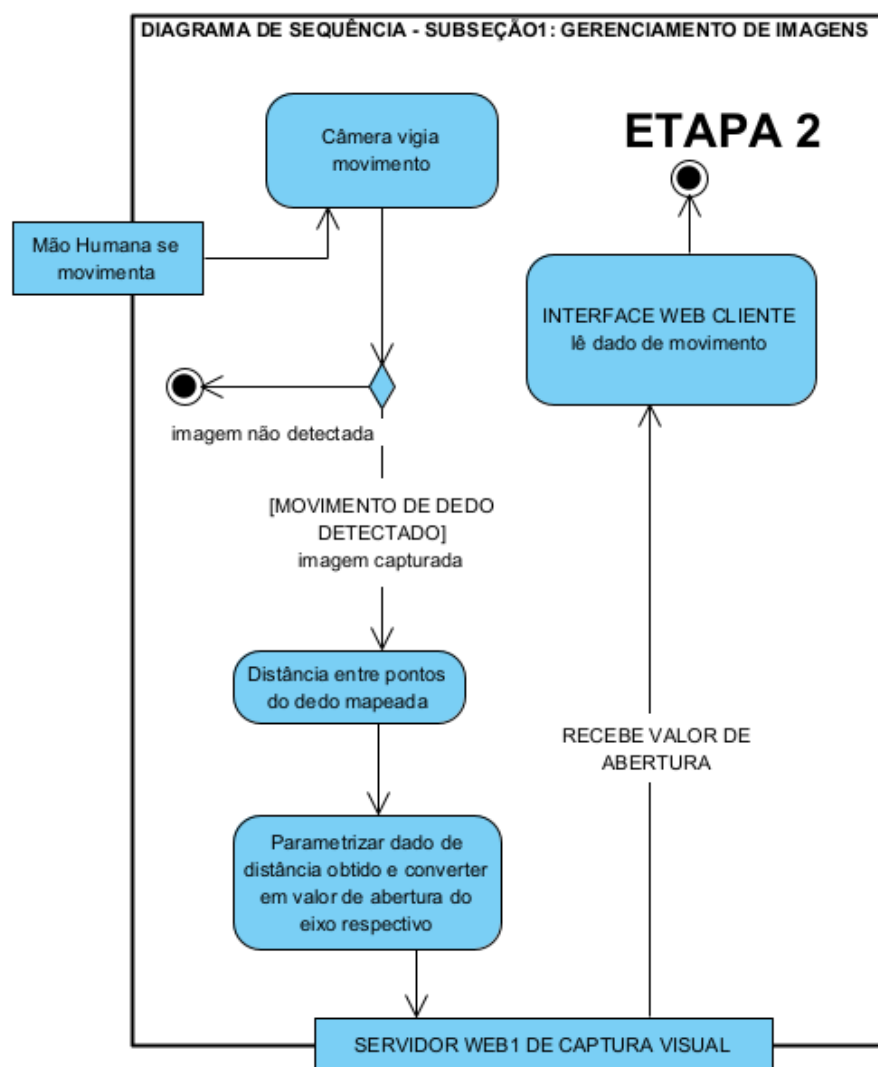


Figura 4.8: Diagrama de Atividade: Etapa 1 - Captura da Imagem

A Figura 4.8 apresenta a etapa inicial, referente à captura e tratamento da imagem.

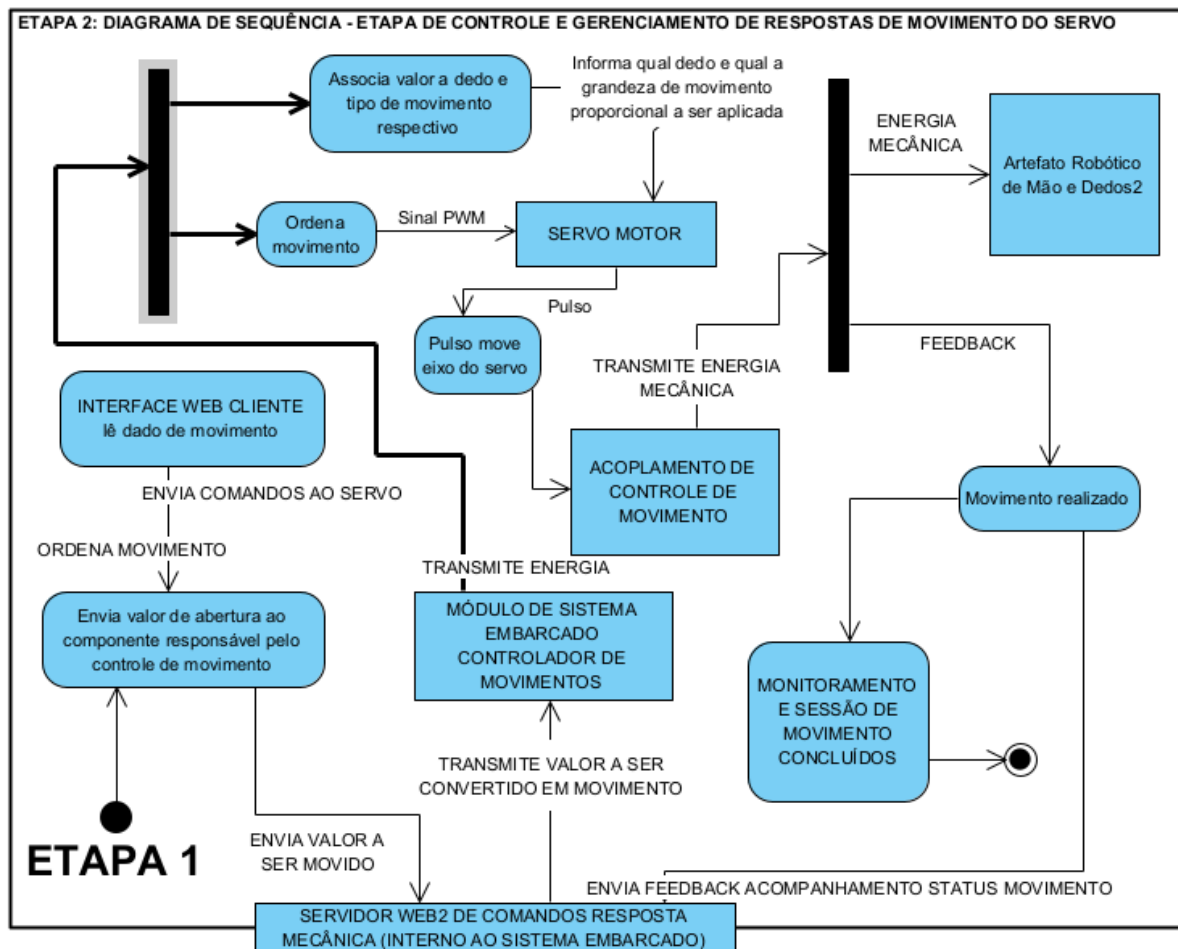


Figura 4.9: Diagrama de Atividade: Etapa 2 - Controle Mecânico

A etapa **conclusiva** é visualizada na Figura 4.9, ilustrando como o segundo servidor web gerencia o recebimento dos dados de imagem e os encaminha para o acionamento dos elementos do subsistema IoT, responsável pelo controle do servomotor e da mão robótica.

4.3.2 Diagrama de Definição de Blocos (*Block Definition Diagram* - BDD)

O diagrama BDD define os principais blocos ou elementos do sistema. Este artefato, visualizado na Figura 4.10, identifica os componentes principais como Webcam, Computador, Software de Reconhecimento Gestual, Camada Cliente, Servidor Web, Módulo ESP32, Módulo Step Down, Servomotor e Mecanismo de Prototipação. Cada componente é representado por um bloco com suas respectivas propriedades e interfaces.

Os principais blocos do sistema são:

- **Sistema Principal:** Bloco de nível mais alto que engloba todos os componentes;
- **Webcam:** Dispositivo passivo de captura da imagem da mão humana;
- **Computador:** Computador pessoal conectado à webcam via USB que executa o software de reconhecimento;
- **Software de Reconhecimento Gestual:** Software que processa as imagens para reconhecer gestos;
- **Camada Cliente:** Componente cliente da solução web que processa os gestos;
- **Servidor Web Backend:** Servidor que recebe dados de gestos e processa imagens em rede;
- **Módulo ESP32 WROOM:** Módulo embarcado que atua como cliente conectado via Wi-Fi;
- **Módulo Step Down:** Conversor de tensão para alimentação adequada dos servomotores;
- **Servomotor 9G SG90:** Motor responsável pelo controle e movimentos dos dedos da mão robô;
- **Mecanismo de Prototipação:** Mecanismo que simula o movimento do dedo robótico conectado ao servomotor.

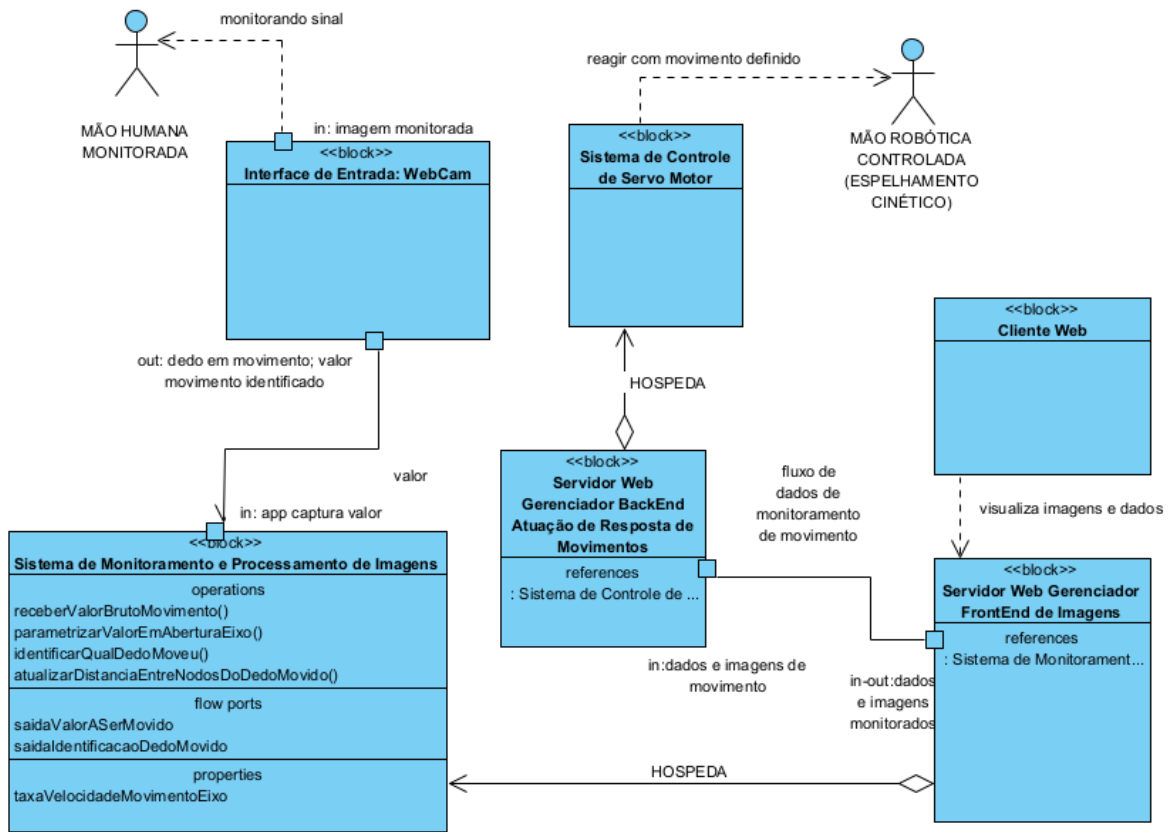


Figura 4.10: Diagrama de Definição de Blocos: Visão de Alta Abstração

4.3.3 Diagrama de Definição de Blocos - Visão de Alta Abstração

O diagrama BDD exibe elementos como blocos e tipos de valor, bem como as relações entre esses elementos. Fornece uma visão estruturada do sistema, identificando os principais blocos e suas relações, fundamental para a compreensão da arquitetura.

O BDD mostra relações entre blocos como associações, generalizações e dependências, representando a estrutura e conexões entre elementos do sistema. Permite representar relações hierárquicas, indicando a composição do sistema onde um bloco pode ser decomposto em sub-blocos.

A Figura 4.11 apresenta uma segunda visão BDD com menor nível de abstração, oferecendo uma perspectiva mais detalhada dos componentes que compõem o modelo proposto.

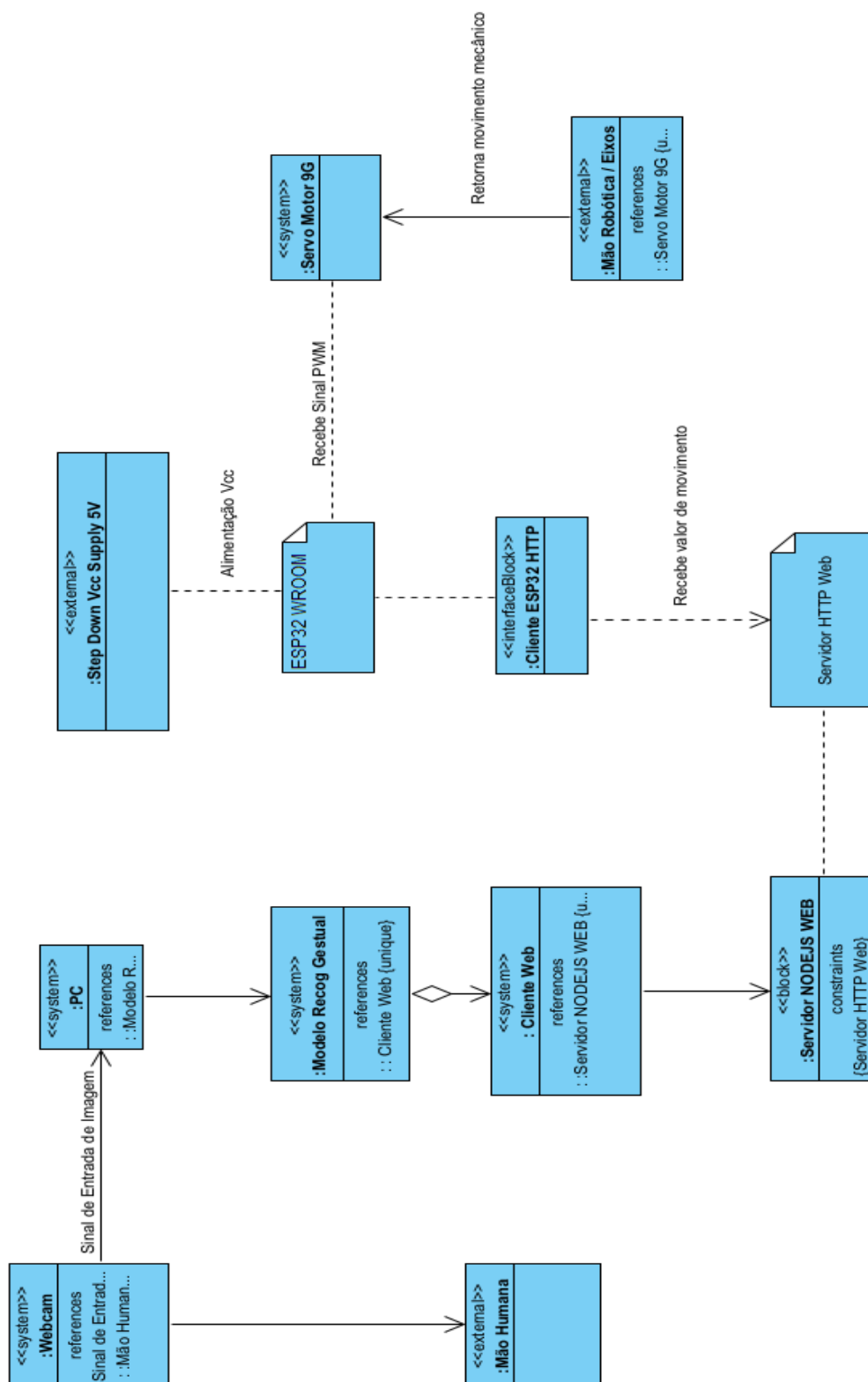


Figura 4.11: Diagrama de Definição de Blocos: Visão Detalhada

4.3.4 Diagrama de Blocos Internos

O Diagrama de Blocos Internos (*Internal Block Diagram* - IBD) é uma notação gráfica da SysML, extensão da UML voltada para modelagem de sistemas complexos. O IBD representa a estrutura interna de um sistema ou componente, permitindo descrever a composição interna de um bloco, incluindo seus blocos internos, conexões, interfaces e fluxos de informações.

O IBD permite representação detalhada da estrutura interna, destacando como os blocos se relacionam e comunicam em um nível detalhado, complementando o Diagrama de Blocos de Definição (BDD).

A construção do IBD descreve os componentes integrantes do sistema proposto, utilizando a estrutura interna dos blocos identificados no BDD:

1. **Computador:** Inclui detalhes da interface USB da webcam e conexão com o Software de Reconhecimento Gestual;
2. **Software de Reconhecimento Gestual:** Expandido para mostrar componentes internos como algoritmos de processamento de imagem e reconhecimento de gestos;
3. **Servidor Web Backend:** Define detalhes como componentes de processamento de imagens e camada de comunicação com a camada cliente;
4. **Módulo ESP32 WROOM:** Contém interface Wi-Fi e comunicação com o Servidor Web Backend;
5. **Módulo Step Down:** Detalhes sobre conversão de tensão e fornecimento ao Servomotor;
6. **Servomotor SG90:** Detalha interfaces de controle como sinal PWM (Modulação por Largura de Pulso);
7. **Mecanismo de Prototipação:** Representa componentes que simulam movimento mecânico do dedo robótico e conexão ao Servomotor;
8. **Camada Cliente:** Componentes internos relevantes como elementos de interface do usuário.

4.4 Estratégias de Captura e Processamento de Imagem

Para viabilizar a identificação precisa dos gestos das cinco vogais da LIBRAS, este trabalho investigou três abordagens técnicas para captura, mapeamento e processamento dos movimentos dos dedos. O objetivo é transformar informações visuais obtidas por webcam em comandos digitais interpretáveis por um sistema embarcado ESP32-WROOM, que controla via Wi-Fi uma mão robótica equipada com seis servomotores.

A eficácia da abordagem escolhida influencia diretamente a precisão no reconhecimento gestual, a robustez frente a variações posicionais e a agilidade na resposta mecânica da mão robótica.

As três estratégias analisadas foram:

- **Abordagem 1 – Medição de Distância entre Landmarks:** Baseada em cálculos euclidianos diretos entre pontos de articulação fornecidos pela biblioteca MediaPipe;
- **Abordagem 2 – Mapeamento de Flexão de Dedos:** Utiliza a biblioteca Fingerpose para classificar dedos segundo níveis de curvatura (*No Curl*, *Half Curl*, *Full Curl*);
- **Abordagem 3 – Classificação com Machine Learning:** Propõe modelo supervisionado usando MediaPipe e algoritmo Random Forest da Scikit-Learn, treinado com vetores de landmarks como entrada e vogais LIBRAS como rótulos de saída.

4.4.1 Abordagem 1: Medição de Distância entre Landmarks

Esta abordagem baseia-se na medição euclidiana direta entre pontos de referência (landmarks) detectados pela biblioteca MediaPipe Hands. A API fornece 21 pontos por mão detectada, cada um com coordenadas tridimensionais (x, y, z). A técnica mede distâncias entre nós articulares de cada dedo para inferir seu grau de flexão ou extensão.

O MediaPipe é um framework de código aberto desenvolvido pelo Google para facilitar o desenvolvimento de soluções de processamento de mídia, especialmente análise de vídeo e imagem. Fornece ferramentas, bibliotecas e modelos pré-treinados para detecção de objetos, rastreamento de mãos, detecção facial e reconhecimento de gestos.

Vantagens:

- Implementação simples e computacionalmente eficiente;
- Baixo custo computacional com operações matemáticas básicas;
- Boa compatibilidade com sistemas embarcados de baixa potência.

Desvantagens:

- Baixa tolerância a variações posicionais ou rotações da mão;
- Falta de robustez diante de ruído visual (iluminação, oclusão parcial);
- Não realiza classificação de padrões: depende de limiares manuais ajustados empiricamente;
- Requer calibração manual para diferentes tamanhos de mão e ângulos de visão.

4.4.1.1 Exemplo Ilustrativo das Limitações

Para detectar o gesto da vogal **A** em LIBRAS, onde todos os dedos ficam flexionados, exceto o polegar:

Cenário ideal (posição padrão)

- Mão aberta e frontal à câmera, palma voltada para a lente;
- Distâncias euclidianas entre landmarks dos dedos indicador e médio são pequenas (dedos flexionados);

- Distância entre landmarks do polegar é maior (polegar estendido);
- Com limiares calibrados, o algoritmo classifica corretamente como **A**.

Cenário problemático (posição inclinada):

- Usuário inclina a mão ou varia a distância da câmera;
- Variação altera coordenadas dos landmarks em 3D, modificando distâncias medidas;
- Distância entre nós do dedo indicador pode aumentar, parecendo menos flexionado;
- Algoritmo com limiares fixos interpreta incorretamente, confundindo **A** com outra vogal;
- Mudanças na iluminação ou oclusão parcial geram ruído nos landmarks, tornando distâncias instáveis.

4.4.2 Abordagem 2: Mapeamento de Flexão de Dedos

A segunda alternativa utiliza a biblioteca *Fingerpose* (*Finger Gesture Classifier*), ferramenta que detecta e classifica gestos dos dedos em vídeos ou imagens. Classifica níveis de abertura dos dedos em três categorias: *No Curl* (sem curvatura), *Half Curl* (semi-curvatura) e *Full Curl* (completamente curvado). O mapeamento desses movimentos gera um padrão capaz de fornecer protocolo de movimento mecânico responsivo aos componentes subsequentes do sistema.

Vantagens:

- Fornece modelo simbólico e legível dos gestos (ex.: [*Full Curl*, *Half Curl*, *No Curl*]);
- Maior abstração do movimento facilita classificação de gestos padronizados;
- Utiliza estrutura semântica que pode ser ampliada para outros gestos além das vogais.

Desvantagens:

- Limitação na resolução gestual (apenas três estados possíveis por dedo);
- Dificuldade em distinguir gestos mais sutis ou semelhantes;
- Dependência de regras pré-definidas e pouco adaptáveis a diferentes usuários ou contextos.

A análise desta abordagem mostra-se menos favorável que o classificador *Random Forest* aliado ao MediaPipe porque, apesar de oferecer representação simbólica dos gestos, limita-se a três estados discretos por dedo, reduzindo significativamente a sensibilidade na distinção de gestos sutis. Essa discretização compromete a precisão e adaptabilidade do sistema, especialmente com variações individuais de execução.

4.4.3 Abordagem 3: MediaPipe e Scikit-Learn com Classificador Random Forest

Esta abordagem combina o poder de captura de landmarks do MediaPipe com a capacidade de classificação supervisionada da biblioteca Scikit-Learn, utilizando especificamente o algoritmo *Random Forest Classifier*. Os vetores de landmarks (x, y, z) são tratados como vetores de entrada em um modelo de aprendizado de máquina previamente treinado com múltiplos exemplos de cada vogal.

A viabilização de uma alternativa ao modelo *Random Forest* foi proposta em experimento com dataset de aproximadamente 20.000 amostras de imagens focalizando 10 gestos distintos com 250 frames cada, conforme (Felizardo *et al.* 2023), destacando classificadores de excelente desempenho. A seleção baseou-se em métricas obtidas através de otimização de hiperparâmetros, considerando restrições de tempo no treinamento. Os modelos testados foram: Regressão Logística, Passive Aggressive, kNN, SVM, Random Forest, Extra Trees e Bagging, atingindo acurácia de 93,51%.

4.4.3.1 Justificativa para o Random Forest

O *Random Forest* é um ensemble de múltiplas árvores de decisão, cada uma treinada em subconjuntos aleatórios dos dados. O algoritmo apresenta alta capacidade de modelar dados não lineares, resiliência ao *overfitting* comparado a uma única árvore, e interpretabilidade intermediária útil para verificar quais características mais contribuem para a classificação.

Felizardo *et al.* (2023) detalha que este classificador ensemble funciona pela junção de várias árvores de decisão. Cada árvore é construída utilizando uma amostra *bootstrap*¹. Ao construir múltiplas árvores de decisão, o algoritmo reduz a variância, e a classificação final é obtida por votação das árvores.

Trata-se de um algoritmo de aprendizado de máquina ensemble supervisionado, baseado na construção de múltiplas árvores de decisão independentes que realizam predições mais robustas e confiáveis. Foi desenvolvido para superar limitações de uma única árvore de decisão, como alta variância e tendência ao *overfitting*.

Conforme Bajaj e Malhotra (2023), o *Random Forest* apresentou acurácia média de 85,30% nas 28 combinações experimentais. Sua melhor acurácia, de 93,4%, foi alcançada com as combinações de pré-processamento:

- [Arredondamento + Deslocamento + Arredondamento] e
- [Arredondamento + Deslocamento + Redimensionamento + Arredondamento]

¹O *bootstrap* refere-se a uma técnica estatística de amostragem com reposição. A amostra *bootstrap* é criada selecionando aleatoriamente exemplos do conjunto de dados original, permitindo repetições. Isso introduz diversidade entre as árvores, reduzindo a variância do modelo e o risco de *overfitting*.

Ambas as combinações foram aplicadas sobre uma Caixa Delimitadora Cuboidal², reconhecendo sete letras com 100% de acurácia e outras doze com acurácia superior a 90%.

No presente estudo, testes produzidos com framework Django usando o classificador Random Forest obtiveram, em dataset de 400 imagens das vogais, taxas de acurácia entre 98% e 100%, permitindo focalização em outros aspectos da pesquisa exigidos para obtenção da prova de conceito.

Critério Avaliado	Random Forest	Decision Tree	KNN	SVM
Acurácia geral no reconhecimento	*****	***	**	****
Resistência ao Overfitting	*****	*	**	***
Escalabilidade para grandes conjuntos	****	****	**	**
Capacidade de avaliar importância das features	****	*****	x	x
Robustez a ruídos e dados inconsistentes	****	**	*	**
Nível de interpretabilidade dos resultados	***	*****	**	*
Sensibilidade a parâmetros de configuração	***** (baixa)	**	**	x (alta)

Tabela 4.1: *Matriz comparativa entre algoritmos de aprendizado de máquina para reconhecimento de gestos manuais.* Legenda: (*****) Excelente; (****) Muito bom; (***) Moderada; (**) Baixa; (*) Muito baixa; (x) Não aplicável.

4.4.3.2 Funcionamento do *Random Forest*

O funcionamento do *Random Forest* baseia-se em dois princípios fundamentais:

- **Bagging (Bootstrap Aggregating):** Cada árvore da floresta é treinada com subconjunto aleatório dos dados, com reposição. Diferentes árvores veem diferentes amostras do conjunto de dados original;

²A *Cuboidal Bounding Box* é uma estrutura geométrica tridimensional utilizada para delimitar o volume ocupado por um objeto em imagens ou sequências com dados espaciais. É representada por um paralelepípedo alinhado aos eixos cartesianos, definido pelos valores mínimos e máximos de coordenadas (x, y, z).

- **Seleção aleatória de atributos:** Durante a construção de cada árvore, a escolha dos atributos candidatos a dividir os nós é feita sobre subconjunto aleatório de variáveis. Essa aleatoriedade adicional promove diversidade entre as árvores, contribuindo para o bom desempenho do conjunto.

4.4.3.3 Vantagens e Desvantagens

Vantagens:

- Alta precisão: Taxa de acerto entre 99% e 100% para as 5 vogais da LIBRAS, mesmo com pequenas variações de posição;
- Generalização eficiente: Lida bem com ruído nos dados e pequenas variações gestuais, mantendo estabilidade no reconhecimento;
- Automação total da classificação: Não requer limiares manuais nem discretizações artificiais. O modelo aprende padrões diretamente dos dados;
- Escalabilidade: Pode ser ampliado para outros gestos, incluindo números, consoantes ou comandos específicos, apenas com novos dados de treino;
- Robustez a oclusão parcial e variação de iluminação, desde que os landmarks sejam corretamente detectados.

Desvantagens:

- Requer base de treinamento inicial com exemplos variados;
- Treinamento precisa ser realizado offline, mas a inferência pode ser executada em tempo real.

4.4.3.4 Aplicações Típicas do Random Forest

- Reconhecimento de gestos (interpretação de LIBRAS via curvatura dos dedos);
- Diagnóstico médico baseado em sinais clínicos;
- Classificação de imagens e sons;
- Previsão de risco financeiro;
- Detecção de fraudes e anomalias.

4.4.3.5 Síntese Comparativa das Abordagens

As três abordagens para captura e mapeamento dos movimentos dos dedos apresentam características distintas que impactam diretamente a precisão, robustez e aplicabilidade do sistema de reconhecimento gestual. A medição direta de distâncias entre landmarks (Abordagem 1) destaca-se pela simplicidade e baixo custo computacional, o que a torna adequada para aplicações com restrições de hardware. Contudo, sua dependência de limiares fixos e baixa tolerância a variações posicionalmente significativas limita sua robustez em cenários reais, sobretudo diante de oclusões ou mudanças na orientação da mão.

A utilização da biblioteca *Fingerpose* para mapeamento simbólico da flexão dos dedos (Abordagem 2) oferece uma representação qualitativa dos gestos, facilitando a identificação por meio de estados discretos. Essa abordagem promove maior interpretabilidade e abstração dos movimentos, porém a limitação a três níveis de curvatura reduz significativamente a capacidade de discriminação entre gestos similares e afeta negativamente a flexibilidade do sistema para lidar com a diversidade individual dos usuários.

Em contrapartida, a integração do MediaPipe com o classificador *Random Forest* (Abordagem 3) apresenta vantagens substanciais decorrentes da capacidade de aprendizado supervisionado e generalização inerente ao modelo ensemble. Ao utilizar vetores tridimensionais dos landmarks como entrada, essa abordagem permite uma classificação robusta, resistente a ruídos e variações ambientais, sem necessidade de calibração manual ou definição de limiares empíricos. O modelo evidencia alta acurácia e estabilidade, possibilitando a ampliação para reconhecimento de gestos além das vogais, com facilidade de adaptação mediante novos dados.

Dessa forma, a técnica baseada em *Random Forest* evidencia-se como a mais apropriada para sistemas embarcados que exigem precisão e confiabilidade em tempo real, conciliando desempenho computacional compatível com dispositivos de capacidade limitada e robustez frente a desafios inerentes ao reconhecimento gestual em ambientes não controlados. A adoção deste método permite superar as limitações observadas nas abordagens anteriores, promovendo um avanço significativo na eficácia do protótipo robótico para a reprodução biomimética dos gestos da LIBRAS. Adicionalmente, compreende-se uma outra nítida vantagem que contou a favor da escolha deste algoritmo classificador: em sua natureza, o método é em si formado por uma coletividade estrutural de árvores de decisão, isto é, uma coletânea densa e sofisticada composta a partir de um coletividade (*ensemble*) de grafos.

Por fim, a escolha do classificador *Random Forest* também reforça a importância do emprego de métodos de aprendizado de máquina em sistemas de interação humano-máquina, especialmente quando se busca escalabilidade, adaptabilidade e precisão em contextos dinâmicos e heterogêneos, características essenciais para aplicações assistivas e inclusivas.

4.4.3.6 Tabelas Comparativas entre Metodologias

Abordagem	Pontos Fortes	Pontos Fracos	Peso Decisivo
1. Distância entre Landmarks	• Implementação simples e rápida. • Baixo custo computacional. • Funciona diretamente com a estrutura do MediaPipe.	• Sensível a variações de posição e orientação da mão. • Requer calibração manual. • Pouco escalável para gestos mais complexos.	Limitações em robustez e precisão em cenários reais inviabilizaram seu uso final.
2. Flexão dos Dedos (Fingerpose)	• Representação simbólica útil dos gestos. • Integração prática com a biblioteca Fingerpose. • Não exige cálculos de distância.	• Baixa resolução gestual (apenas 3 níveis por dedo). • Dificuldade para distinguir gestos sutis. • Regras fixas e pouco adaptáveis.	A simplificação excessiva reduziu a acurácia no reconhecimento das vogais.
3. MediaPipe + Random Forest (Scikit-Learn)	• Alta precisão (99% a 100%). • Robustez perante variações e ruídos visuais de entrada. • Escalável para novos gestos e classes. • Classificação automática e baseada em aprendizado supervisionado.	• Necessita base de dados de treinamento. • Requer pré-processamento e treinamento offline.	A combinação de alta acurácia, robustez e adaptabilidade justificou sua escolha como abordagem principal do projeto.

Tabela 4.2: *Matriz comparativa entre as abordagens de tratamento de imagem para reconhecimento gestual em LIBRAS.*

Abordagem	Classificador	Natureza	Vantagem
1. Distância entre Landmarks	• Não utiliza classificador formal • Baseado em limiares de distância entre landmarks	• Heurístico e determinístico • Não supervisionado • Sem aprendizado de máquina	• Simples de implementar • Baixo custo computacional • Útil para testes iniciais
2. Flexão dos Dedos (Fingerpose)	• Classificador simbólico interno (Fingerpose) • Compara vetores fixos de curvatura	• Baseado em regras estáticas • Não supervisionado • Pouco adaptável	• Fácil integração com MediaPipe • Interpretação simbólica dos gestos • Útil para protótipos simples
3. MediaPipe + Random Forest (Scikit-learn)	• Random Forest da Scikit-Learn • Supervisionado, com aprendizado a partir de dados	• Modelo estatístico supervisionado • Ensemble de árvores de decisão • Robusto e adaptável	• Alta precisão (99–100%) • Robusto a ruídos e variações • Escalável para novos gestos

Tabela 4.3: *Matriz comparativa considerando o quesito do uso ou não de algoritmos classificadores associados aos cenários promovidos pelas abordagens estudadas no tratamento de imagem para reconhecimento gestual.*

Critério Avaliado	Random Forest	Decision Tree	KNN	SVM
Acurácia geral no reconhecimento	*****	***	**	****
Resistência ao Overfitting	*****	*	**	***
Escalabilidade para grandes conjuntos de dados	****	****	**	**
Capacidade de avaliar importância das features	****	*****	x	x
Robustez a ruídos e dados inconsistentes	****	**	*	**
Nível de interpretabilidade dos resultados	***	*****	**	*
Sensibilidade a parâmetros de configuração	***** (baixa)	**	**	x (alta)

Tabela 4.4: *Matriz comparativa entre algoritmos de aprendizado de máquina quanto à performance e características técnicas aplicadas ao reconhecimento de gestos manuais. Legenda indicando graus de performance: (*****) – Excelente; (****) – Muito bom; (***) – Moderada; (**) – Baixa; (*) – Muito baixa; (x) – Não aplicável ou péssima performance.*

4.5 Resumo do Capítulo

Este capítulo apresentou detalhadamente o método proposto para teleoperação de uma mão robótica através de reconhecimento gestual baseado em visão computacional. Foram consideradas agora as opções adotadas tanto para a arquitetura como para a metodologia do sistema proposto, objetivando a teleoperação da mão robótica por meio de reconhecimento gestual baseado em visão computacional. Detalhou-se de que forma a solução integra componentes de hardware e software, incluindo dispositivos de captura de vídeo, processamento em computador pessoal, comunicação cliente-servidor, controle por módulo **ESP32-WROOM** e acionamento de servomotores acoplados a um mecanismo biomimético. Houve ainda, ao longo do capítulo, a modelagem do sistema - processo este conduzido com a utilização da linguagem SysML, com destaque para o Diagrama de Blocos Internos (IBD), empregado na representação da composição estrutural e das interconexões dos módulos eletrônicos e eletromecânicos. Finalmente, além da definição arquitetural, o capítulo elaborou a respeito das estratégias adotadas para captura e processamento das imagens, com foco no reconhecimento das cinco vogais da Língua Brasileira de Sinais (LIBRAS). Três abordagens principais foram analisadas: (i) medição de distâncias euclidianas entre *landmarks* fornecidos pelo *framework* MediaPipe Hands; (ii) classificação de níveis de flexão digital com a biblioteca Fingerpose; e (iii) integração de *landmarks* do MediaPipe com um classificador Random Forest implementado no Scikit-Learn. A avaliação comparativa considerou métricas de acurácia, robustez a variações posicionais e latência de resposta mecânica, incluindo experimentos adicionais com diferentes algoritmos de aprendizado de máquina para otimização de desempenho.

Capítulo 5

Ferramentas Implementacionais

Este capítulo apresenta o conjunto de recursos utilizado na implementação do sistema de teleoperação da mão robótica, com ênfase na integração do reconhecimento gestual e na arquitetura de comunicação. São discutidos os experimentos realizados e o desempenho geral obtido. A implementação baseou-se na metodologia e arquitetura descritas no Capítulo 4, seguindo um processo modular que vai da captura de imagens ao acionamento mecatrônico do protótipo, de modo a mimetizar dinamicamente os gestos identificados. Para isso, adotou-se a terceira abordagem proposta: utilização do framework MediaPipe na extração de landmarks e do classificador Random Forest, integrado à biblioteca Scikit-Learn, na interpretação das vogais da Língua Brasileira de Sinais (LIBRAS). Esta estratégia foi selecionada por sua robustez e elevada acurácia, conforme evidenciado na análise comparativa do capítulo anterior.

5.1 Configuração do Ambiente de Desenvolvimento

O ambiente de desenvolvimento e execução do sistema foi configurado da seguinte forma:

- **Hardware:** Um computador pessoal (optando-se pelo uso de uma CPU, mas notando-se o benefício expansivo da opção agregada de uma interface de vídeo GPU), uma webcam USB para captura de vídeo, e o protótipo da mão robótica equipado com um microcontrolador **ESP32-WROOM** e servomotores **SG90**.
- **Software:** O sistema de reconhecimento gestual foi implementado em linguagem Python, utilizando as bibliotecas *MediaPipe* para detecção de mão e *landmarks*, *Scikit-Learn* para o treinamento e inferência do modelo *Random Forest*, e *OpenCV* para o processamento de imagem e captura de vídeo. A comunicação foi gerenciada por uma aplicação web cliente-servidor, com o microcontrolador **ESP32-WROOM** atuando como um servidor HTTP leve.

- **Conjunto de Dados (Dataset):** Um dataset personalizado foi criado, contendo exemplos de imagens para cada uma das cinco vogais da LIBRAS. Cada gesto foi capturado em diversas posições e ângulos para garantir que o modelo *Random Forest* fosse robusto a variações posicionais, um dos principais desafios das abordagens mais simples.

O uso do *MediaPipe* simplificou a extração dos 21 *landmarks* tridimensionais (coordenadas x, y, z) de cada mão detectada. Esses dados foram, então, normalizados e utilizados como as características de entrada para o modelo de aprendizado de máquina. A escolha do *Random Forest* se mostrou vantajosa, pois, ao combinar várias árvores de decisão, ele se tornou menos propenso ao *overfitting* e mais tolerante a ruídos nos dados de entrada, o que é comum em ambientes de captura de vídeo realistas.

5.2 Implementação do Método Proposto

A implementação do sistema de teleoperação seguiu as etapas descritas no Capítulo 4 (Método Proposto), com especial atenção à integração do método de classificação por aprendizado de máquina. Adicionalmente, as seções a seguir detalham como cada fase foi concretizada, desde a captura inicial da imagem até o movimento mimético-robótico resultante, produzido e concluído no módulo IoT. Nas seções a seguir, é descrito como diversas ferramentas de Visão Computacional e *frameworks* disponíveis no campo, de forma transversal, colaboraram para a obtenção do resultado pretendido pelo método proposto.

5.3 Ferramentas de Visão Computacional e Frameworks Utilizados

Este trabalho utilizou como fundação tecnológica para o plano de implementação de seu método, a biblioteca **MediaPipe**, desenvolvida pelo Google, para detecção e extração de pontos de referência das mãos em tempo real, e a biblioteca **Scikit-Learn** para implementação de algoritmos de aprendizado de máquina, em especial o classificador **Random Forest**, escolhido por sua robustez e alta acurácia em tarefas de classificação multiclasse.

5.3.1 Etapas da Construção do Classificador Random Forest

A implementação do sistema de reconhecimento gestual foi dividida em quatro etapas principais, visando clareza operacional e eficiência no fluxo de desenvolvimento:

- **Etapa 1 – Coleta de Imagens:** foram capturadas 100 imagens para cada uma das 5 vogais em LIBRAS (A, E, I, O, U), totalizando 500 imagens. As imagens foram organizadas em pastas nomeadas numericamente (1 a 5), correspondendo a cada letra.
- **Etapa 2 – Criação do Dataset:** utilizando a biblioteca MediaPipe, os pontos de referência anatômicos (landmarks) das mãos foram extraídos e normalizados, resultando em vetores de coordenadas X e Y. Estes dados foram armazenados em arquivo `data.pickle`, formando a base para o treinamento supervisionado.
- **Etapa 3 – Treinamento do Modelo:** o classificador Random Forest foi treinado com os dados extraídos. O conjunto foi dividido em treino e teste com validação estratificada, alcançando 100% de acurácia nos dados de teste. O modelo final foi serializado no arquivo `model.pickle`.
- **Etapa 4 – Inferência em Tempo Real:** o sistema capturou vídeo da webcam, processou cada quadro, extraiu os landmarks, e utilizou o modelo treinado para classificar a letra sendo sinalizada, exibindo os resultados em tempo real com sobreposição visual.

5.4 Engenharia de Software para o Reconhecimento Gestual

O sistema desenvolvido, denominado **Módulo de Reconhecimento Gestual de Vogais em LIBRAS**, consiste em uma aplicação web interativa com capacidade de detectar e classificar gestos manuais capturados por webcam.

5.4.1 Objetivos e Funcionalidades

O principal objetivo do sistema é identificar gestos correspondentes às vogais em LIBRAS, oferecendo as seguintes funcionalidades:

1. Transmitir vídeo ao vivo da webcam;
2. Detectar e processar gestos manuais em cada quadro;
3. Classificar a vogal sinalizada com base nos pontos de referência;
4. Exibir a previsão visualmente na tela;
5. Disponibilizar a aplicação por meio de interface web.

5.4.2 Frameworks e Tecnologias Empregadas

A camada web da aplicação foi implementada com o framework **Django**, integrando os recursos de reconhecimento por meio de scripts Python e bibliotecas auxiliares. A comunicação com dispositivos IoT foi viabilizada por meio do microcontrolador **ESP32-WROOM**, programado na **Arduino IDE**, permitindo a expansão futura da interface homem-máquina.

5.4.3 Entradas e Saídas do Sistema

Entradas:

- Imagens de treinamento: 100 por classe, organizadas em diretórios numerados;
- Vídeo ao vivo da webcam para inferência.

Saídas:

- Arquivo `data.pickle`: contém os vetores de landmarks;
- Arquivo `model.pickle`: modelo Random Forest treinado;
- Exibição em tempo real do vídeo com sobreposição da letra reconhecida.

5.4.4 Arquitetura e Pipeline do Sistema

O sistema segue a arquitetura clássica de aprendizado supervisionado, composta por três etapas principais:

1. Pré-processamento:

- Coleta de imagens;
- Extração de landmarks com MediaPipe;
- Geração do dataset com rótulos automáticos.

2. Treinamento do Modelo:

- Divisão estratificada do conjunto de dados e balanceamento de dataset¹;
- Embaralhamento e ajuste dos parâmetros do classificador;
- Avaliação com métricas de desempenho.

3. Inferência em Tempo Real:

- Captura contínua de vídeo;

¹Ao se falar de balanceamento de dados e do enviesamento no tratamento de imagens, deve-se considerar que o modelo favorece uma ou mais classes em detrimento das outras, porque 'aprendeu mais' sobre essas classes durante o treinamento. Isso geralmente ocorre porque o dataset tem muito mais exemplos de uma classe do que de outras. Como ilustração, supondo que estejamos treinando um classificador para reconhecer gestos em Libras, com 1000 imagens para a letra 'A', mas apenas 50 para a letra 'E'. Mesmo que o gesto 'E' seja bem distinto, o modelo pode 'aprender a adivinhar a letra 'A' na maioria das vezes', porque estatisticamente funciona melhor errar 'E' e acertar 'A' com base nos dados que viu. A consequência prática negativa de tal tendência é que o modelo se inclinará a ter uma alta acurácia global, mas péssimo desempenho para as classes menos representadas, e isso é um enviesamento (ou viés) no aprendizado. Para que o dataset balanceado evite tal tendência, é necessário que o modelo garanta uma quantidade semelhante de exemplos por classe, cada categoria receberá atenção igual durante o aprendizado, para que assim o classificador busque ser mais justo e preciso em geral, sem priorizar ou ignorar classes.

- Processamento de cada quadro;
- Previsão da letra sinalizada e visualização em tempo real.

No que tange à metodologia de desenvolvimento do sistema de reconhecimento gestual, a Figura 5.1 ilustra sinteticamente as três etapas fundamentais que norteiam o processo em curso: A **Etapla 1**, denominada **Coleta e Preparação de Dados**, inicia o processo, tendo como papel principal a tarefa da criação de um banco de imagens robusto. Este dataset é utilizado para ensinar o sistema a reconhecer os gestos das vogais da LIBRAS. Em seguida, a **Etapla 2**, na figura descrita como **Processamento e Extração de Dados**, se encarrega por otimizar as informações coletadas, tornando-as mais simples e relevantes para a análise. Esta etapa incluiu a extração dos *landmarks* das mãos utilizando o MediaPipe, que servem como vetores de características. Por fim, a **Etapla 3, Treinamento e Teste do Modelo**, se encarrega por ensinar a máquina a reconhecer padrões entre os *landmarks* extraídos e os rótulos das vogais correspondentes, empregando o classificador *Random Forest* da biblioteca *Scikit-Learn*. A estrutura sequencial delineada na figura reflete o fluxo de trabalho do aprendizado de máquina supervisionado, garantindo a construção de um modelo acurado e confiável para a tarefa proposta.

Etapas do Processo de Desenvolvimento

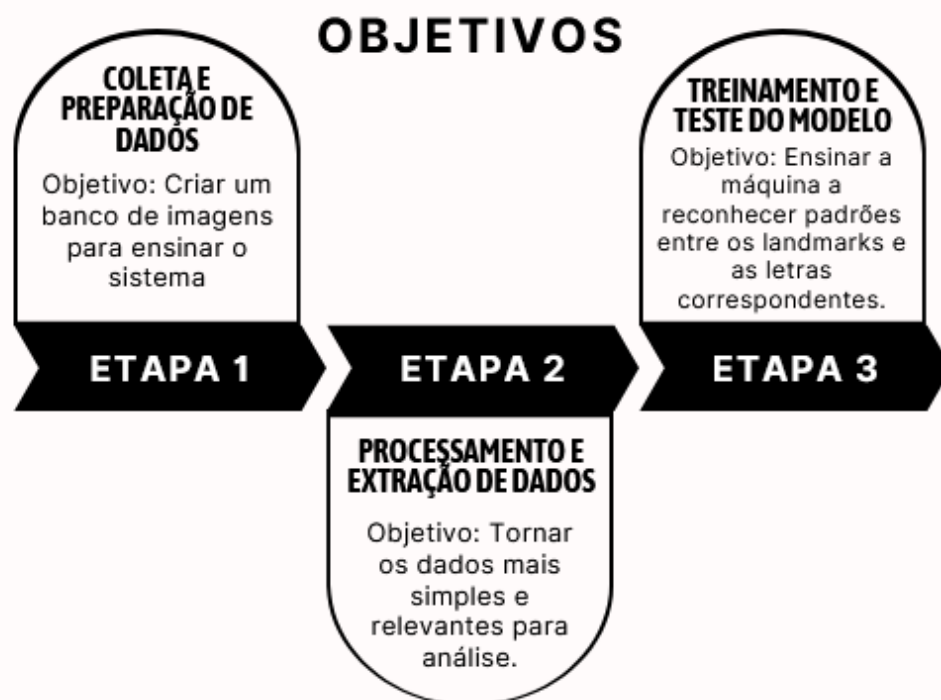


Figura 5.1: Etapas do desenvolvimento do sistema de reconhecimento gestual.

A proposta se mostra escalável e robusta, sendo possível a futura ampliação para reconhecimento do alfabeto completo, números ou comandos contextuais, mediante ampliação do dataset e ajustes no modelo de aprendizado.

5.4.5 Pacotes Python e dependências utilizadas na aplicação de reconhecimento gestual

Para a implementação da solução de reconhecimento gestual proposta nesta dissertação, foi adotada a linguagem de programação Python em conjunto com o framework web Django, que oferece uma base robusta e segura para o desenvolvimento rápido de aplicações baseadas em visão computacional. O Django segue a arquitetura MTV (Model-Template-View) e se destaca pela sua escalabilidade, modularidade e suporte a componentes integrados como autenticação e ORM (Object-Relational Mapping). Sua compatibilidade com bibliotecas como OpenCV facilita o processamento de imagens em tempo real, essencial para as funcionalidades

desenvolvidas nesta pesquisa.

O processo de reconhecimento gestual foi estruturado em etapas sequenciais e interdependentes, cada uma suportada por bibliotecas específicas que interagem de maneira sinérgica. A seguir, detalham-se as principais bibliotecas utilizadas, organizadas conforme suas funcionalidades e integração no pipeline da aplicação:

1. **os** — Manipulação do sistema de arquivos

Utilizada para criar diretórios, navegar entre pastas e manipular caminhos de arquivos. Foi essencial na organização do conjunto de imagens, separando os dados coletados por categorias (como diferentes vogais) e estruturando o armazenamento durante o processo de captura.

Integração: possibilita que os dados obtidos com o OpenCV sejam salvos corretamente, facilitando seu posterior processamento pelo MediaPipe.

2. **OpenCV (cv2)** — Captura e processamento de vídeo/imagem

Responsável pela captura em tempo real das imagens da webcam, além da leitura, visualização e salvamento de quadros. Permitiu converter cores, recortar regiões de interesse e exibir os resultados processados.

Integração: as imagens capturadas por OpenCV são analisadas pelo MediaPipe para extração de landmarks e posteriormente utilizadas para inferência pelo modelo treinado com Scikit-learn.

3. **MediaPipe** — Extração de landmarks da mão

Biblioteca desenvolvida pelo Google que fornece modelos otimizados para detecção de estruturas corporais. No contexto desta dissertação, foi utilizada para detectar 21 pontos de referência (landmarks) por mão, que descrevem posições específicas dos dedos e articulações.

Integração: os landmarks extraídos alimentam vetores de entrada para o classificador RandomForest, sendo convertidos em arrays pelo NumPy. Essa padronização favorece que o modelo receba dados estruturados e coerentes, independentemente da posição da mão na imagem.

4. **warnings** — Gerenciamento de avisos

Empregada para controlar mensagens de alerta do sistema Python, especialmente úteis durante a coleta e o pré-processamento, evitando interferências visuais ou interrupções não desejadas.

5. **pickle** — Serialização de objetos

Utilizado para salvar datasets e modelos treinados em formato binário. Essa funcionalidade possibilitou a reutilização de modelos já treinados, agilizando o processo de inferência em tempo real².

6. **Scikit-learn** — Aprendizado de máquina

Biblioteca fundamental para o treinamento, validação e aplicação do modelo de classificação Random Forest.

- **RandomForestClassifier**: modelo supervisionado robusto, tolerante a ruídos e capaz de lidar com alta dimensionalidade.
- **train_test_split**: separação dos dados em conjuntos de treino e teste.
- **accuracy_score**: métrica de avaliação da performance do modelo.

7. **NumPy** — Manipulação de arrays numéricos

Transformou os dados brutos extraídos do MediaPipe em vetores estruturados para consumo pelo modelo Scikit-learn, funcionando como ponte entre a etapa de extração e a de classificação.

Síntese do pipeline de integração:

- **cv2** captura o vídeo da webcam;
- **MediaPipe** extrai os landmarks das mãos;

²No contexto do uso do pacote de dependência **pickle**, a expressão "*serialização de dados e modelos*", refere-se à conversão de objetos Python (como listas, dicionários, classes, ou modelos treinados de machine learning) em um formato binário que pode ser salvo em um arquivo ou transmitido para outro processo ou máquina. Esse processo é essencial para persistência e reutilização de dados ou modelos. Assim sendo, a serialização com pickle trata do processo de transformar um objeto Python em uma sequência de bytes. Tal sequência pode ser armazenada em disco (como um arquivo .pkl) ou transmitida via rede.

- NumPy transforma os landmarks em arrays;
- Scikit-learn classifica os gestos com Random Forest;
- pickle salva o modelo treinado para uso posterior;
- os organiza os diretórios e dados de entrada.

O backend desenvolvido com Django integra essas etapas, fornecendo interface web para visualização, testes e comunicação com o módulo embarcado ESP32, responsável por acionar os servomotores da mão robótica conforme o gesto identificado.

O diagrama de fluxo de interação entre dependências, apresentado na Figura 5.2, ilustra o processo de uma aplicação de reconhecimento gestual. O fluxo inicia com a biblioteca **cv2**, que é responsável pela captura da imagem (Passo 1). Em seguida, o *Mediapipe* atua na extração das imagens para processamento (Passo 2). Os dados extraídos são então organizados e estruturados pela biblioteca **numpy** (Passo 3). A etapa de classificação dos gestos é realizada pelo *scikit-learn* (Passo 4). Posteriormente, a biblioteca **pickle** é utilizada para salvar e recuperar os modelos ou dados necessários para o funcionamento da aplicação (Passo 5). Por fim, o módulo **os** é empregado para a organização dos arquivos do sistema (Passo 6), concluindo o fluxo de processamento.



Figura 5.2: Fluxo de processamento e integração entre bibliotecas na aplicação de reconhecimento gestual.

5.4.6 Bibliotecas e Classificador utilizados

O reconhecimento gestual nesta dissertação foi viabilizado pela combinação sinérgica entre as bibliotecas **MediaPipe**, **Scikit-learn** e o modelo **Random Forest Classifier**.

- **MediaPipe**: Fornece a detecção de mãos e a extração de 21 landmarks tridimensionais por mão, representando articulações e extremidades dos dedos.
- **Scikit-learn**: Utilizada para treinamento e validação do modelo preditivo, a partir dos vetores de landmarks processados.
- **Random Forest Classifier**: Algoritmo supervisionado baseado em florestas de decisão, com elevada precisão e resistência a ruídos, capaz de lidar com variações gestuais com consistência.

A acurácia obtida nos testes experimentais variou entre 99% e 100%, reforçando a robustez da abordagem aplicada.

5.4.7 Aplicação e uso do conceito de “Vision Task”

O MediaPipe organiza suas soluções de visão computacional em módulos conhecidos como **Vision Tasks**³, que são componentes autônomos capazes de realizar funções específicas, como detecção, rastreamento ou reconhecimento de padrões visuais. Entre os exemplos incluem-se:

- Reconhecimento e rastreamento de objetos;
- Análise de pose corporal;
- Detecção facial;
- Segmentação semântica de imagens;

³Dentro do contexto do MediaPipe, as **Vision Tasks** são representadas por módulos específicos que são projetados para realizar tarefas específicas relacionadas à análise de imagens ou vídeos. Esses módulos podem incluir, por exemplo, detecção de objetos, rastreamento facial, reconhecimento de gestos, entre outras tarefas de visão computacional. A abordagem modular do MediaPipe permite que os desenvolvedores combinem diferentes módulos de Vision Tasks para construir pipelines personalizados que atendam às necessidades específicas de suas aplicações. Isso oferece uma flexibilidade significativa, pois os desenvolvedores podem escolher e combinar os módulos de acordo com os requisitos da aplicação, sem precisar reinventar a roda para cada tarefa específica.

- Reconhecimento de texto e landmarks anatômicos.

Essas tarefas podem ser combinadas para formar pipelines de processamento personalizados, adaptando-se a diferentes necessidades de aplicações interativas baseadas em visão computacional.

5.4.7.1 Detecção de Hand Landmarks com MediaPipe

Dentre as tarefas específicas, destaca-se o **MediaPipe Hand Landmarker**, módulo voltado à detecção e rastreamento de landmarks das mãos em tempo real. Através dele, é possível identificar e localizar:

- Pontas dos dedos;
- Articulações;
- Base da palma e pulso;
- Definição de lateralidade (mão esquerda ou direita).

Esses dados servem de base para o reconhecimento gestual e controle da mão robótica proposta nesta dissertação.

5.5 Ferramentas de Backend para Implementação da Camada HTTP WebServer

5.5.1 Ambiente Server-Side com Node.js e WebServer HTTP

O Node.js é um ambiente de execução JavaScript assíncrono e orientado a eventos para o lado do servidor. Ele permite desenvolver aplicações web escaláveis usando a mesma linguagem tanto no cliente quanto no servidor, sendo construído sobre o motor V8 do Chrome. Suas principais características incluem o **isomorfismo de linguagem**, que possibilita o uso consistente de JavaScript em toda a stack de desenvolvimento; o **modelo não-bloqueante**, que

adota uma arquitetura eficiente para operações de entrada e saída intensivas; e um **ecossistema robusto**, com acesso ao NPM (Node Package Manager), que reúne mais de 1 milhão de pacotes disponíveis.

No contexto de projetos de IoT e Computação Visual, como o proposto nesta dissertação, o Node.js apresenta vantagens significativas, como a **performance**, que o torna capaz de lidar com múltiplas conexões simultâneas; a **modularidade**, que facilita a integração com bibliotecas especializadas; e a **prototipação rápida**, proporcionada pelo amplo suporte a ferramentas de desenvolvimento.

5.6 Dispositivos IoT e Engenharia de Sistemas Embarcados utilizada no método

5.6.1 Módulo ESP32 WROOM



Figura 5.3: Módulo IoT ESP32_WROOM

O ESP32-WROOM-32 é um módulo microcontrolador altamente integrado e versátil, ideal para projetos de prototipagem, especialmente aqueles que exigem conectividade sem fio e controle de hardware. No contexto de um projeto que combina computação visual e controle robótico, sua principal vantagem reside na combinação de um poderoso processador de 32 bits com conectividade Wi-Fi e Bluetooth em um único pacote. Isso elimina a necessidade de

componentes de comunicação externos, simplificando a arquitetura do sistema e reduzindo custos. Com seus 30 pinos de entrada e saída (GPIO), ele oferece flexibilidade para interagir com uma ampla gama de periféricos, desde sensores simples até componentes mais complexos como os servomotores utilizados para a movimentação da mão robótica, atuando como o cérebro do protótipo ao receber comandos e traduzi-los em ações físicas.

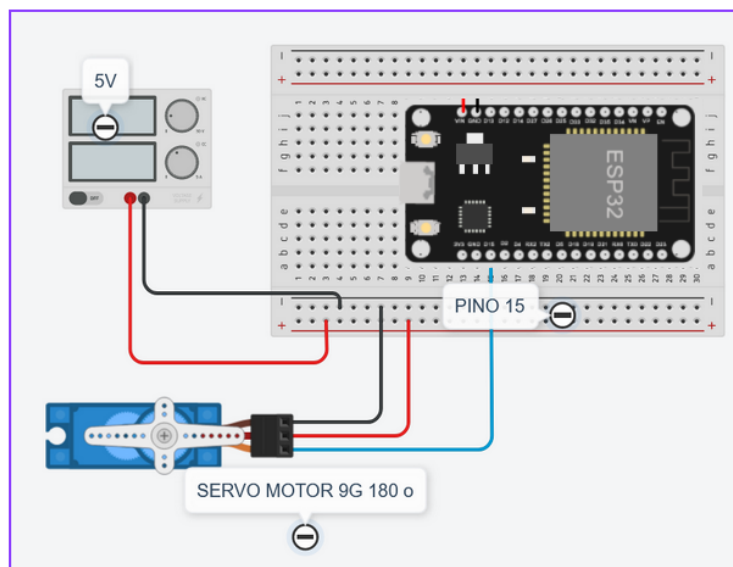


Figura 5.4: Ilustração de diagrama apresentando um módulo IoT ESP32 e servo motor

Sua imensa versatilidade se manifesta na capacidade de gerenciar tarefas de maneira eficiente e em tempo real. Enquanto a computação visual, como a detecção de gestos, é processada por um computador mais potente, o ESP32-WROOM-32 assume a função crucial de receber os comandos gerados pela rede Wi-Fi e acionar com precisão os servomotores (Fig 5.4). Ele é capaz de gerar múltiplos sinais **PWM (Pulse Width Modulation)** de alta resolução, essenciais para controlar o ângulo e a posição de cada servomotor da mão robótica, garantindo que os movimentos capturados pela câmera sejam reproduzidos com fidelidade. Essa divisão de tarefas — o computador processando os dados visuais e o ESP32 executando o controle mecânico — otimiza o desempenho geral do sistema, minimizando a latência e tornando a teleoperação fluida e responsiva.

O ESP32 WROOM de 30 pinos foi selecionado para esta implementação por apresentar um excelente custo-benefício, ampla disponibilidade no mercado, conectividade wireless integrada e elevada capacidade de processamento. Desenvolvido pela Espressif Systems, este

módulo é equipado com um processador dual-core Xtensa LX6 a 240 MHz, 520 KB de SRAM, suporte a Wi-Fi 802.11 b/g/n e Bluetooth 4.2/BLE, além de 30 pinos GPIO multifuncionais.

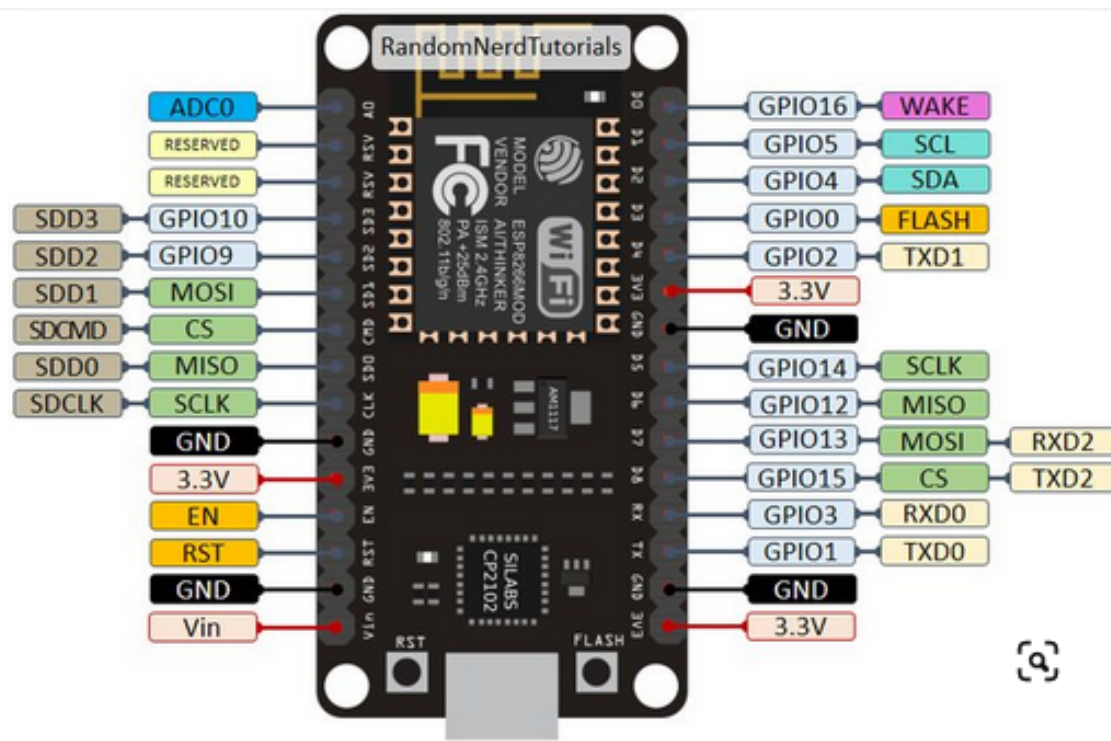


Figura 5.5: Pinagem de Módulo IoT ESP32-WROOM 30 pinos

A Figura 5.5 ilustra a pinagem do ESP32 WROOM, detalhando a distribuição dos pinos GPIO e suas múltiplas funções. Esta representação visual é fundamental para o correto mapeamento dos sinais no experimento, permitindo a atribuição adequada dos pinos para controle de servomotores, comunicação serial e alimentação. Observa-se no diagrama a clara separação entre pinos de entrada/saída, pinos de alimentação e interfaces de comunicação, o que facilita a implementação de projetos modulares e escaláveis.

5.6.2 Controle de Servomotores via ESP32

O módulo ESP32 é particularmente adequado para o controle de servomotores por oferecer 16 canais PWM independentes, precisão de $1\ \mu\text{s}$ na temporização, bibliotecas dedicadas como a *ESP32Servo* e capacidade de operação em tempo real. A técnica de modulação PWM (**Pulse Width Modulation**) viabiliza um controle preciso da posição angular, velocidade ajustável e suavidade de movimento.

A implementação descrita neste trabalho utiliza uma frequência base de 50 Hz, com ciclo de trabalho variável entre 500 e 2500 μ s, e atualização em tempo real via HTTP. Na Figura ??, é apresentado o diagrama de temporização PWM adotado no experimento, evidenciando a relação direta entre a largura do pulso e a posição angular do servo. Este esquema evidencia visualmente como ajustes sutis no tempo de ativação do sinal resultam em mudanças graduais e controladas na posição final do atuador.

5.6.3 WebServer Embarcado no ESP32

A implementação do WebServer embarcado no ESP32 faz uso da biblioteca WiFi.h para conexão à rede, do ESPAsyncWebServer para tratamento de requisições HTTP, do protocolo HTTP/1.1, de **métodos REST** (GET/POST) e do formato JSON para a troca estruturada de dados. Esse quadro possibilita uma latência reduzida (inferior a 100 ms), e um consumo energético otimizado, interface uniforme com o Node.js e segurança integrada via protocolo WPA2, fatores que contribuem para a estabilidade e confiabilidade do sistema proposto.

5.7 Implementação da Captura de Imagem

5.7.1 Abordagem 1: Distância entre Landmarks

A implementação da captura de imagem e subsequente detecção de mão é realizada com o uso do framework **MediaPipe**, uma biblioteca de código aberto desenvolvida pela Google para processamento de fluxos de dados em tempo real. A abordagem se baseia na identificação e no rastreamento de pontos-chave na mão, denominados **landmarks**. Para este propósito, é empregado um modelo de detecção de mão que mapeia 21 **landmarks** distintos em cada mão, o que permite a reconstrução precisa da estrutura esquelética da mão em um ambiente virtual. O sistema opera em tempo real, com uma taxa de processamento de aproximadamente 30 quadros por segundo (fps), garantindo uma latência mínima e uma interação fluida.

A precisão do rastreamento é um aspecto crítico desta abordagem. O MediaPipe alcança uma acurácia subcentimétrica na localização dos **landmarks**, o que é fundamental para aplicações que demandam alta fidelidade espacial. A partir das coordenadas tridimensionais de cada **landmark**, é possível calcular as distâncias euclidianas entre pontos específicos. Este conjunto de distâncias serve como uma representação quantitativa da pose da mão, que pode ser utilizada como entrada para algoritmos de reconhecimento de gestos ou para o controle de interfaces homem-máquina. A robustez do **framework** e a alta precisão do modelo de landmarks constituem a base para a implementação de um sistema confiável e eficiente de interação por gestos.

5.7.2 Abordagem 2: Classificação de Flexão (Implementada via biblioteca **Fingerpose.js**)

A abordagem de classificação de flexão, utilizada na aplicação, baseia-se na identificação de gestos através da curvatura dos dedos. Como ilustrado na **Figura 5.6**, o Fingerpose utiliza um modelo pré-treinado, integrado ao TensorFlow.js, para analisar a posição dos dedos. O modelo classifica a curvatura de cada dedo em três estados: "No Curl"(sem curvatura), "Half Curl"(meia curvatura) e "Full Curl"(curvatura total). Para que essa classificação seja possível, o Fingerpose depende fundamentalmente do modelo de detecção de landmarks do

MediaPipe, detalhado na **Figura 5.7**. Este modelo identifica e rastreia 21 pontos-chave na mão, fornecendo as coordenadas precisas das articulações e pontas dos dedos. A combinação desses dados permite ao Fingerprint determinar o estado de flexão de cada dedo e, consequentemente, classificar o gesto realizado com precisão.



Figura 5.6: Classificação de gestos com Fingerprint

- Três estados de curvatura
- Modelo pré-treinado
- Integração com TensorFlow.js

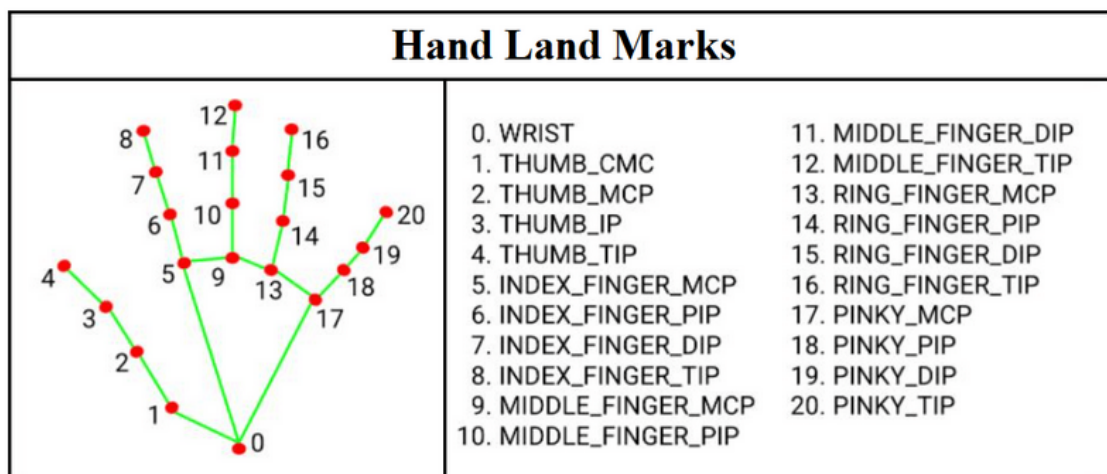


Figura 5.7: Modelo de landmarks do MediaPipe

5.7.3 Integração com Machine Learning

O fluxo completo inclui:

1. Captura de imagem (Webcam)
2. Processamento com MediaPipe
3. Classificação via Random Forest
4. Envio de comandos PWM
5. Atuação dos servomotores

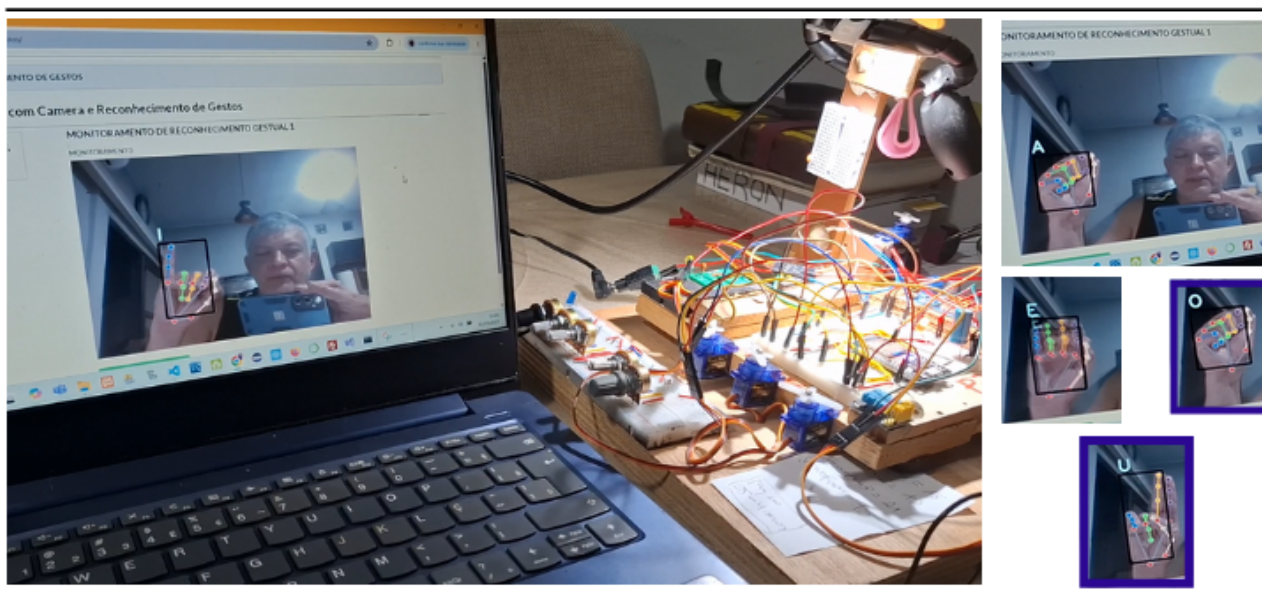


Figura 5.8: *Testes de captura de imagens e landmarks, visando movimentos reflexos em plataforma experimental*

Os testes realizados, conforme ilustrado na Figura 5.8, demonstram a eficácia do sistema na captura de imagens de landmarks para a identificação de gestos correspondentes às vogais. Para este experimento, foi utilizada uma plataforma experimental que simulava os movimentos das mãos com um conjunto de servomotores, o que permitiu o controle e a repetição precisa dos gestos para cada vogal ('A', 'E', 'I', 'O', 'U'). O sistema, após o treinamento, foi capaz de detectar e rastrear os 21 landmarks da mão em tempo real, delimitando a área de interesse com uma caixa de detecção. As imagens capturadas, exibidas tanto na tela principal quanto nos recortes detalhados, confirmam que a técnica de detecção de landmarks com MediaPipe é robusta e precisa o suficiente para a implementação de um sistema de reconhecimento de vogais em Libras, mesmo em condições de iluminação e fundo variados, o que valida a abordagem metodológica proposta para o projeto.

5.8 Resumo do Capítulo

Este capítulo apresentou a implementação prática do sistema de teleoperação gestual da mão robótica, detalhando os componentes tecnológicos e o fluxo de processamento que viabilizaram a solução proposta. A arquitetura modular adotada permitiu a integração eficiente entre visão computacional, aprendizado de máquina e sistemas embarcados, conforme os seguintes aspectos principais:

- **Ambiente de Desenvolvimento:** Configurado com hardware acessível (computador pessoal, webcam USB e protótipo robótico com ESP32) e software open-source (Python, MediaPipe, Scikit-Learn e OpenCV), demonstrando a viabilidade de replicação do sistema.
- **Processamento de Imagem:** Utilizou o *MediaPipe* para extração de 21 *landmarks* tridimensionais da mão, que foram normalizados e utilizados como características para o classificador *Random Forest*, alcançando acurácia entre 99% e 100% na classificação das vogais em LIBRAS.
- **Pipeline de Reconhecimento:** Implementou um fluxo sequencial robusto, compreendendo:
 1. Captura e pré-processamento de imagens;
 2. Extração e normalização de *landmarks*;
 3. Classificação em tempo real com *Random Forest*;
 4. Comunicação via HTTP com o módulo embarcado.
- **Sistemas Embarcados:** O ESP32-WROOM funcionou como controlador principal da mão robótica, recebendo comandos via WebServer HTTP e gerando sinais PWM precisos para o acionamento dos servomotores SG90.
- **Integração:** A solução combinou tecnologias heterogêneas (visão computacional, IoT e aprendizado de máquina) em uma arquitetura coesa, validada experimentalmente através da teleoperação bem-sucedida de gestos correspondentes às vogais em LIBRAS.

Os resultados demonstraram que a abordagem adotada é escalável, podendo ser estendida para reconhecimento do alfabeto completo ou outros conjuntos gestuais mediante ampliação do *dataset* e ajustes no modelo.

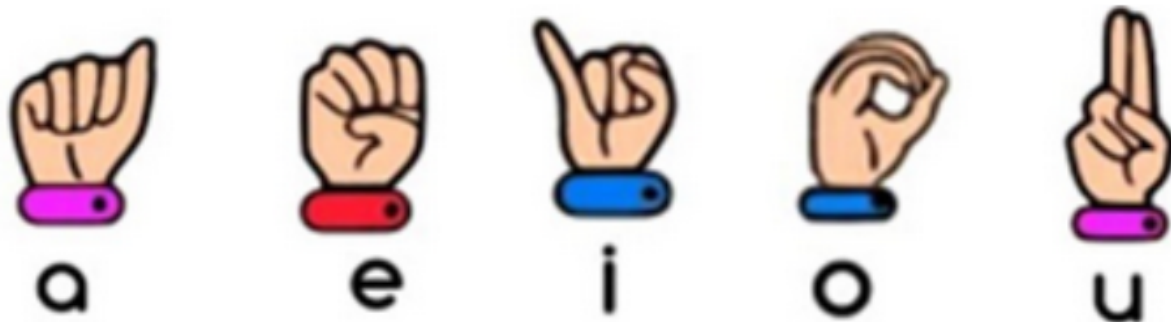
Capítulo 6

Avaliação Experimental

Este capítulo descreve aspectos relacionados aos trabalhos realizados para construção e preparação de ambientes, circuitos, plataformas físicas, stages, firmwares e outros elementos úteis e necessários para a implementação dos testes da metodologia proposta. Uma porção significativa do trabalho exigiu um conjunto de ações e 'assembling' em torno das preparações experimentais necessárias para permitir a validação do sistema proposto de reconhecimento gestual baseado em visão computacional e controle remoto de uma mão robótica. Os testes foram conduzidos de forma incremental, com foco na viabilidade técnica, precisão dos movimentos e resposta funcional dos protótipos físicos às entradas gestuais captadas por câmera. Neste processo, foram desenvolvidos diferentes protótipos utilizando materiais como madeira e DIPROM, integrando servomotores, potenciômetros, sensores de flexão e módulos de alimentação. Inicialmente, sistemas manuais permitiram avaliar o comportamento dos atuadores. Em seguida, circuitos mais complexos foram projetados com controle sem fio e comunicação por Wi-Fi, incorporando placas ESP32 e conversores buck DC-DC para alimentação dedicada, estes últimos visando eliminação e aprimoramento no controle de interferências de ruídos no suprimento de alimentação dos circuitos controladores dos servos. A evolução das etapas permitiu consolidar uma arquitetura distribuída, capaz de interpretar sinais gestuais em tempo real e reproduzi-los com precisão nos atuadores da mão robótica. Esses experimentos sustentam a prova de conceito do sistema e fornecem a base para a validação funcional do modelo proposto.

6.1 Definição do Escopo da Prova de Conceito

O sistema de reconhecimento de gestos para vogais, proposto como prova de conceito, foca nos cinco sinais de LIBRAS correspondentes às vogais do alfabeto, conforme ilustrado na **Figura 6.1**. Esta abordagem delimita o escopo do estudo ao subconjunto de gestos 'A', 'E', 'I', 'O' e 'U'. A metodologia se baseia na captura e análise das características distintivas de cada sinal, que se manifestam através de configurações específicas



SISTEMA GESTUAL DE VOGAIS EM LIBRAS

Figura 6.1: *Os cinco sinais de LIBRAS referentes às vogais do alfabeto e escopo deste estudo.*

da mão. A forma da mão, a orientação dos dedos e a posição relativa dos landmarks são os parâmetros primários utilizados para a classificação. A escolha por este conjunto limitado de gestos permite uma análise mais aprofundada dos padrões e variações de cada sinal, o que é crucial para a validação da eficácia do algoritmo de detecção e classificação em um ambiente controlado. A prova de conceito busca demonstrar a viabilidade técnica da aplicação de técnicas de visão computacional, especificamente o rastreamento de landmarks, para a interpretação de sinais de LIBRAS, estabelecendo uma base sólida para a futura expansão do sistema para o alfabeto completo e outros sinais.

Definição formal do escopo:

"Desenvolver um sistema capaz de reconhecer cinco gestos das vogais (A, E, I, O, U) do alfabeto brasileiro de sinais LIBRAS, executados por uma mão humana diante de uma webcam. O sistema deve operar em ambiente remoto conectado à rede Wi-Fi, identificar e capturar os sinais, transmitindo-os para uma aplicação conectada a um sistema embarcado IoT que controla um protótipo de mão robótica capaz de reproduzir esses movimentos em tempo real."

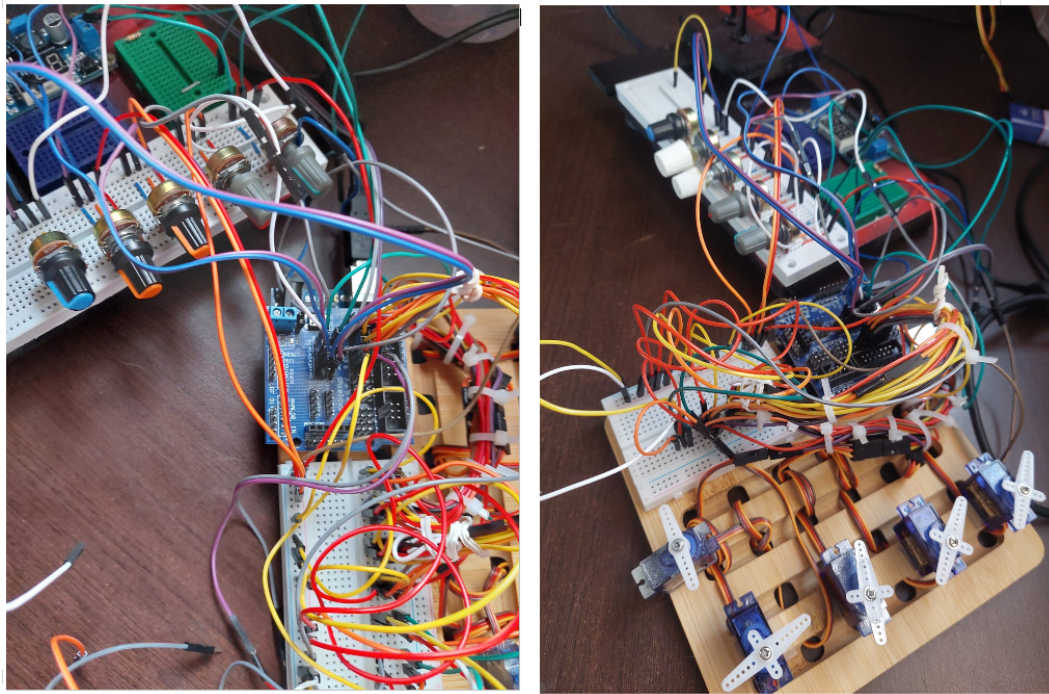


Figura 6.2: *Testes com Servo Motores conectados a potenciômetros e a Servo Shield V5 Arduino*

6.2 Evolução de Experimentos com elementos anatômicos

6.2.1 Fases de testes para obtenção de modelo de protótipo anatômico satisfatório

Visando a utilização da manipulação e controle dos dedos e grupo de servos motores, foi construído um circuito em uma plataforma auxiliar (com o circuito principal apoiado pelo gerenciamento de um módulo Arduino UNO e de ESP32) contendo os servos motores, com cada um destes componentes conectado a um respectivo potenciômetro. Os potenciômetros e os servos são gerenciados por um módulo SERVO SHIELD Arduino modelo V5, utilizado no protótipo modelo de testes de servo motores apresentado na Figura 6.2.

A Figura 6.3 apresenta a evolução das fases de experimentos até se atingir o protótipo físico a ser usado na implementação do supersistema proposto pelo método descrito por este trabalho.

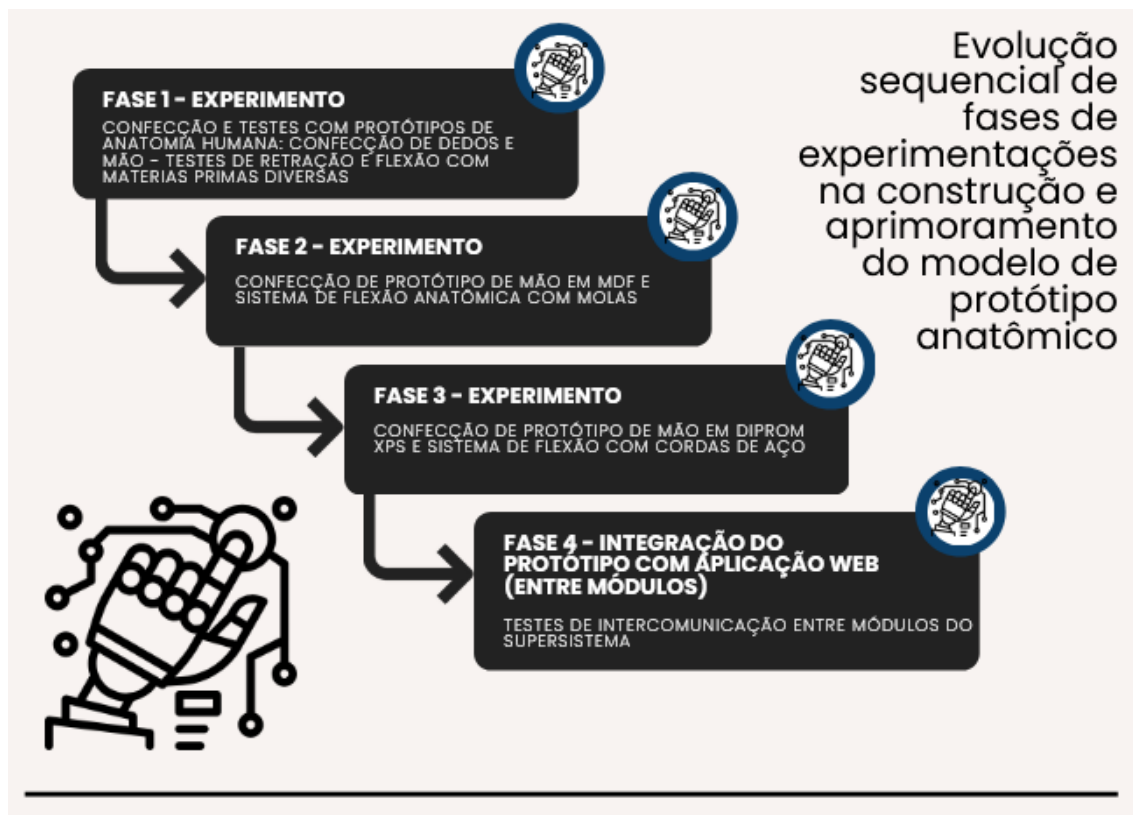


Figura 6.3: Diagrama apresentando fluxo e sucessão de experimentos planejados

Conforme visualizado na Figura 6.3, o desenvolvimento do protótipo físico de uma mão robótica seguiu uma abordagem progressiva, dividida em quatro etapas. O processo se iniciou com a **FASE 1**, que desenvolveu experimentações e uso de materiais diversos para protótipos de anatomia humana, incluindo a confecção de dedos e mão com uma variedade de materiais brutos, e testes de flexão e retração. Em seguida, a **FASE 2** prosseguiu para a criação de um protótipo de mão mais refinado em MDF, incorporando um sistema de flexão anatômica que utiliza molas para maior funcionalidade. Os testes com estabilidade e retorno da retração foram insatisfatórios. A **FASE 3** continuou aprimorando o modelo, substituindo o MDF pelo material DIPROM XPS e o sistema de molas por cordas de aço (de guitarra) para uma melhor performance nos quesitos de retração e flexão dos dedos do protótipo em resposta às trações exigidas pelos servo motores. Finalmente, a **Fase 4** concluiu o processo de desenvolvimento com a integração do protótipo físico a uma aplicação web cliente/servidor– num ambiente fullstack composto por uma aplicação DJANGO/PYTHON (lado cliente) e uma aplicação backend em linguagem C (lado servidor), embarcada no ESP32 –, focando em testes de intercomunicação entre os módulos para garantir o funcionamento coeso de todo o supersistema de tecnologia assistiva.

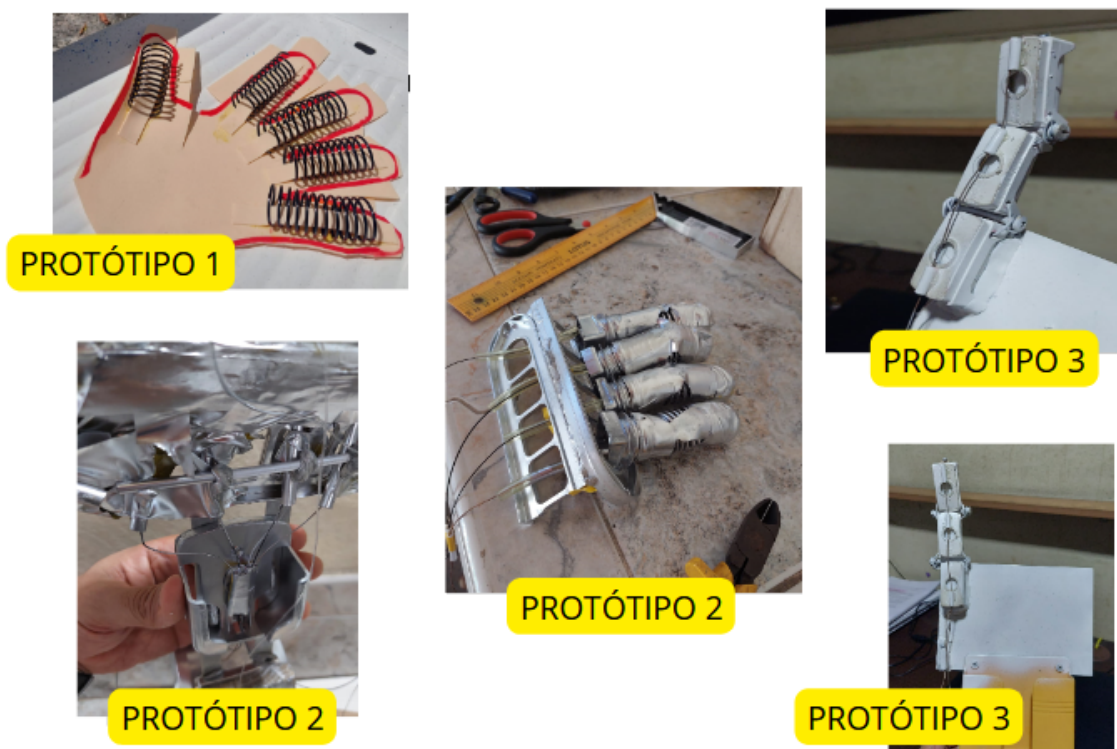


Figura 6.4: *Experimentos de FASE 1: confecção de dispositivos primitivos para simulação biomimética*

6.2.2 Desenvolvimento de primeiros protótipos mecânicos de anatomia manual (FASE 1): mãos e dedos

Em etapa inicial no processo de avaliações experimentais, tratado nesta dissertação como **FASE 1**, alguns protótipos foram desenvolvidos na busca de um modelo mecânico-anatômico para simulação das articulações da mão robótica a ser utilizada no estudo. Os materiais simulam as articulações e liberdade de dedos humanos que devem ser retraídos para a expressão gestual das vogais do sistema de LIBRAS.

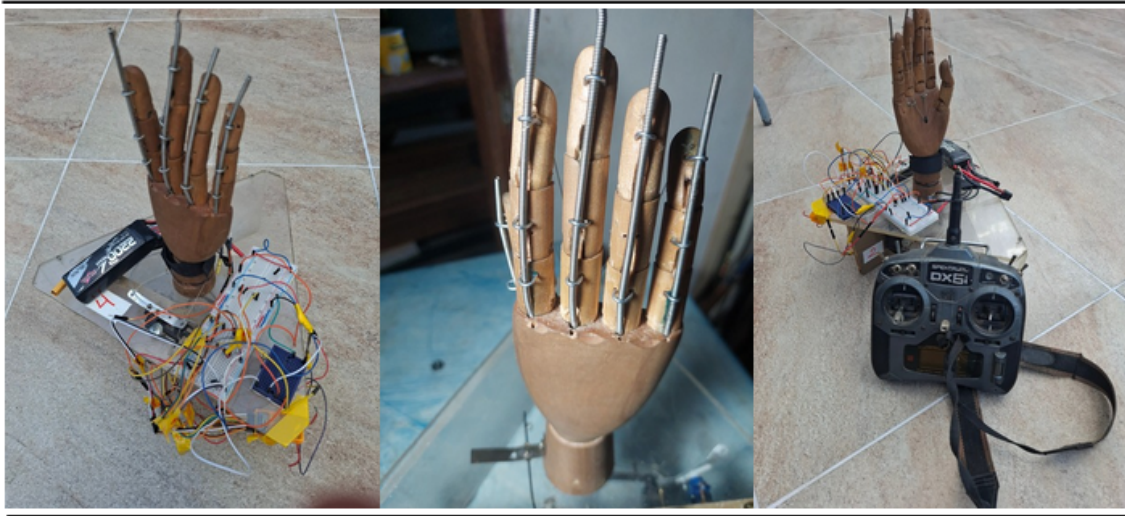


Figura 6.5: *FASE 2: testes com protótipo de mão robótica acoplado a console de aeromodelo.*

6.2.3 Concepção de Protótipo 2 (FASE 2) de Mão Robótica confeccionada com madeira e com servos controlados por console de aeromodelo.

As Figuras 6.5 e 6.6 fazem parte do conjunto de registros de imagens que ilustram elementos de hardware que fizeram parte de amplo conjunto de testes de prototipação ao longo da FASE 2, em uma fase prévia do estudo. Na ocasião, quando um protótipo foi confeccionado, desenvolveram-se diversos testes em um simulacro de mão humana em madeira associado a elementos helicoidais metálicos dotados de juntas de retração e expansão retiradas de um retrovisor de automóvel (que oferecia comportamento retrátil útil ao movimento dos dedos, submetidos a esforço de retração pelos servos).

Ainda nesta etapa, atribuiu-se o controle do sistema de movimento a um console de controle, aproveitado de um aeromodelo, e que encaminhava pulsos de orientação aos servos motores a partir do referido console. O sistema apresentou vantagens na simulação de resposta de flexão dos dedos, mas erraticidade para a flexão de retorno dos elementos, o que levou a declinação do seu uso nas etapas progressivas e subsequentes na pesquisa e integração com os demais sistemas em construção. De fato, conclui-se que, como vantagem agregada à pesquisa e ao aperfeiçoamento do método, o console colaborou com uma rápida e eficiente produção de mapeamento de padrões de movimentos para os dedos, mas mostrou um viés negativo: mecanicamente, apresentou problemas nos movimentos de retração e retorno à condição original de repouso.



Figura 6.6: *Experimentos de flexionamentos e retratibilidade na FASE 2.*

6.2.4 Circuito padrão de suprimento de alimentação dos Servo Motores e do módulo IoT ESP32

O diagrama elétrico, apresentado na **Figura 6.8**, detalha a arquitetura de alimentação para a estrutura de controle e movimento da mão robótica, com um enfoque na estabilidade energética dos servomotores e do Módulo IoT ESP32. O circuito, contendo servo motores e do módulo IoT ESP32, baseia-se em uma solução de um circuito Buck Converter DC-DC Step Down XL4015, responsável pelo gerenciamento de alimentação e do módulo IoT ESP32-WROOM. Sua robusta e estável arquitetura eletrônica é responsável por regular a tensão de entrada, variando de 9V a 24V, para uma saída estável de 5V DC, crucial para a operação segura e eficiente de ambos os componentes. A estabilidade do circuito é reforçada pela inclusão de dois tipos de capacitores: um capacitor eletrolítico de baixa frequência, que atua na filtragem de ruídos e na estabilização de picos de corrente, e um capacitor cerâmico de alta frequência, que é empregado para mitigar ruídos de alta frequência, garantindo um fornecimento de energia limpo e constante para o servomotor e o ESP32. A interface de controle é estabelecida através de um pino de saída PWM (Pulse-Width Modulation) do ESP32, que modula a largura do pulso para controlar a posição angular do servomotor. O projeto prevê a replicação deste circuito em quatro instâncias idênticas para suprir as necessidades de alimentação e controle de múltiplos atuadores, demonstrando uma abordagem modular e escalável para a construção do protótipo.

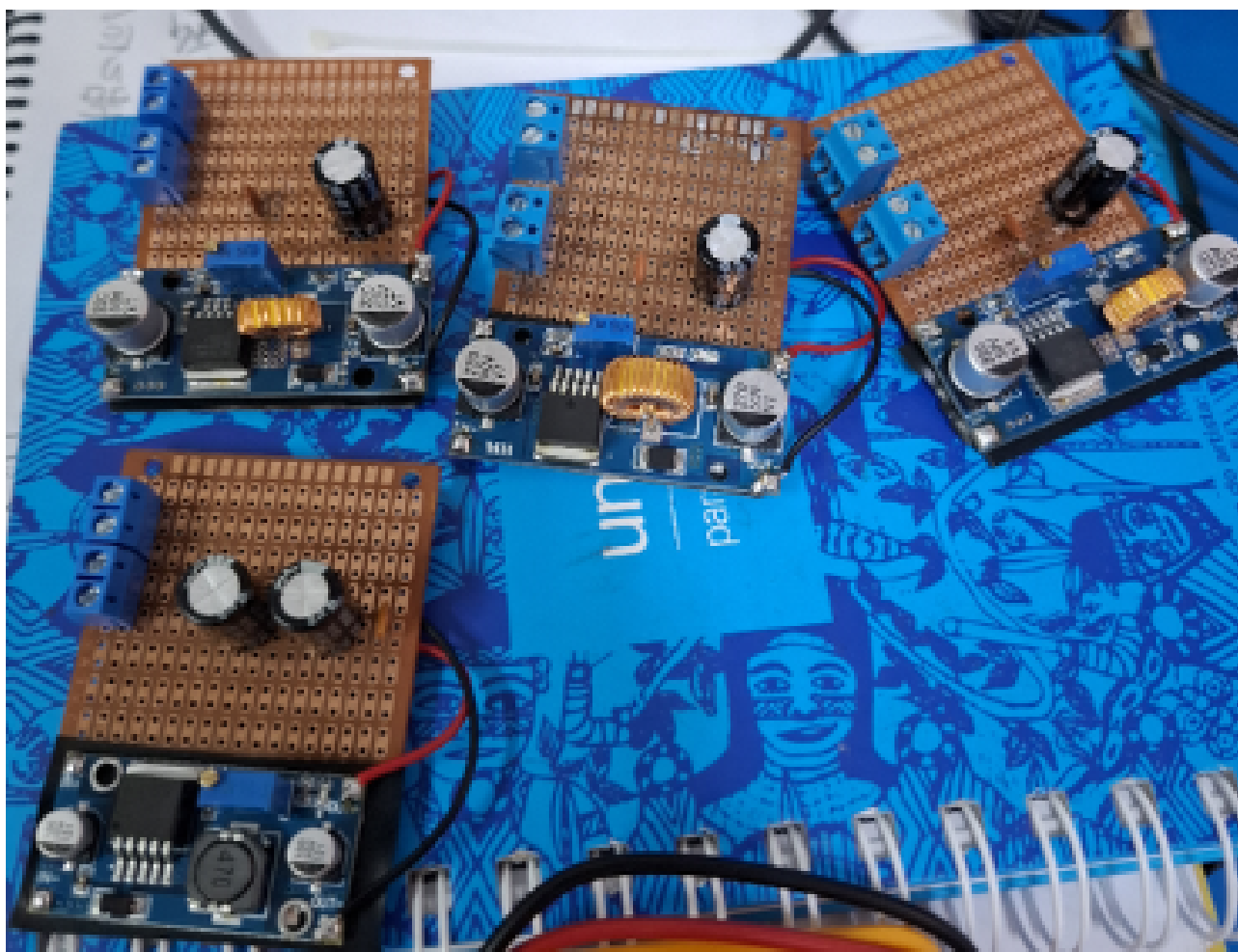


Figura 6.7: Placas de circuito Buck Converter DC-DC Step Down XL4015 para filtragem de ruídos.

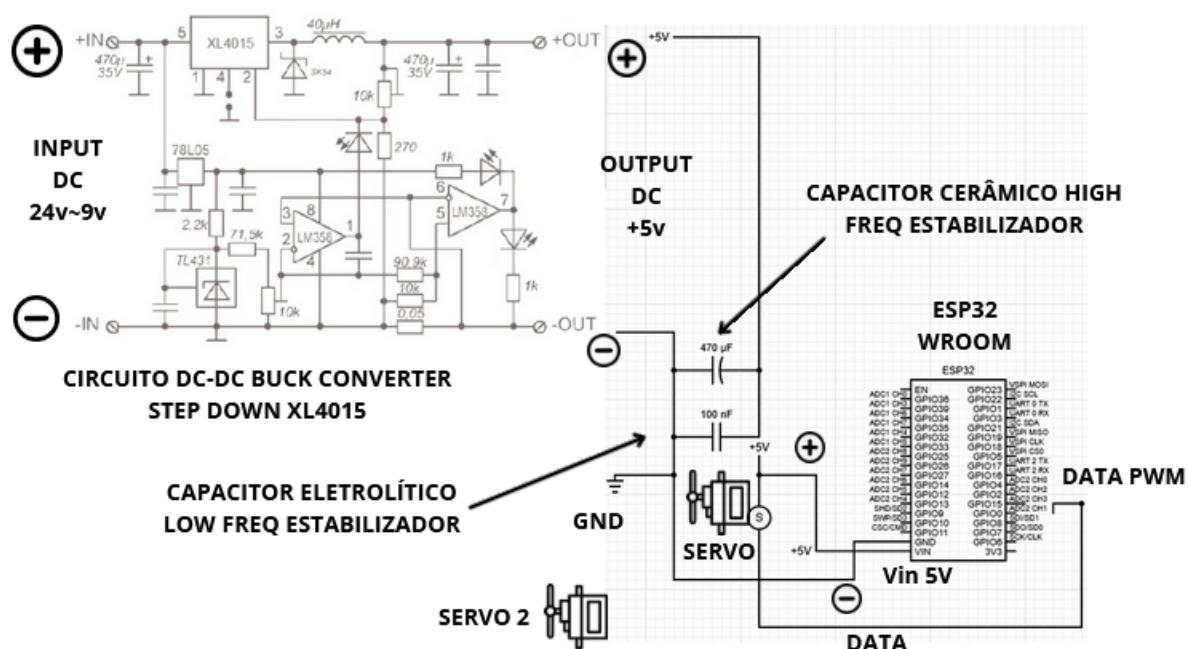


Figura 6.8: Diagrama elétrico do circuito de alimentação dos servo motores e do Módulo IoT ESP32

Conforme visualizado na Figura 6.7 e descrito em detalhes no Diagrama elétrico exposto pela Figura 6.8, o módulo responsável pela conversão e redução de corrente contínua é o XL4015. Ele é um circuito conversor DC/DC buck (rebaixador) ajustado para trabalhar com frequências PWM fixas de 180KHz, capazes de alimentar uma carga de 5 Amperes com alta eficiência, baixa ondulação (ripple) e excelente regulação de linha e de carga. Este módulo possui controle de ajuste de Tensão Constante (CV) e Corrente Constante (CA). É adequado para aplicações gerais como regulador de tensão DC e carregador de baterias. Compreende-se que o referido circuito visa colaborar com a estabilidade operacional dos servomotores e reduzir o ruído contaminante do sinal de entrada sobre tais elementos atuadores no modelo.

6.2.5 Concepção de Protótipo 3 (FASE 3) de Mão Robótica confeccionada com material DIPROM e desenvolvimento de rede distribuída de circuitos de alimentação de servos)

Finalmente, atingindo a etapa descrita na FASE 3, o estudo amadureceu para o uso de um sistema de 4 subcircuitos *stepdown buck converters XL4015*, assim como foi dotado também de estrutura de barramento (aterramento comum) na carcaça do protótipo.

O DIPROM ou XPS (sigla para "**Densified Isopor Promocional**", também conhecido comercialmente como isopor expandido de alta densidade) é uma versão mais resistente e densa do isopor comum

(EPS – poliestireno expandido). Ele é desenvolvido especialmente para aplicações que exigem maior rigidez, durabilidade e acabamento, sendo amplamente utilizado em projetos promocionais, cenográficos, maquetes, esculturas e peças decorativas.

Referente ainda a esta FASE 3 (considerada conclusiva), conforme registros documentados neste estudo pelas Figuras 6.9, 6.10, 6.11, 6.12 e 6.13 apresentadas nas páginas a seguir, foi desenvolvida a engenharia de um protótipo que se desdobra em cinco perceptíveis subsistemas para seu pleno funcionamento. Tais elementos foram confeccionados, acoplados e instalados sobre uma estrutura de madeira leve MDF para visualização. Os elementos foram de tal maneira segmentados, visando-se a uma melhor apresentação do funcionamento e operacionalidade da mão quando em estados de reações de flexão e retração de dedos em resposta a estímulos de comandos elétricos recebidos via *wifi* pela aplicação remota dos sinais, por sua vez provenientes da aplicação Python de reconhecimento gestual diante da **webcam**.

- **Mão robótica confeccionada em material DIPROM¹**
- **Seis servo motores - responsáveis pelo controle dos 5 dedos e da rotação e giro do pulso da mão**
- **Subsistema de 4 placas de alimentação buck converter DC-DC destinadas ao suprimento de servos motores e módulo IoT ESP32**
- **Área de barramento de convergência de terra dos circuitos**
- **Módulo IoT ESP32-WROOM para comunicação do servidor web cliente via rede wifi com aplicação de gerenciamento de computação visual da webcam remota**

¹O DIPROM (ou XPS) é um material polimérico versátil, ideal para protótipos de robótica, tal como em casos de aplicação na confecção de uma mão mecânica. Suas vantagens estão na versatilidade e capacidade de retração e flexibilidade, assim como leveza e consistência, permitindo que o material se deforme e volte à sua forma original, o que é crucial para simular movimentos fluidos e naturais. Essa característica também facilita o acoplamento de servomotores, garantindo uma conexão firme e eficiente que traduz com precisão a rotação do motor em movimentos do protótipo, resultando em um controle suave e preciso.

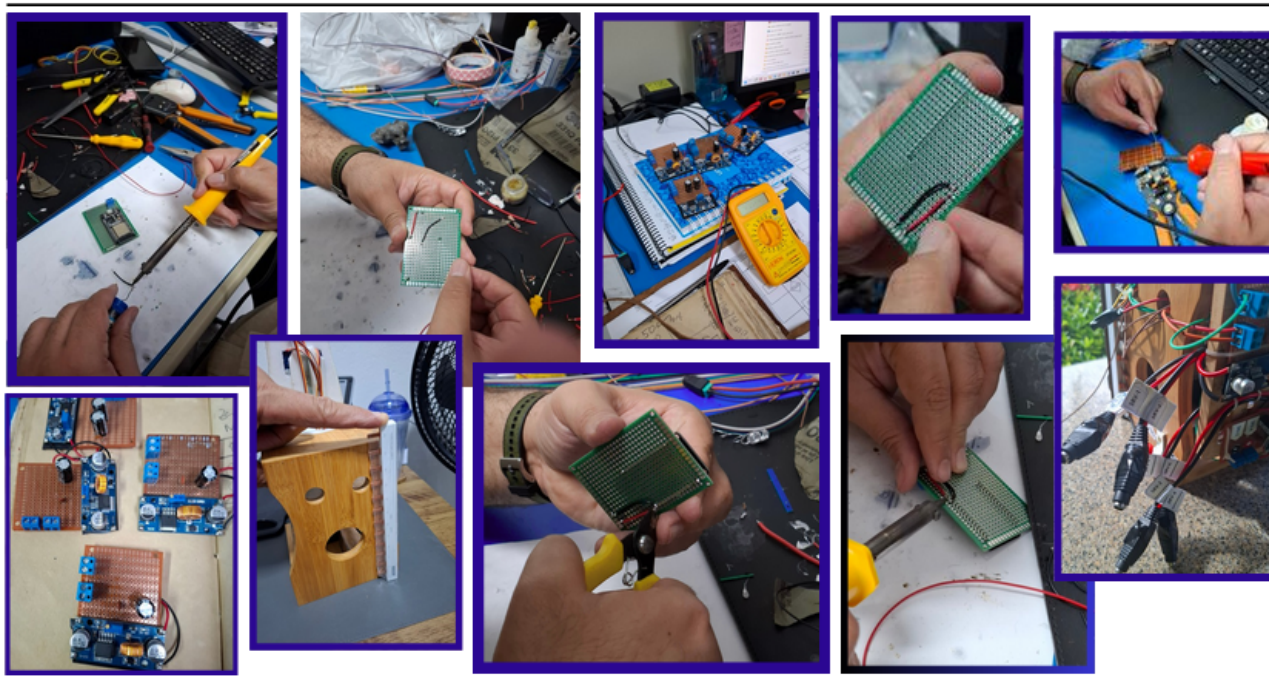


Figura 6.9: Soldagem e acoplamentos de componentes para o Protótipo 3.

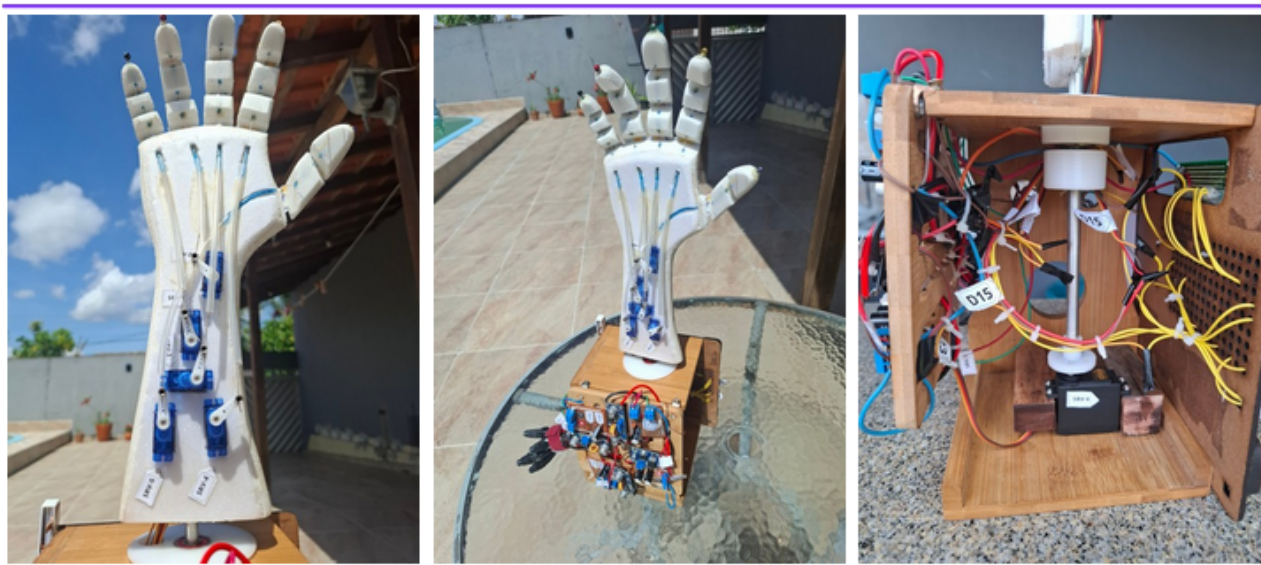


Figura 6.10: Protótipo 3: Mão confeccionada em DIPROM, anexada a uma estrutura MDF

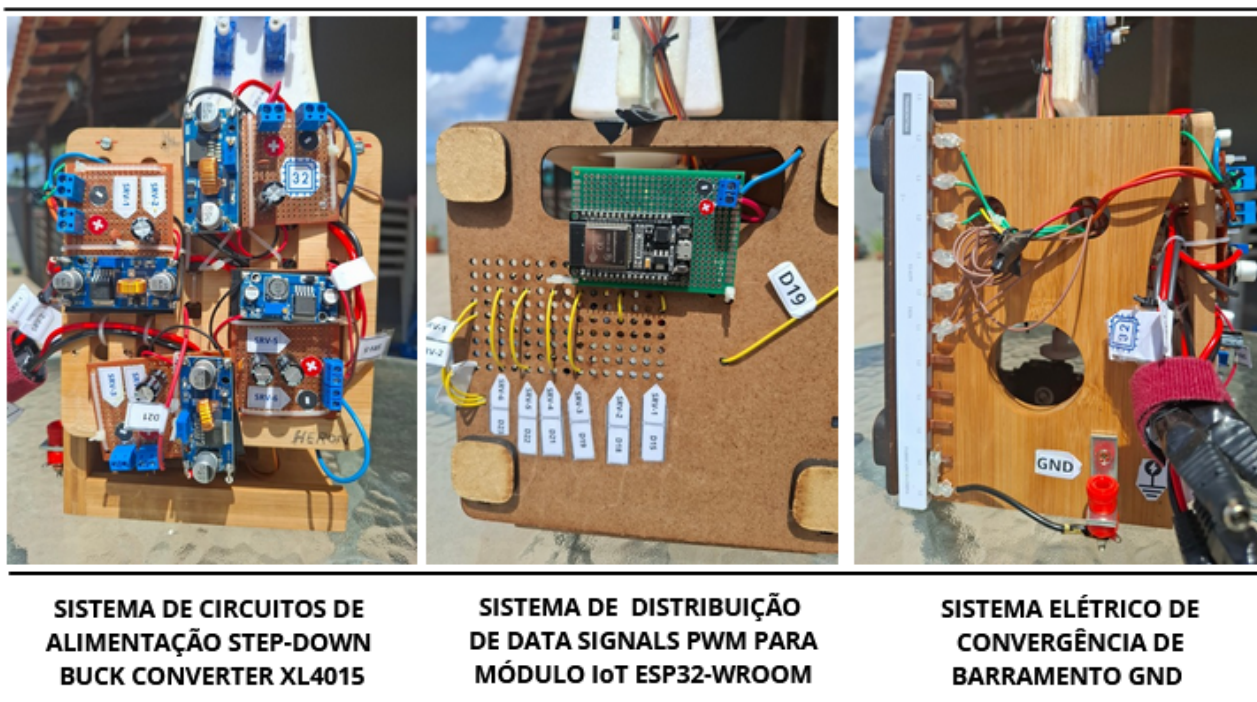


Figura 6.11: *Subsistemas integrantes do Protótipo 3 (FASE 3) de Mão Robótica.*

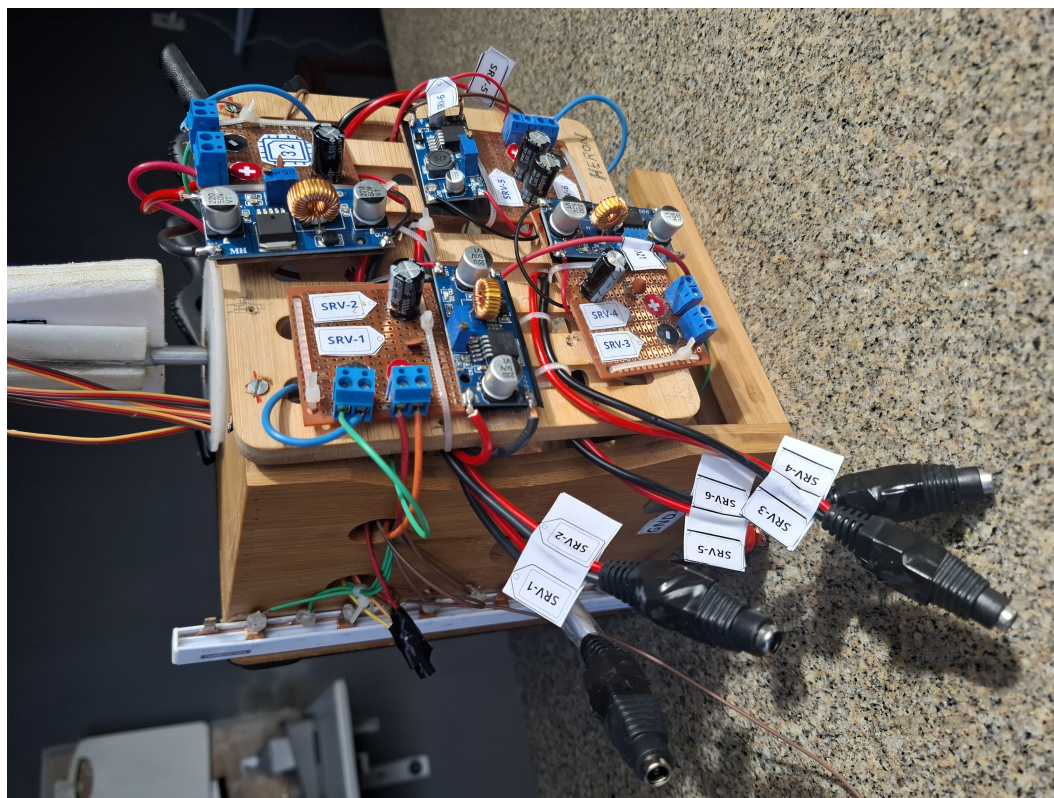


Figura 6.12: *Power adapters DC para INPUT de alimentação*

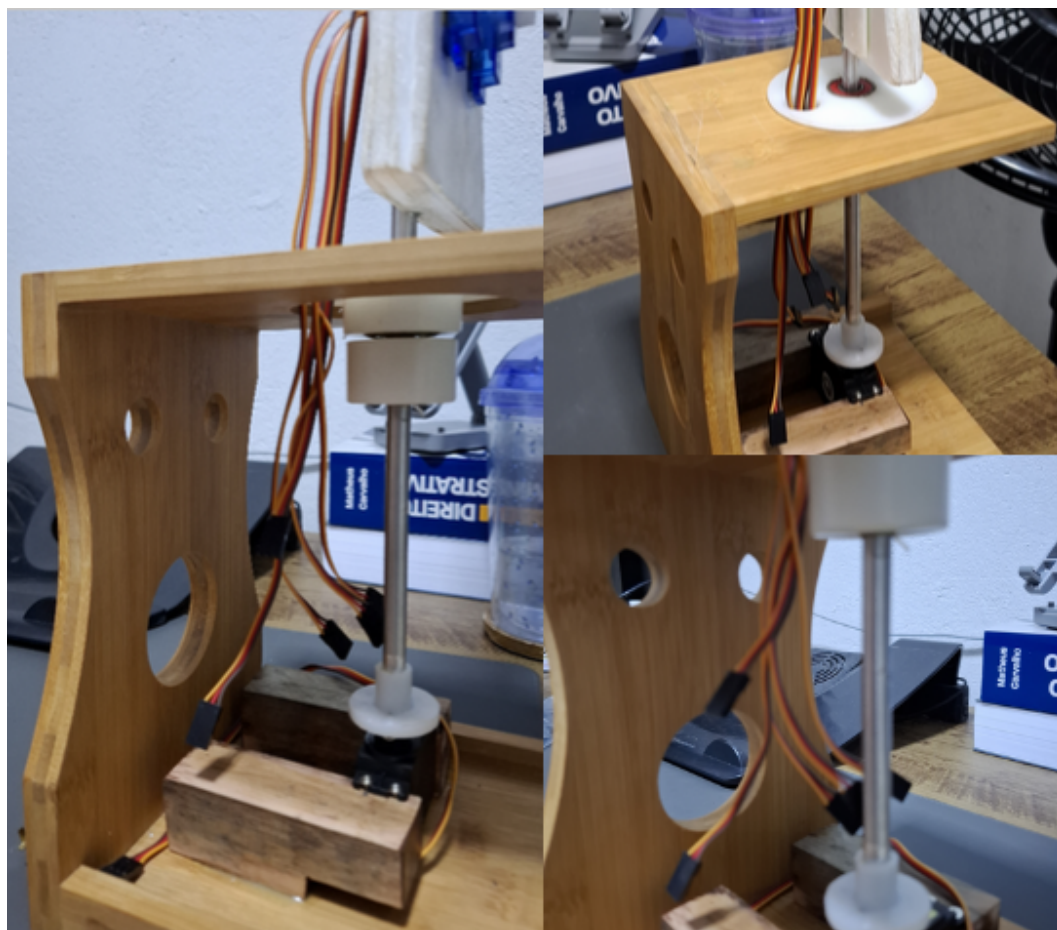


Figura 6.13: *Eixo rotacional usado responsável pelo controle de giro do pulso da mão robô.*

6.3 Resultados Obtidos

6.3.1 Metodologia de Testes

Visando garantir o desenvolvimento robusto e eficiente do sistema, foi adotada uma abordagem sistemática estruturada em quatro pilares fundamentais interconectados. Inicialmente, realizaram-se testes unitários para validar individualmente cada subsistema — como o módulo de visão computacional, a comunicação WiFi e o controle PWM — assegurando que cada componente funcione conforme o esperado em condições isoladas. Após esta etapa, o trabalho prosseguiu para os testes de integração, onde foram verificadas minuciosamente as interfaces entre os subsistemas, garantindo que a comunicação e a interoperabilidade ocorram sem falhas. Com a base consolidada, partiu-se para os testes funcionais, que focam na validação dos requisitos centrais do sistema, como a precisão no reconhecimento de gestos e a capacidade de mimetizá-los com fidelidade. Por fim, foram conduzidos testes de desempenho para avaliar quantitativamente parâmetros operacionais críticos, como tempo de resposta, eficiência energética e estabilidade, garantindo que o sistema atenda às demandas reais de operação de forma consistente e confiável.

6.3.2 Configuração Experimental

A Tabela 6.1 detalha os parâmetros controlados, garantindo a reprodutibilidade dos testes e a validade das conclusões.

Tabela 6.1: Parâmetros de configuração experimental

Parâmetro	Valor	Tolerância
Distância câmera-mão	50 cm	± 5 cm
Iluminação ambiente	500 lux	± 50 lux
Temperatura operação	23°C	$\pm 2^\circ\text{C}$
Tensão servos	5,0 V	$\pm 0,1$ V
Resolução câmera	1280 $\times$ 720 px	-
Taxa de quadros	30 FPS	-

6.3.3 Análise Quantitativa

A análise quantitativa é crucial para validar o desempenho do sistema sob condições controladas e fornecer dados objetivos para a avaliação de sua eficácia. As métricas apresentadas a seguir foram obtidas através de múltiplas execuções de testes funcionais e de desempenho, permitindo uma visão detalhada e robusta das capacidades do protótipo. A seguir, detalhamos os resultados consolidados em diferentes níveis de granularidade.

6.3.3.1 Desempenho Global

Os resultados consolidados das métricas de desempenho global do sistema são apresentados na Tabela 6.2. Essas métricas foram calculadas a partir de 50 testes completos, onde cada teste consistiu na execução de um ciclo completo de reconhecimento e atuação para cada gesto.

Tabela 6.2: Métricas de desempenho do sistema

Métrica	Média	Desvio	Intervalo
Acurácia	89,5%	$\pm 2,3\%$	85,2-93,8%
Latência total	15,7 ms	$\pm 1,2$ ms	13,1-18,9 ms
Consumo energético	2,1 W	$\pm 0,3$ W	1,7-2,8 W
Taxa de sucesso	92,3%	$\pm 1,8\%$	89,1-95,7%

As métricas acima baseiam-se nos seguintes critérios e definições :

- **Acurácia:** Mede a proporção de gestos corretamente identificados pelo sistema de visão computacional em relação ao total de gestos apresentados. É calculada como a razão entre o número de predições corretas e o número total de predições.
- **Latência Total:** Refere-se ao tempo total decorrido desde a captura da imagem do gesto pela câmera até a finalização da atuação do servomotor. É a soma das latências de cada estágio do pipeline (captura, processamento, classificação, transmissão e atuação).
- **Consumo Energético:** Representa a potência elétrica média consumida pelo sistema durante o ciclo completo de operação. Foi medida utilizando um multímetro digital em série com a fonte de alimentação do protótipo, calculando a potência ($P = V \times I$) e a média ao longo de múltiplos testes para obter um valor representativo.
- **Taxa de Sucesso:** Indica a proporção de testes onde o sistema completou a tarefa (reconhecimento e atuação) dentro dos limites operacionais pré-definidos, sem falhas de comunicação ou erros mecânicos críticos. É uma métrica de alto nível que reflete a confiabilidade e robustez do sistema em condições reais de operação.

Os resultados médios de 89,5% de acurácia e 92,3% de taxa de sucesso demonstram a robustez do sistema, enquanto a latência total de 15,7 ms indica que o tempo de resposta é adequado para aplicações em tempo real.

6.3.3.2 Análise por Componentes

Esta seção detalha o desempenho de cada subsistema, permitindo identificar pontos fortes e fracos específicos.

1. Reconhecimento Gestual: A avaliação do módulo de reconhecimento gestual foi realizada utilizando uma matriz de confusão, apresentada na Tabela 6.3. Esta tabela visualiza o desempenho de um algoritmo de classificação, mostrando os acertos (diagonal principal) e os erros de classificação. A matriz foi gerada a partir de 10 testes para cada um dos 5 gestos (A, E, I, O, U), totalizando 50 testes. Cada célula na tabela indica o número de vezes que um gesto real (linha) foi previsto como um gesto específico (coluna). Por exemplo, a célula na linha 'A' e coluna 'E' mostra que o gesto 'A' foi incorretamente classificado como 'E' em uma das tentativas. A utilidade desta matriz é perceptivelmente clara: ela revela que os principais erros ocorreram na distinção entre os gestos 'A' e 'E', o que aponta para a similaridade de suas características visuais na representação do sistema de LIBRAS como a principal fonte de falha.

Tabela 6.3: Matriz de confusão (n=50 testes)

Real/Previsto	A	E	I	O	U
A	8	2	0	0	0
E	1	9	0	0	0
I	0	0	10	0	0
O	0	0	0	10	0
U	0	0	0	0	10

2. Atuação Mecânica: A atuação mecânica dos servomotores em conjunto na formação de cada um dos 5 gestos mapeados na Prova de Conceito, e avaliada pela precisão angular, é apresentada na Tabela 6.4. As métricas de Ângulo Alvo, Erro Médio e Precisão foram utilizadas para quantificar a capacidade do sistema em reproduzir os gestos com fidelidade. O Ângulo Alvo representa a posição angular definida para cada gesto. O Erro Médio foi estimado através de observações visuais e calibrações manuais, refletindo a variação típica entre a posição desejada e a alcançada. Já a Precisão indica o quão próximo o movimento realizado se aproxima do ideal.

Com base nas informações fornecidas, a Tabela 6.4 detalha os ângulos de cada um dos seis servomotores para a criação dos gestos das cinco vogais. O Dedo 0 (Polegar), Dedo 1 (Indicador), Dedo 2 (Médio), Dedo 3 (Anelar) e Dedo 4 (Mínimo) são controlados por um servo motor cada. Adicionalmente, o Servo 5 (Giro do Pulso) é utilizado para o gesto da vogal 'O'.

Tabela 6.4: Matriz de movimentação sistêmica angular dos 6 servomotores para comandar ações tanto sobre os 5 dedos como sobre giro da base do punho no eixo longitudinal.

Vogal	Servo 0 (Polegar)	Servo 1 (Indicador)	Servo 2 (Médio)	Servo 3 (Anelar)	Servo 4 (Mínimo)	Servo 5 (Giro Pulso)
A	5°	160°	160°	160°	150°	0°
E	90°	160°	160°	160°	150°	0°
I	90°	160°	160°	160°	5°	0°
O	45°	90°	90°	90°	90°	90°
U	90°	5°	5°	160°	160°	0°

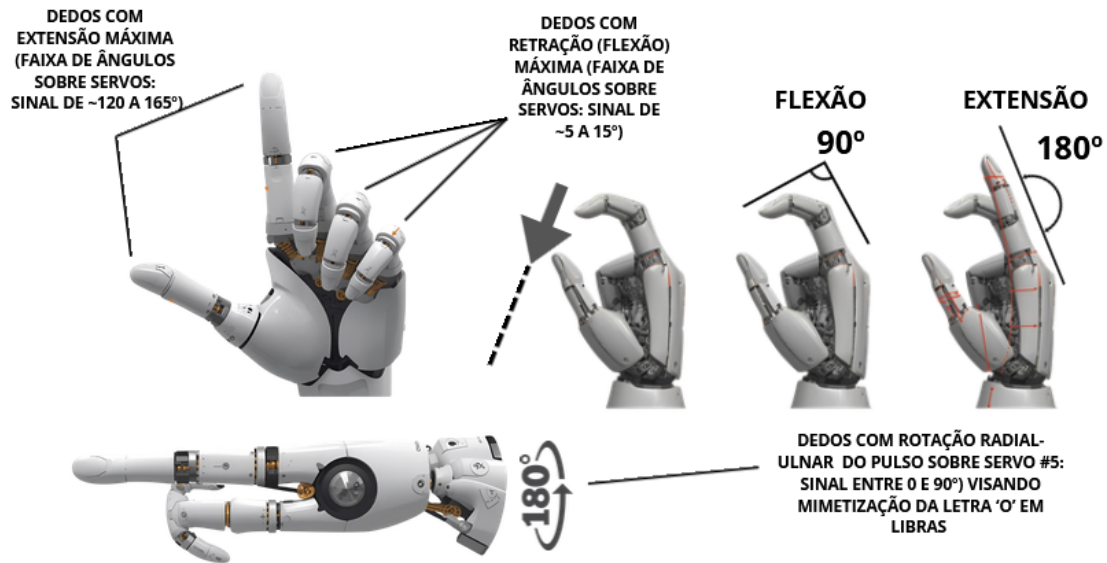


Figura 6.14: Ilustrações de flexão de dedos e pulso em resposta a pulsos PWM.

6.3.4 Análise Temporal Detalhada

O desempenho temporal é um fator crítico em aplicações em tempo real como esta. Uma análise detalhada do tempo gasto em cada etapa do pipeline foi conduzida para identificar gargalos e oportunidades de otimização. A Tabela 6.5 descreve a contribuição de cada componente para a latência total.

Como demonstrado, o estágio de processamento do MediaPipe representa a maior fatia da latência total (40,8%), o que é esperado devido à complexidade computacional da extração de pontos-chave das mãos. O segundo maior contribuinte é a classificação do gesto, que, embora mais leve, ainda consome uma parcela significativa do tempo. A transmissão WiFi, apesar de ser uma etapa de comunicação, mostra-se eficiente com apenas 17,8% da latência, mas ainda representa um ponto a ser considerado para futuras otimizações. A atuação dos servomotores é a etapa mais rápida, indicando a resposta mecânica do sistema é ágil.

Tabela 6.5: Latência por estágio (ms)

Estágio	Tempo	% Total
Captura de imagem	$2,1 \pm 0,3$	13,4%
Processamento MediaPipe	$6,4 \pm 0,7$	40,8%
Classificação do gesto	$3,2 \pm 0,5$	20,4%
Transmissão WiFi	$2,8 \pm 0,4$	17,8%
Atuação servomotores	$1,2 \pm 0,2$	7,6%

6.3.5 Limitações e Melhorias

Tabela 6.6: Análise SWOT do sistema

FORÇAS	FRAQUEZAS
Baixo custo	Sensível a variações luminosas
Baixa latência	Erros entre gestos similares
Razoável precisão mecânica	Calibração manual necessária
Inércia controlada	Picos de erraticidade em corrente
OPORTUNIDADES	AMEAÇAS
Otimização do pipeline	Limitações do MediaPipe
Adoção de feedback háptico (tátil)	Interferência WiFi
Otimização do pipeline	Limitações do MediaPipe
Possibilidades com novos classificadores para captura de imagem	Riscos com Back-Powering (Ricochete de Alimentação Fantasma)

6.4 Resumo do Capítulo

Este capítulo apresentou a avaliação experimental do sistema proposto para reconhecimento gestual e controle remoto de uma mão robótica. O desenvolvimento seguiu uma abordagem incremental, dividida em quatro fases principais:

- **FASE 1:** Prototipação inicial com diversos materiais para simulação da anatomia manual, testando flexão e retração dos dedos.
- **FASE 2:** Aprimoramento do modelo com estrutura em madeira MDF e sistema de molas, ainda com limitações nos movimentos de retorno.
- **FASE 3:** Consolidação do protótipo utilizando material DIPROM/XPS e cordas de aço de violão, com desenvolvimento de circuitos dedicados de alimentação baseados em conversores buck DC-DC XL4015 para maior estabilidade operacional.
- **FASE 4:** Integração final com sistema cliente-servidor para comunicação via Wi-Fi entre os módulos de visão computacional e controle robótico.

O escopo experimental focou nos cinco gestos de vogais em LIBRAS (A, E, I, O, U), permitindo uma validação controlada da arquitetura distribuída. O protótipo final incorpora seis servomotores (cinco para os dedos e um para rotação do pulso), circuitos estabilizadores de energia e módulo ESP32 para comunicação sem fio.

Os resultados demonstraram a viabilidade técnica do sistema proposto, com capacidade de interpretar sinais gestuais em tempo real e reproduzi-los precisamente no dispositivo robótico, estabelecendo as bases para futuras expansões do vocabulário de gestos reconhecidos.

Capítulo 7

Conclusão e Trabalhos Futuros

Nesta dissertação, apresentamos uma arquitetura de robótica assistiva para a reprodução biomimética de gestos manuais das cinco vogais da Língua Brasileira de Sinais (LIBRAS). A pesquisa propôs uma prova de conceito para traduzir sinais visuais captados por uma câmera em movimentos físicos executados por uma mão robótica de cinco dedos. O sistema incumbiu-se de atestar a viabilidade de um método que permitisse executar tal tarefa sem a necessidade de dispositivos vestíveis ou sensores mecânicos, oferecendo uma alternativa econômica e simplificada. Para tanto, a abordagem utilizou tecnologias como Internet das Coisas (IoT), Sistemas Embarcados e Rede de Computadores para a teleoperação de um servomotor experimental.

Concomitantemente, o estudo buscou abordar a necessidade de movimentar um manipulador robótico de mão sem a dependência de tecnologias de contato, com foco em fatores como baixo custo, facilidade de implementação e acessibilidade. Por meio de um mapeamento sistemático do estado da arte, foram identificadas lacunas e oportunidades, proporcionando uma visão comparativa das metodologias existentes. O objetivo foi oferecer uma alternativa eficiente e acessível para o avanço da pesquisa em manipulação robótica teleoperável de mãos por meio de uma técnica de interface não vestível.

7.1 Limitações

Durante o desenvolvimento deste trabalho, identificaram-se diversos desafios técnicos que delimitaram a implementação e o desempenho do protótipo. A instabilidade na alimentação elétrica dos servomotores foi um obstáculo significativo, resultando em ruído e movimentos imprecisos durante a flexão e retração dos dedos. Essa dificuldade exigiu uma análise aprofundada e o uso de circuitos **Step-Down converter** e atenuadores de ruído para garantir a estabilidade da corrente.

A precisão angular dos servomotores também representou uma limitação, impactando a sutileza necessária para reproduzir fielmente os sinais de LIBRAS. Além disso, a mecânica de tração dos dedos foi um

ponto de grande desafio, resultando em fracassos na fase de testes com protótipos que utilizavam êmbolos de seringas de água e molas de retração (retiradas de retrovisores de carro). A seleção de materiais para os fios e a fixação demandou um processo de tentativa e erro para alcançar a tensão e o alinhamento adequados.

Outras limitações significativas que não foram abordadas ou desenvolvidas no decorrer deste trabalho foram a usabilidade e controle de segurança no transporte e transmissão dos pacotes de dados dentro do ambiente cliente-servidor wifi do protótipo.

Desta forma, no quesito relacionado aos testes de usabilidade, o desenvolvimento deste trabalho não realizou simulações controladas envolvendo usuários pertencentes a nenhum grupo específico de surdos-mudos. Tal reticência experimental, em eventuais abordagens futuras do trabalho, aponta para uma necessidade de posteriores aprimoramentos que adêquem a usabilidade apropriada ao seu principal grupo de usuários-fim.

Por sua vez, considerando que dados e acesso ao ambiente do sistema desenvolvido se fundamentam no uso de meios de acesso remoto pela Web, o não tratamento da questão de segurança na transmissão dos pacotes de dados REST a serem enviados pela página Web representa uma limitação e um fator que precisa ser considerado e revisitado, levando-se em conta que se optou pela viabilização da transmissão com o uso do método GET HTTP, visando a facilitação nas experimentações.

Por fim, o treinamento e os testes iniciais do modelo, utilizando bibliotecas internas do **Django**, apresentaram instabilidade. Embora essas dificuldades tenham sido superadas com a evolução do projeto, elas foram importantes no processo de desenvolvimento.

7.2 Contribuições alcançadas

Este trabalho focalizou avanços na área de robótica assistiva, contribuindo com uma nova abordagem para a avaliação robótica integrada à computação visual. A solução tecnológica desenvolvida e os ambientes de tarefas são os principais produtos deste estudo, oferecendo um método viável e simples de telecinemática assistiva e o ensejo para aprimoramento da tecnologia, um sistema embarcado útil a tal campo de aplicação prática.

Considerando-se tal conquista, compreende-se que alguns aspectos do processo de desenvolvimento da concepção desta dissertação podem ser identificados como contribuições, tais como: a ampliação do conhecimento teórico com o apoio de *frameworks* no campo de tecnologias assistivas por meio de sistemas embarcados, bibliotecas e recursos computacionais gratuitos que se mostram promissores para acréscimo ao estado da arte atual; a produção de respostas e resultados das experimentações com o método de movimentos por reconhecimento gestual em visão robótica que referenciam e indicam a viabilidade do método; o enriquecimento no estado da arte das metodologias de Visão Robótica que podem propor novas facilidades e acelerações na produção de aplicações assistivas; a expansão de conhecimento na transversalidade e interdisciplinaridade do uso de Visão Robótica, Desenvolvimento **Web Full Stack** e Sistemas Embarcados; a análise de estabilidade de alimentação elétrica para sustentar e manter o movimento dos dedos, com o uso de

circuitos **Step-Down DC-DC Converter** e atenuadores de ruídos na entrada de sinais dos servomotores; a compreensão maior sobre aspectos de anatomia humana da mão para o aprimoramento de protótipos que imitem a gestualidade e a sutileza da formação de sinais de LIBRAS; e, finalmente, tanto a versatilidade de comunicação Wi-Fi como a facilidade e uso de módulos ESP32-WROOM na integração e sinergia de sistemas embarcados IoT com aplicações que solicitem uso de Computação Visual.

7.3 Trabalhos futuros

Como trabalhos futuros, consideram-se como campos de expansão da pesquisa até aqui atingida, novos modelos que incluam a substituição dos atuais servomotores por elementos de IoT mais sofisticados e precisos, a fim de aprimorar a fidelidade dos movimentos.

Assim sendo, observa-se que a tecnologia assistiva descrita neste trabalho também abre espaço para diversas possibilidades de aplicações do método em campos tão diversos como a telemedicina, a cinemática, ou as engenharias mecânicas e mecatrônicas, assim como também em sistemas de controle de tráfego, energia, transporte e logística.

Além disso, considerando-se ainda a perspectiva de aprimoramentos do trabalho em foco, a pesquisa pode doravante se expandir em duas direções significativas, que refletem as tendências da área e a correspondente demanda por sistemas mais intuitivos e acessíveis:

- **Sistemas de controle baseados em realidade aumentada e virtual:** Em um futuro e próximo passo para esta pesquisa, uma evolução natural seria a substituição dos atuadores físicos por efetadores virtuais, como os encontrados em sistemas holográficos e 3D. Isso permitiria a manipulação de objetos e ambientes virtuais, abrindo portas para aplicações em telemedicina e treinamento, onde o usuário poderia interagir com o sistema sem a necessidade de um braço robótico físico, utilizando apenas dispositivos como óculos de realidade virtual.
- **Aprimoramento da comunicação e portabilidade:** Para garantir o desempenho em tempo real, um dos pontos-chave do trabalho, considera-se essencial explorar novos protocolos de comunicação, como MQTT e WebSocket, que se proponham a oferecer maior estabilidade e menor latência. Além disso, a integração com dispositivos portáteis, como smartphones, permitiria que o sistema fosse mais acessível e flexível, utilizando a entrada de vídeo desses aparelhos como fonte de dados para o controle, viabilizando o uso em ambientes cotidianos.

Portanto, ao invés de se focar na substituição dos componentes robóticos, a pesquisa futura poderá - fazendo uso deste trabalho como base - migrar para um paradigma alternativo em termos de sistema assistivo que substitua a camada final física da mão robótica por um efetador virtual, preservando-se o uso da solução de captura obtida por meio da aplicação de *machine learning* desenvolvida neste trabalho.

Apêndices

Vídeo com sumário de apresentação do trabalho

Endereço URL do video MP4 (Vide Figura 7.1). Tempo: 3min36seg



Figura 7.1: *URL e QR-Code para video de apresentação do trabalho*

Glossário

A. Detecção de Objetos

1. **YOLO (You Only Look Once):** Método de detecção de objetos em tempo real com alta velocidade e precisão. Aplicações incluem vigilância de segurança e veículos autônomos.

B. Classificação de Imagem

1. **Redes Neurais Convolucionais (CNNs):** Técnica de classificação precisa de imagens usando aprendizado profundo. Usada para reconhecimento de objetos em fotos.
2. **Classificação de Nível de Pixel - Segmentação por Instância:** Método de segmentação precisa de objetos individuais, com aplicação em identificação de objetos em imagens médicas.
3. **Random Forest:** Método de aprendizado de máquina versátil que pode ser usado para classificação e regressão. Ele funciona construindo múltiplas árvores de decisão durante o **treinamento** e produzindo a classe que é a moda (mais frequente) das classes das árvores individuais.
4. **Classificadores:** Algoritmos de aprendizado de máquina usados para categorizar dados. Recebem um conjunto de dados de entrada e atribuem uma etiqueta ou classe a eles. No contexto de imagens, um classificador pode ser treinado para determinar se uma foto contém um gato ou um cachorro, por exemplo.
5. **Treinamento:** Fase em que um modelo de aprendizado de máquina, como um **classificador**, é exposto a um conjunto de dados conhecido para aprender os padrões e as regras necessárias para fazer previsões precisas. Durante o **treinamento**, o modelo ajusta seus parâmetros internos para minimizar erros, melhorando seu desempenho em dados futuros.

C. Detecção de Humanos em Cenas Visuais

1. **OpenPose:** Método de rastreamento preciso de movimento e análise de postura humana. Usado em interação Humano-Máquina e controle de dispositivos.

D. Interação Humano-Máquina Baseada em Gestos

1. **Reconhecimento de Gestos:** Técnica de interação intuitiva e natural com identificação de gestos. Aplicado em controle de dispositivos e jogos.

E. Colaboração Multimodal de Máquinas

1. **Fusão de Dados Multimodais:** Abordagem que compreende holisticamente o ambiente, integrando informações de sensores. Aplicações em robótica, assistência médica e veículos autônomos.

F. Reconhecimento de Ação

1. **LSTM (Long Short-Term Memory):** Modelo para modelagem de sequências de ações e reconhecimento de padrões temporais em movimentos e atividades humanas.

G. Reconhecimento de Gestos das Mãos

1. **CNNs para Reconhecimento de Gestos das Mãos:** Método de interação precisa com gestos e detecção de gestos específicos para controle de interfaces e robôs.

H. Visão Robótica

1. **Sensor RGB-D:** Sensor versátil de percepção tridimensional do ambiente usado em robótica para mapeamento e navegação.
2. **Algoritmos de Suporte:** Tradução de imagens em sinais de controle com adaptação a diferentes cenários, usado para orientação e controle de robôs.
3. **Manipuladores:** Robôs projetados para tarefas de manipulação física, como movimentar, agarrar e interagir com objetos no ambiente, amplamente utilizados em aplicações industriais e de logística.
4. **Autores:** Indicam os pesquisadores ou equipes responsáveis pelas contribuições na área de controle de robôs manipuladores.
5. **Gestos Manuais:** Controle de robôs por meio de gestos manuais, permitindo interações intuitivas e naturais entre humanos e robôs manipuladores.
6. **Rotação do Braço:** Utilização do controle de rotação do braço para fins de navegação visual de robôs móveis em ambientes desconhecidos.
7. **Abertura/Fechamento da Garra:** Controle da abertura e fechamento da garra robótica, aplicado em linhas de montagem industrial e cirurgia de intervenção colaborativa.
8. **Posicionamento de Objetos na Palma da Mão Aberta:** Técnica que permite o posicionamento preciso de objetos usando reconhecimento de gestos em ambientes de fabricação.

9. **Retorno à posição quando a Palma da Mão está fechada:** Controle de posição de braço robótico para manipulação de objetos 3D em espaço cartesiano.
10. **Seleção de Objetos através do Ato de Apontar:** Método de seleção de objetos por meio de gestos de apontar.
11. **Rastreamento de Alta Velocidade:** Técnica de rastreamento de objetos em imagens para aplicações de visão por computador.
12. **Combinação de Gestos com Movimentos Corporais:** Integração de gestos com movimentos corporais para o controle de robôs manipuladores.
13. **Ações Colaborativas:** Controle de robôs com base em instruções naturais via sensações visuais e táteis, possibilitando a colaboração homem-máquina.
14. **Captação de um Objeto:** Técnica que envolve a captura de um objeto selecionado por um gesto de apontar da mão de um humano e soltura do objeto na palma da mão da pessoa.

I. Visão Baseada em Reconhecimento de Gestos Manuais

1. **Fundo Complexo:** Problema relacionado à interferência do fundo nas imagens usadas para o reconhecimento de gestos.
2. **Seleção de Features:** Desafio na escolha das características mais discriminativas para representar gestos.
3. **Rastreamento de Mão em Tempo Real:** Técnica que permite o rastreamento de mãos em tempo real, combinando informações de movimento, cor da pele e borda.
4. **Coarticulação:** Fenômeno em que gestos não são produzidos de forma isolada, mas sim em contexto contínuo, complicando o reconhecimento de gestos.
5. **Epêntese de Movimento (ME):** Inserção de movimentos não intencionais durante a execução de um gesto principal, que pode dificultar o reconhecimento de gestos.
6. **Detector de Sinais Curtos:** Componente que auxilia a detectar sinais curtos em sequências contínuas de gestos.
7. **Modelo de Limiar Baseado em HMM:** Abordagem que utiliza um modelo baseado em HMM para classificar sequências de sinais como sinais pré-treinados ou ME.

J. Reconhecimento de Gestos Baseado em Esqueletos

1. **Esqueleto:** Uma representação gráfica das articulações e partes do corpo humano, usada para rastrear e descrever os movimentos das pessoas em um ambiente.
2. **Descritores Artesanais:** Características específicas projetadas manualmente para representar dados, neste caso, informações de articulações da mão em gestos.
3. **Codificação do Vetor de Fisher:** Uma técnica de codificação de características que descreve a distribuição de gradientes em uma imagem ou conjunto de dados.
4. **Pirâmide Temporal:** Uma abordagem que considera informações de vários níveis de granularidade temporal para a análise de gestos.
5. **SVM Linear:** Máquina de Vetor de Suporte (SVM) é um algoritmo de aprendizado de máquina, e a versão linear refere-se a um SVM que cria uma fronteira de decisão linear para classificação.
6. **Redes Neurais Recorrentes (RNN):** Um tipo de rede neural projetado para lidar com sequências de dados, como séries temporais, por serem capazes de manter estados anteriores ao processar novas entradas.
7. **RNN Bidirecional:** Uma RNN que processa sequências de dados em ambas as direções (passado para o futuro e futuro para o passado), capturando informações contextuais de ambas as direções.
8. **LSTM (Long Short-Term Memory):** Um tipo de unidade de rede recorrente que ajuda a resolver o problema do gradiente fenecente, permitindo que informações relevantes sejam armazenadas e recuperadas em sequências longas.
9. **Rede Convolutacional (CNN):** Uma arquitetura de rede neural projetada para processar dados com estrutura de grade, como imagens, usando camadas convolucionais.
10. **Pose 3D:** A representação tridimensional da posição e orientação de um objeto ou parte do corpo humano, neste caso, a pose da mão.
11. **Convolução de Travessia:** Um tipo de operação de convolução em redes neurais que ajuda a processar sequências temporais ou tridimensionais.
12. **Mecanismo de Atenção:** Um componente em redes neurais que permite que a rede se concentre em partes específicas dos dados de entrada.
13. **Rede Neural Convolutacional Paralela (CNN):** Uma arquitetura de rede neural que processa várias sequências de dados em paralelo usando camadas convolucionais.

14. **MLP (Multilayer Perceptron):** Um tipo de rede neural feedforward com várias camadas ocultas entre a entrada e a saída.
15. **Aprendizado Profundo:** Uma abordagem de aprendizado de máquina que utiliza redes neurais com muitas camadas para aprender características complexas dos dados.
16. **Autoatenção:** Uma técnica que permite que uma rede neural atribua diferentes pesos a partes diferentes dos dados de entrada.
17. **Aprendizado de Variedade SPD (Simétrica e Definida Positiva):** Um método de aprendizado que lida com matrizes simétricas definidas positivas para representar gestos.
18. **Matrizes SPD:** Matrizes simétricas definidas positivas usadas para representar informações espaciais e temporais em gestos.
19. **Fluxos 3D e 2D de CNN:** Abordagem que utiliza redes neurais convolucionais em duas dimensões (2D) e três dimensões (3D) para analisar gestos com foco na postura e movimento da mão.
20. **Arquitetura Complexa:** Uma rede neural com muitas camadas e componentes, geralmente exigindo mais recursos de treinamento e implantação.
21. **Modelagem de Contexto:** Processo de consideração de informações contextuais para melhorar o reconhecimento de gestos, especialmente quando os gestos estão sujeitos a coarticulação ou epêntese de movimento.
22. **Análise de Sequências:** Técnicas que envolvem a análise de sequências temporais de gestos para identificar e filtrar gestos de epêntese de movimento durante a interpretação.

Referências Bibliográficas

- Ahmed, S. M. H., Alexander, T. C. e Anagnostopoulos, G. C. (2008), Real-time, static and dynamic hand gesture recognition for human-computer interaction.
- Arenas, J. O. P., Beleno, R. D. H. e Moreno, R. J. (2017), Deep convolutional neural network for hand gesture recognition used for human-robot interaction, *Journal of Engineering and Applied Sciences* 12(11), 9278–9285.
- Bajaj, Y. e Malhotra, P. (2023), American sign language identification using hand trackpoint analysis, *International Journal of Computer Applications* 185(48), 1–5.
- Barker, N. e Jewitt, C. (2022), Future touch in industry: Exploring sociotechnical imaginaries of tactile (tele) robots, *Futures* 136, 102885.
- Bekey, G. A. (2005), *Autonomous robots: From biological inspiration to implementation and control*, MIT press.
- Bonin-Font, F., Ortiz, A. e Oliver, G. (2008), Visual navigation for mobile robots: A survey, *Journal of Intelligent and Robotic Systems* 53(3), 263–296.
- Cao, Z., Simon, T., Wei, S. e Sheikh, Y. (2017), Realtime multi-person 2d pose estimation using part affinity fields, *in* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR’17).
- Cascone, L., Nappi, M., Narducci, F. e Passero, I. (2021), Dtpaal: Digital twinning pepper and ambient assisted living, *IEEE Transactions on Industrial Informatics* 18(2), 1397–1404.
- Chen, M., Wang, S., Liang, P. P., Baltrušaitis, T., Zadeh, A. e Morency, L.-P. (2017), Multimodal sentiment analysis with word-level fusion and reinforcement learning, *in* Proceedings of the 19th ACM International Conference on Multimodal Interaction, ICMI ’17, Association for Computing Machinery, New York, NY, USA, p. 163–171. URL : <https://doi.org/10.1145/3136755.3136801>
- Chen, S. Y. (2011), Kalman filter for robot vision: A survey, *IEEE Transactions on Industrial Electronics* 59(11), 4409–4420.

- Chen, X., Guo, H., Wang, G. e Zhang, L. (2017), Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition, *in* Proceedings of the International Conference on Image Processing (ICIP), Beijing, China, Sep. 2017, pp. 2881–2885.
- Choudhary, G. B. e Chethan, R. B. V. (2015), Real-time robotic arm control using hand gestures, *in* Proceedings of the 2014 International Conference on High-Performance Computing and Applications, ICHPCA 2014.
- De Smedt, Q., Wannous, H. e Vandeborre, J.-P. (2016), Skeleton-based dynamic hand gesture recognition, *in* Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Las Vegas, NV, United States, Jun. 2016, pp. 1206–1214.
- Delligatti, L. (2013), *SysML Distilled: A Brief Guide to the Systems Modeling Language*, 1st edn, Addison-Wesley Professional.
- Devineau, G., Moutarde, F., Xi, W. e Yang, J. (2018), Deep learning for hand gesture recognition on skeletal data, *in* Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition (FG), Xi'an, China, May 2018, pp. 106–113.
- Fareed, M. M. F. M., Akram, Q. I., Anees, S. B. A. e Fakhri, A. H. (2015), Gesture-based wireless single-armed robot in cartesian 3d space using kinect, *in* Proceedings of the 2015 5th International Conference on Communication Systems and Network Technologies (CSNT'15).
- Felizardo, H. S., Guimarães, L. J. M., Teixeira, H. M. P., Souza, R. C. T., Aylon, L. B. R., Costa, Y. M. G. e Thomaz, R. (2023), Machine learning for gesture recognition using leap motion controller, *Proceedings of the 24th Brazilian Symposium on Computer Human Interaction (IHC)*. Available from the State University of Maringá and Federal University of Paraná.
- Fong, T., Rochlis Zumbado, J., Currie, N., Mishkin, A. e Akin, D. L. (2013), Space telerobotics: Unique challenges to human–robot collaboration in space, *Reviews of Human Factors and Ergonomics* 9(1), 6–56.
- Fontalis, A., Raj, R. D., Kim, W. J. *et al.* (2023), Functional implant positioning in total hip arthroplasty and the role of robotic-arm assistance, *International Orthopaedics (SICOT)* 47, 573–584. URL : <https://doi.org/10.1007/s00264-022-05646-0>
- He, L., Yan, Z. e Atiquzzaman, M. (2018), Lte/lte-a network security data collection and analysis for security measurement: A survey, *IEEE Access* 6, 4220–4242.
- Henriques, J. F., Caseiro, R., Martins e Batista, J. (2015), High-speed tracking with kernelized correlation filters, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(3), 583–596.

- Hochreiter, S. e Schmidhuber, J. (1997), Lstm can solve hard long time lag problems, *in* Advances in Neural Information Processing Systems 9, MIT Press, Cambridge, MA. Presented at NIPS 96.
- Hou, J., Wang, G., Chen, X., Xue, J.-H., Zhu, R. e Yang, H. (2018), Spatial-temporal attention res-tcn for skeleton-based dynamic hand gesture recognition, *in* Proceedings of the European Conference on Computer Vision Workshops (ECCVW), Munich, Germany, Sep. 2018, pp. 273–286.
- Huang, M. J., Liang, J. L., Wang, H., Kang, L., Deng, Y. H. e Wang, J. P. (2011), Laparoscopic-assisted versus open surgery for rectal cancer: a meta-analysis of randomized controlled trials on oncologic adequacy of resection and long-term oncologic outcomes, *International Journal of Colorectal Disease* 26, 415–421.
- Ikai, T., Kamiya, S. e Ohka, M. (2016), Robot control using natural instructions via visual and tactile sensations, *Journal of Computer Science* 12(5), 246–254.
- Jaimes, A. e Sebe, N. (2007), Multimodal human–computer interaction: A survey, *Computer Vision and Image Understanding* 108(1-2), 116–134.
- Karim, E., Nambiappan, H. R., Acharya, S. e Makedon, F. (2023), Remote operated human robot interactive system using hand gestures for persons with disabilities, *in* Proceedings of the 16th International Conference on Pervasive Technologies Related to Assistive Environments (PETRA '23), Association for Computing Machinery, New York, NY, USA, pp. 137–139.
- Khan, M. R., Sunny, S. H., Ahmed, T., Shahria, T. I., Modi, P. P., Zarif, I. I., Ahamed, J. S. I., Ahmed, H. U., Ketchum, E. M. e Rahman, M. H. (2023), Development of a robot-assisted telerehabilitation system with integrated iiot and digital twin, *IEEE Xplore Access* . URL : <https://doi.org/10.1109/ACCESS.2023.3291803>
- Khoshelham, K. e Elberink, S. O. (2012), Accuracy and resolution of kinect depth data for indoor mapping applications, *Sensors (Basel, Switzerland)* 12, 1437–1454.
- Lalonde, J. F., Vandapel, N., Huber, D. F. e Hebert, M. (2006), Natural terrain classification using three-dimensional radar data for ground robot mobility, *Journal of field robotics* 23(10), 839–861.
- LeCun, Y., Bottou, L., Bengio, Y. e Haffner, P. (1998), Gradient-based learning applied to document recognition, *Proceedings of the IEEE* 86(11), 2278–2324.
- Lehman, A. C., Berg, K. A., Dumpert, J., Wood, N. A., Visty, A. Q., Rentschler, M. E., Platt, S. R., Farritor, S. M. e Oleynikov, D. (2008), Surgery with cooperative robots, *Computer Aided Surgery* 13(2), 95–105. PMID: 18317958. URL : <https://doi.org/10.3109/10929080801956706>
- Lima, B., Júnior, G. L. N., Amaral, I., Vieira, T., Ferreira, B. e Vieira, T. (2019), Real-time hand pose tracking and classification for natural human-robot control, *in* Proceedings of the 14th International Joint

- Conference on Computer Vision, Imaging, and Computer Graphics Theory and Applications (VISIGRAPP 19), Vol. 5.
- Liu, H. e Wang, L. (2018), Gesture recognition for human-robot collaboration: A review, *International Journal of Industrial Ergonomics* 68, 355–367. URL : <https://www.sciencedirect.com/science/article/pii/S0169814117300690>
- Liu, J., Liu, Y., Wang, Y., Prinet, V., Xiang, S. e Pan, C. (2020), Decoupled representation learning for skeleton-based gesture recognition, *in* Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR).
- Liu, J., Liu, Y., Wang, Y., Prinet, V., Xiang, S. e Pan, C. (2021), Han: An efficient hierarchical self-attention network for skeleton-based gesture recognition.
- Luo, J., Liu, W., Qi, W., Hu, J., Chen, J. e Yang, C. (2023), A vision-based virtual fixture with robot learning for teleoperation, *Robotics and Autonomous Systems* 164, 104414.
- Luo, X., Amighetti, A. e Zhang, D. (2019), A human-robot interaction for a mecanum wheeled mobile robot with real-time 3d two-hand gesture recognition, *Journal of Physics: Conference Series* 1267.
- Martin, J. B. e Moutarde, F. (2019), Real-time gestural control of robot manipulator through deep learning human-pose inference, *in* Computer Vision Systems. Lecture Notes in Computer Science, Vol. 11754, Springer International Publishing.
- Moh, J. J., Kijima, T., Zhang, B. e Lim, H. O. (2019), Gesture recognition and effective interaction based dining table cleaning robot.
- Nguyen, X. S., Brun, L., Lézoray, O. e Bougleux, S. (2019), A neural network based on spd manifold learning for skeleton-based hand gesture recognition, *in* Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, United States.
- Nunez, J. C., Cabido, R., Pantrigo, J. J., Montemayor, A. S. e Velez, J. F. (2018), Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition, *Pattern Recognition* 76, 80–94.
- Pierson, H. A. e Gashler, M. S. (2017), Deep learning in robotics: A review of recent research, *Advanced Robotics* 31(16), 821–835.
- Quintero, C. P., Tatsambon, R., Gridseth, M. e Jagersand, M. (2015), Visual pointing gestures for bi-directional human-robot interaction in a pick-and-place task, *in* Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication.

- Rautaray, S. S. e Agrawal, A. (2015), Vision-based hand gesture recognition for human-computer interaction: A survey, *Artificial Intelligence Review* 43(1), 1–54.
- Redmon, J., Divvalay, S., Girshick, R. e Farhadi, A. (2016), You only look once: Unified, real-time object detection, <https://arxiv.org/pdf/1506.02640.pdf>.
- Robinson, N., Tidd, B., Campbell, D., Kulić, D. e Peter Corke, P. (2023), Robotic vision for human-robot interaction and collaboration: A survey and systematic review, *J. Hum.-Robot Interact.* 12(1), Article 12. 66 pages.
- Russell, S. J. e Norvig, P. (2016), *Artificial Intelligence: A Modern Approach*, Pearson Education Limited, London.
- Russell, S. e Norvig, P. (2003), *Artificial Intelligence: A Modern Approach*, second edn, Prentice-Hall.
- Schmidt, T. e Fox, D. (2020), Self-directed lifelong learning for robot vision, in N. M. Amato, G. Hager, S. Thomas e M. Torres-Torriti, eds, *Robotics Research*, Springer International Publishing.
- Song, H., Feng, W., Guan, N., Huang, X. e Luo, Z. (2017), Towards robust ego-centric hand gesture analysis for robot control, in *Proceedings of the 2016 IEEE International Conference on Signal and Image Processing (ICSIP'16)*.
- Song, W. *et al.* (2019), Design of a flexible wearable smart semg recorder integrated gradient boosting decision tree based hand gesture recognition, *IEEE Transactions on Biomedical Circuits and Systems* 13(6), 1563–1574.
- Sünderhauf, N., Brock, O., Scheirer, W., Hadsell, R., Fox, D., Leitner, J. e Upcroft, B. (2018), The limits and potentials of deep learning for robotics, *International Journal of Robotics Research* 37(4-5), 405–420.
- Valle, A. F., Alenyà, G., Chance, G., Caleb-Solly, P., Dogramadzi, S. e Torras, C. (2019), Personalized robot assistant for support in dressing, *IEEE Transactions on Cognitive and Developmental Systems* 11(3), 363–374.
- Varshini, A. S., Bhavani, G., Vithya e Thilagavathy, R. (2021), Real-time hand gesture recognition for robotic arm and home automation, in *2021 International Symposium on Electrical, Electronics and Information Engineering*.
- Viola, P. e Jones, M. J. (2004), Robust real-time face detection, *International Journal of Computer Vision* 57, 137–154.
- Weng, J., Liu, M., Jiang, X. e Yuan, J. (2018), Deformable pose traversal convolution for 3d action and gesture recognition, in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, pp. 142–157.

- Xia, Z., Lei, Q., Yang, Y., Zhang, H., He, Y., Wang, W. e Huang, M. (2019), Vision-based hand gesture recognition for human-robot collaboration: A survey, *in* Proceedings of the 2019 5th International Conference on Control, Automation, and Robotics (ICCAR'19).
- Yuh, J., Marani, G. e Blidberg, D. R. (2011), Applications of marine robotic vehicles, *Intelligent Service Robotics* 4, 221–231.
- Zhang, H. B., Zhang, Y. X., Zhong, B., Lei, Q., Yang, L., Du, J. X. e Chen, D. S. (2019), A comprehensive survey of vision-based human action recognition methods, *Sensors* 19(5), 1005.