UNIVERSIDADE FEDERAL DO CEARÁ

CAMPUS DE QUIXADÁ

CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

LUIS GOMES DAMASCENO NETO

SMARTFIGHT: ANÁLISE DE MOVIMENTOS EM VÍDEOS DE TREINAMENTO DE MUAY-THAI COM VISÃO COMPUTACIONAL

QUIXADÁ

2025

LUIS GOMES DAMASCENO NETO

SMARTFIGHT: ANÁLISE DE MOVIMENTOS EM VÍDEOS DE TREINAMENTO DE MUAY-THAI COM VISÃO COMPUTACIONAL

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas de Informação do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Cristiano Bacelar de Oliveira.

QUIXADÁ

2025

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

D162s Damasceno Neto, Luis Gomes.
SMARTFIGHT: ANÁLISE DE MOVIMENTOS EM VÍDEOS DE TREINAMENTO DE MUAY-THAI COM VISÃO COMPUTACIONAL / Luis Gomes Damasceno Neto. – 2025.
65 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Sistemas de Informação, Quixadá, 2025.
Orientação: Prof. Dr. Cristiano Bacelar de Oliveira.

1. Muay Thai. 2. Visão Computacional. 3. Estimação de Pose. 4. FFT. 5. DTW. I. Título.
CDD 005

LUIS GOMES DAMASCENO NETO

SMARTFIGHT: ANÁLISE DE MOVIMENTOS EM VÍDEOS DE TREINAMENTO DE MUAY-THAI COM VISÃO COMPUTACIONAL

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Sistemas de Informação do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Sistemas de Informação.

Aprovada em: 31 de Julho de 2025

BANCA EXAMINADORA

Prof. Dr. Cristiano Bacelar de Oliveira (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Me. Carlos Igor Ramos Bandeira
Universidade Federal do Ceará (UFC)

Prof. Dr. André Ribeiro Braga
Universidade Federal do Ceará (UFC)

AGRADECIMENTOS

À Universidade Federal do Ceará – Campus Quixadá, pela excelência no ensino na área de Sistemas de Informação e pelo apoio institucional e financeiro ao longo destes anos de formação.

Ao Prof. Dr. Cristiano Bacelar de Oliveira, pela receptividade à proposta deste trabalho, pela paciência e pela orientação técnica e acadêmica fundamentais para o desenvolvimento desta pesquisa.

À minha mãe, Veridiana Ferreira dos Santos Batista, por seu apoio incondicional, por sempre estender a mão e abrir o coração para que este sonho se tornasse possível.

Ao meu pai, Luiz Ironésio Gomes, por sua presença constante, por cuidar de mim da melhor forma, mesmo à distância.

À minha futura esposa, Rayssa de Abreu Tavares, por todo o suporte emocional e intelectual, pelas inúmeras revisões de texto, incentivo contínuo e por sua valiosa colaboração na construção do conjunto de dados desta pesquisa.

RESUMO

Este trabalho propõe o desenvolvimento de um sistema para análise de movimentos em vídeos de treinamento de Muay-Thai, utilizando técnicas de Visão Computacional e análise temporal. A solução aplica algoritmos de estimação de pose humana, extraindo coordenadas 2D de articulações por meio do modelo YOLOv11-Pose, a partir de vídeos de atletas executando diferentes golpes. Essas informações são convertidas para arquivos CSV e analisadas com métodos como a Transformada Rápida de Fourier (FFT), para identificar a periodicidade dos movimentos, e o Dynamic Time Warping (DTW), para medir a similaridade entre execuções. O sistema permite identificar ciclos de golpes, analisar seu ritmo e comparar execuções com modelos de referência, sendo uma ferramenta de apoio à avaliação técnica e ao treinamento. Os resultados mostram que o sistema é capaz de segmentar e analisar execuções de golpes com boa precisão, abrindo caminho para aplicações robustas em esportes de combate e instrução assistida por tecnologia.

Palavras-chave: Muay-Thai; Visão Computacional; Estimação de Pose; FFT, DTW; Análise de Movimento

ABSTRACT

This work proposes the development of a system for analyzing movements in Muay Thai training videos using Computer Vision and temporal analysis techniques. The solution employs human pose estimation algorithms to extract 2D joint coordinates from athlete performances using the YOLOv11-Pose model. These data are converted into CSV files and analyzed using methods such as the Fast Fourier Transform (FFT) to identify motion periodicity and Dynamic Time Warping (DTW) to assess similarity between executions. The system enables detection of strike cycles, rhythm analysis, and comparison with reference models, serving as a support tool for technical evaluation and training. Results demonstrate that the system can effectively segment and analyze strike executions, offering potential robust applications in combat sports and technology-assisted instruction.

Keywords: Muay Thai; Computer Vision; Pose Estimation; FFT; DTW; Motion Analysis.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

CVPR	*Computer Vision and Pattern Recognition Conference*
DTW	*Dynamic Time Warping*
FFT	Transformada Rápida de Fourier
FPS	*Frames Per Second*
IA	Inteligência Artificial
KNN	*K-Nearest Neighbors*
LDCRF	*Latent-Dynamic Conditional Random Field*
MS-COCO	*Microsoft Common Objects in Context*
OCR	*Optical Character Recognition*

SUMÁRIO

1 INTRODUÇÃO

Tecnologias que envolvem Processamento de Imagens e Visão Computacional estão hoje integradas a vários campos como indústria, astronomia, esportes, medicina, trânsito, e qualquer outra área que possua regra de negócio (Gonzalez; Woods, 2000). As áreas da Tecnologia da Informação ganham espaço a cada dia, com as inovações produzidas à medida que as ferramentas avançam em poder computacional. A Visão Computacional é um campo dentro da Ciência da Computação, que é relativamente novo, mas tem evoluído constantemente desde os seus experimentos no final dos anos 50 (Ballard; Brown, 1982). Essa área está em constante expansão baseada em métodos estatísticos, geométricos e algoritmos de aprendizado de máquina para aprimorar e expandir técnicas de análise de imagens e vídeos. Cobrindo tarefas como localização de câmera, estimação de pose, detecção, reconhecimento e rastreamento de objetos (Solem, 2012).

O reconhecimento da postura no setor de esportes é um tópico de pesquisa complexo e esteve recebendo bastante atenção nos últimos anos, onde foram desenvolvidos vários métodos para resolver essa questão (Niu, 2024). O cenário das artes marciais envolve várias entidades, desde civis iniciantes a atletas profissionais, que diariamente buscam aperfeiçoar suas técnicas de combate. É comum que durante a era da informação as pessoas tenham a necessidade de usar a tecnologia para aperfeiçoar suas habilidades esportivas (Drumond, 2011). Analisar a postura dos atletas com precisão é fundamental para otimizar treinamento, elevar o desempenho competitivo e prevenir lesões (Niu, 2024).

A modalidade abordada nesta pesquisa é o Muay Thai. Esta arte marcial surgiu na Tailândia e também é conhecida como boxe tailandês ou também a arte das oito armas, caracterizada por utilizar golpes com os punhos, cotovelos, joelhos, canelas (Santos *et al.*, 2021). Os praticantes desse esporte se submetem a treinamentos de alta intensidade, caso não for conduzido e orientado adequadamente pode causar lesões graves (Vicente *et al.*, 2016).

A análise automática dos movimentos é capaz de auxiliar o praticante durante as sessões de treinamento. Isso pode ser realizado com a técnica de estimação de pose humana, que é capaz de extrair informações detalhadas sobre a posição e movimento dos competidores através da detecção de pontos-chave referentes às articulações do corpo (Mendes-Neves *et al.*, 2023; Vicente *et al.*, 2016).

Um exemplo prático de aplicação de Visão Computacional em esportes de combate é o DeepStrike (Jabbr.ai, 2024). Esta ferramenta introduziu uma inteligência artificial voltada

especificamente para esportes de combate. O sistema utiliza técnicas de visão computacional para detectar lutadores de boxe em vídeos, gerando estatísticas detalhadas sobre os combates.

Além disso, a pesquisa de Vicente *et al.* (2016) propõe uma análise de movimentos de atletas de alto rendimento, na modalidade Taekwondo. A metodologia proposta utiliza a filtragem de pontos-chave das poses, reduzindo a quantidade de dados necessários para o treinamento de modelos. Após segmentar os movimentos, o estudo apresenta três etapas principais: extração de pontos-chave, rotulação e filtragem desses pontos-chave, e treinamento de um modelo discriminativo. A abordagem utilizada por Vicente *et al.* (2016) alcançou uma taxa de reconhecimento mais alta e uma segmentação mais precisa em comparação com técnicas convencionais.

Esta pesquisa visa desenvolver um sistema de análise de golpes de Muay Thai a partir de vídeos, combinando técnicas de Visão Computacional e modelos probabilísticos para extrair, analisar padrões de movimento e verificar similaridade entre amostras. O estudo propõe utilizar um conjunto de dados próprio contendo execuções de golpes como jab, direto, cruzado, gancho, chutes (alto, baixo, frontal) e joelhadas. Com base nesses dados, o sistema será capaz de reconhecer e categorizar o golpe executado, analisando a sequência de movimentos registrada em vídeo.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo geral deste trabalho é desenvolver um sistema capaz de extrair informações a partir da detecção de pontos corporais de atletas em vídeos contendo execuções de golpes de Muay Thai.

1.1.2 Objetivos Específicos

1. Analisar o comportamento dos golpes no tempo.
2. Analisar e obter o período em que um golpe é executado.
3. Analisar a similaridade da execução de um golpe em relação a um modelo de referência.

1.2 Organização

Neste trabalho, os principais tópicos abordados são: Visão Computacional, estimação de poses humanas, técnicas de aprendizagem supervisionada e um contexto geral sobre artes marciais. O Capítulo 2 apresenta um breve resumo dos conceitos fundamentais que embasam o desenvolvimento deste estudo. No Capítulo 3, são apresentados os trabalhos relacionados, bem como um quadro comparativo entre essas pesquisas, destacando aspectos relevantes em relação aos objetivos deste trabalho. O Capítulo 4 descreve detalhadamente as etapas da aplicação e as ferramentas utilizadas no projeto de pesquisa. O Capítulo 5 expõe os resultados obtidos a partir da execução da metodologia proposta. Por fim, o Capítulo 6 conclui este trabalho, sintetizando os principais pontos da pesquisa, os resultados alcançados, as contribuições geradas e as possibilidades de trabalhos futuros.

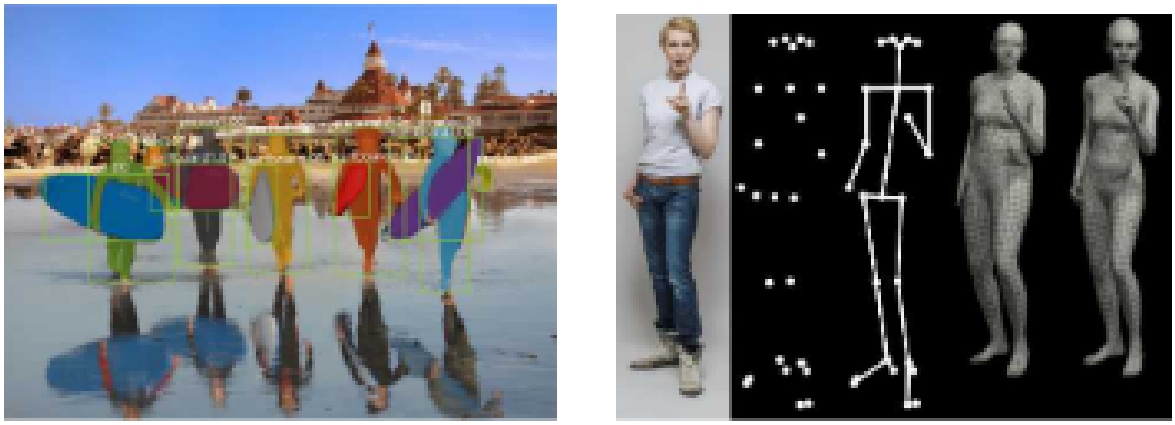
2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo faz uma breve apresentação sobre os conceitos abordados durante o desenvolvimento deste trabalho.

2.1 Visão Computacional

A Visão Computacional pretende interpretar e extrair informações das imagens em um nível próximo ou equivalente ao do olho humano. Ao longo dos anos, foram desenvolvidas uma variedade de técnicas e algoritmos utilizados para analisar e interpretar imagens. A Figura 1 mostra algumas destas técnicas, como segmentação de imagem, estimação de pose, detecção, reconhecimento e rastreamento de objetos.

Figura 1 – Segmentação de Imagem e Estimação de Pose



Fonte: (Szeliski, 2022)

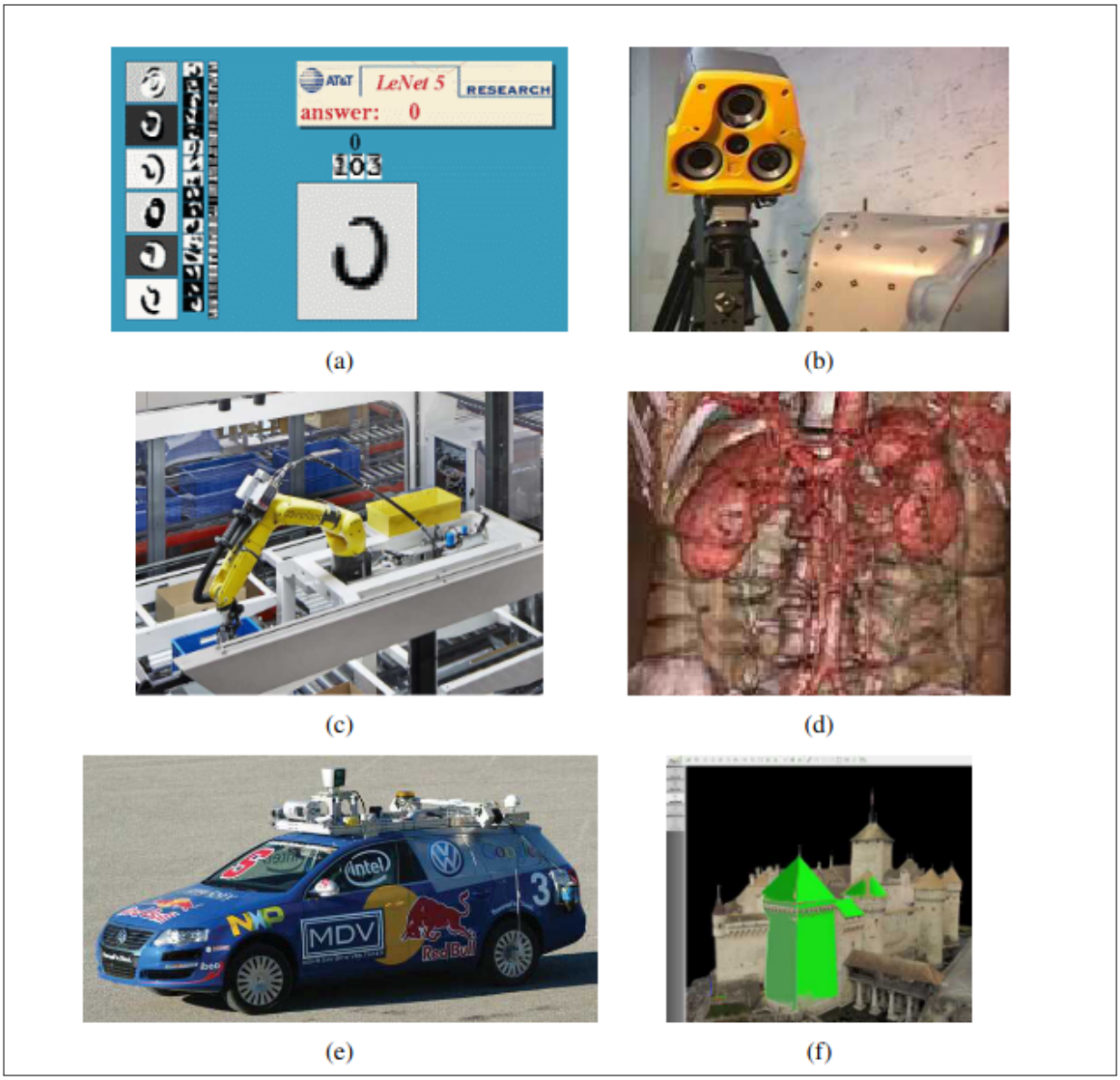
A visão computacional oferece uma ampla variedade de aplicações práticas em diversos setores. No campo de reconhecimento de padrões, destaca-se o *Optical Character Recognition* (OCR) (Reconhecimento Óptico de Caracteres), utilizado desde a leitura de códigos postais em cartas até o reconhecimento automático de placas veiculares (Figura 2a). Na indústria, a tecnologia é empregada para inspeção de máquinas e controle de qualidade, como na verificação de peças aeronáuticas usando visão estéreo com iluminação especializada ou na detecção de defeitos em fundições de aço através de imagens de raio-x (Figura 2b).

O setor logístico também se beneficia com soluções como entrega autônoma de pacotes, sistemas de transporte de paletes e coleta automatizada de peças por braços robóticos (Figura 2c). Na área médica, a visão computacional possibilita desde o registro de imagens

cirúrgicas até estudos longitudinais da morfologia cerebral durante o envelhecimento (Figura 2d).

Outras aplicações significativas incluem a navegação autônoma de veículos entre cidades (Figura 2e) e a construção automatizada de modelos 3D através de fotogrametria com imagens aéreas capturadas por drones (Figura 2f). Cada uma dessas aplicações demonstra a versatilidade e o potencial transformador da visão computacional em diferentes domínios tecnológicos.

Figura 2 – Aplicações no mundo real



Fonte: (Szeliski, 2022)

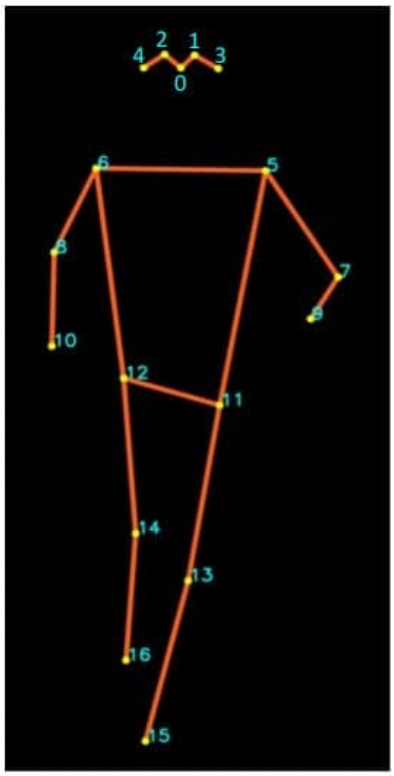
2.1.1 Estimação de Pose

A Estimação de Pose é uma tarefa de Visão Computacional cujo objetivo é localizar as partes do corpo humano e construir representações estruturadas, como esqueletos corporais, a partir de dados de entrada (Zheng *et al.*, 2023). Modelos baseados em Aprendizagem Profunda normalmente seguem uma abordagem em dois estágios: (i) detecção de objetos e (ii) localização dos pontos-chave. A Figura 3 ilustra a estrutura dos pontos-chave.

Apesar dos avanços alcançados com técnicas de aprendizagem profunda, a estimativa de pose humana ainda enfrenta desafios relevantes, como a escassez de dados anotados, ambiguidades na percepção de profundidade e problemas de oclusão (Zheng *et al.*, 2023).

Figura 3 – Mapa dos pontos-chave de retorno no YOLO11

Index	Key point
0	Nose
1	Left-eye
2	Right-eye
3	Left-ear
4	Right-ear
5	Left-shoulder
6	Right-shoulder
7	Left-elbow
8	Right-elbow
9	Left-wrist
10	Right-wrist
11	Left-hip
12	Right-hip
13	Left-knee
14	Right-knee
15	Left-ankle
16	Right-ankle



Fonte: (Gupta; Patil, 2021)

2.1.1.1 Modelos de Estimação de Pose

2.1.1.1.1 HRNet

O HRNet (*High-Resolution Network*) é um modelo de estimação de pose que mantém representações em alta resolução durante todo o processamento. Ao contrário de métodos

tradicionais que reduzem a resolução para depois tentar recuperá-la, o HRNet inicia com uma rede de alta resolução e vai adicionando, em paralelo, sub-redes com resoluções menores, realizando várias fusões entre essas escalas. Isso permite que informações de diferentes resoluções sejam constantemente combinadas, gerando representações mais ricas e detalhadas. Como resultado, o modelo produz mapas de calor para pontos-chave corporais com maior precisão espacial. A eficácia do HRNet foi comprovada com resultados superiores em conjuntos de dados benchmark como COCO e MPII (Sun *et al.*, 2019).

2.1.1.1.2 OpenPose

O OpenPose é o primeiro sistema em tempo real capaz de detectar simultaneamente os pontos-chave do corpo, mãos, rosto e pés (totalizando 135 pontos) de múltiplas pessoas em imagens. Desenvolvido por Ginés Hidalgo e colaboradores, ele utiliza o conjunto de dados CMU Panoptic Studio para treinar seus modelos. O sistema oferece detecção 2D para várias pessoas e também reconstrução 3D para indivíduos, suportando diversas fontes de entrada como vídeo, webcam e câmeras especializadas. Além disso, possui ferramentas para calibração de câmeras e rastreamento de pessoas, garantindo maior precisão e desempenho constante independentemente do número de pessoas na cena. O OpenPose funciona em múltiplos sistemas operacionais e suporta GPUs Nvidia e AMD, além de CPU-only. Pode ser usado via linha de comando ou integrado em projetos por meio de APIs em C++ e Python, facilitando customizações e extensões (Simon *et al.*, 2017).

2.1.1.1.3 DeepCut

O DeepCut é uma abordagem para estimação da pose humana de múltiplas pessoas em imagens reais que integra simultaneamente a detecção e a estimativa da pose. Diferente de métodos tradicionais que primeiro detectam as pessoas e depois estimam suas poses, o DeepCut infere o número de pessoas presentes, identifica partes do corpo ocultas e resolve ambiguidades entre pessoas próximas. Para isso, utiliza uma formulação baseada em programação linear inteira que agrupa hipóteses de partes do corpo, geradas por detectores CNN, aplicando supressão de não-máximos e respeitando restrições geométricas e visuais. Seus resultados alcançam desempenho de ponta em vários conjuntos de dados para poses de indivíduos únicos e múltiplos (Insafutdinov *et al.*, 2016).

2.1.1.1.4 AlphaPose

O AlphaPose é um framework de estimação de pose de múltiplas pessoas que combina alta precisão e velocidade. Baseado na técnica RMPE (*Regional Multi-Person Pose Estimation*), ele é o primeiro sistema open-source a superar 70 mAP no COCO e 80 mAP no MPII. Com suporte para diferentes conjuntos de pontos-chave (COCO, Halpe, SMPL), o AlphaPose alcança resultados superiores ao OpenPose e Mask R-CNN, sendo ideal para aplicações em tempo real em Linux e Windows (Fang *et al.*, 2022).

2.1.1.1.5 PoseNet

O PoseNet é um modelo leve desenvolvido pelo Google para detecção de pose 2D em tempo real, sendo especialmente indicado para aplicações em navegadores e dispositivos móveis. Ele detecta até 17 pontos-chave do corpo humano, seguindo o padrão COCO, com foco em eficiência e baixa demanda computacional. Diferente de outros modelos como MoveNet ou BlazePose, o PoseNet suporta a detecção de múltiplas pessoas em uma mesma imagem, tornando-se uma solução prática para ambientes web e aplicações em tempo real que exigem simplicidade e portabilidade (Kendall *et al.*, 2016).

2.1.1.1.6 GHUM

O GHUM (*Generalized Human Model*) é um pipeline completo para modelagem estatística de corpos humanos em 3D, treinado com mais de 60 mil digitalizações de pessoas em diferentes poses, expressões faciais e gestos das mãos. A arquitetura usa redes profundas com autoencoders variacionais para capturar variações de forma e pose, integrando articulação corporal completa, expressões faciais e mãos. O modelo possui espaços latentes de forma (16 dimensões) e expressão facial (20 dimensões), esqueleto articulado com 63 juntas e restrições anatômicas, além de modelos estatísticos prévios cinemáticos baseados em mais de 2 milhões de movimentos corporais e 4,8 mil gestos manuais. São disponibilizados dois modelos prontos: GHUM (alta resolução) e GHUMLite (resolução reduzida) (Xu *et al.*, 2020).

2.2 Processamento Digital de Sinais

O processamento digital de sinais é uma área da Engenharia e da Ciência da Computação dedicada ao estudo e à aplicação de métodos para analisar, modificar e interpretar sinais representados por sequências de números. Esses sinais são discretos no tempo e no valor, ou seja, são amostras digitais de fenômenos físicos que originalmente variam de forma contínua, como som, temperatura, luz ou pressão (Diniz *et al.*, 2014).

Os sinais podem ser processados de forma analógica ou digital. No modo analógico, o sinal contínuo é manipulado diretamente por circuitos eletrônicos, mas com limitações em precisão e resistência a ruídos. Já no processamento digital, o sinal é convertido em valores numéricos por amostragem e quantização, permitindo manipulações mais precisas, como filtragem, análise espectral e remoção de ruídos, por meio de sistemas computacionais (Diniz *et al.*, 2014). Exemplos práticos de processamento digital de sinais incluem:

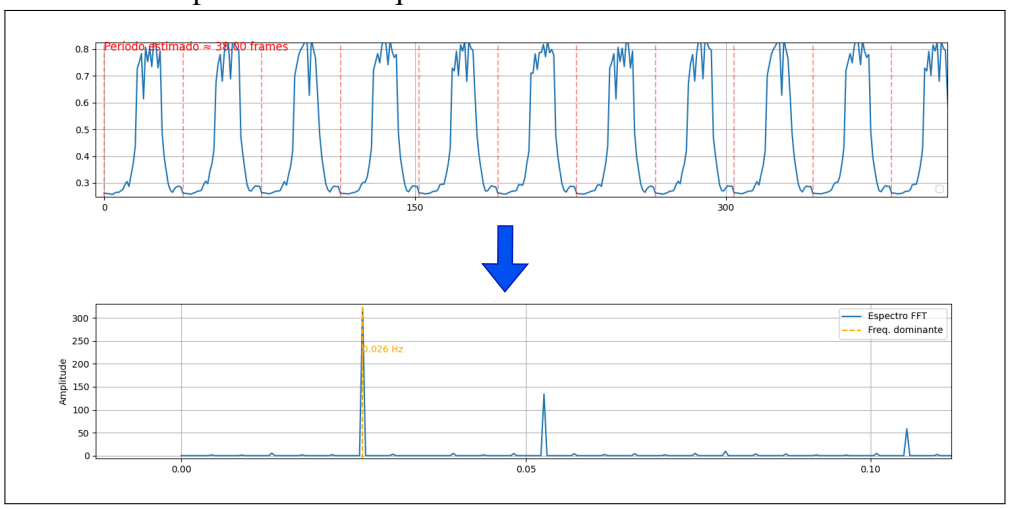
1. CD players: que processam sinais de áudio digital para reprodução de som.
2. Tomografia computadorizada: onde sinais são processados para formar imagens médicas detalhadas.
3. Sistemas de comunicação: como os telefones celulares, que convertem e processam voz e dados digitalmente.
4. Brinquedos eletrônicos e assistentes virtuais: que utilizam reconhecimento de voz.
5. Aplicações geológicas: para análise de dados sísmicos.

2.2.1 Transformadas de Fourier e o Algoritmo FFT

A Transformada Rápida de Fourier (FFT) é uma técnica fundamental do Processamento Digital de Sinais utilizada para converter sinais do domínio do tempo para o domínio da frequência (Heckbert, 1995), representado na Figura 4. Essa conversão possibilita a identificação das componentes frequenciais que compõem o sinal original, permitindo a análise de padrões periódicos presentes na sequência temporal.

A FFT pode ser aplicada para identificar as frequências dominantes que correspondem ao ritmo de execução dos movimentos. Dessa forma, pode-se estimar o período médio de um ciclo de um movimento, facilitando a identificação do padrão temporal do gesto.

Figura 4 – Sinal do tempo de um movimento de uma articulação decomposto em seus componentes de frequência usando FFT



Fonte: Elaborado pelo autor (2025).

Matematicamente, a transformada discreta de Fourier (DFT), que serve como base para a FFT, é dada por:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi kn/N} \tag{2.1}$$

onde:

- x_n representa o valor da sequência no domínio do tempo no instante n,
- X_k é o valor da sequência no domínio da frequência na componente k,
- N é o número total de amostras,
- k é o índice da frequência analisada.

Antes da aplicação da transformada, é comum realizar a centralização do sinal para remoção da componente referente ao deslocamento (DC [1]), garantindo que o valor médio não distorça o espectro de frequência:

$$x_n = y_n - \frac{1}{N}\sum_{n=0}^{N-1} y_n \tag{2.2}$$

onde y_n é o sinal original no domínio do tempo. A magnitude da componente espectral X_k, que indica a amplitude associada à frequência f_k, é calculada pelo módulo complexo:

$$|X_k| = \sqrt{\Re(X_k)^2 + \Im(X_k)^2} \tag{2.3}$$

A frequência dominante f_{dom}, que reflete o ritmo principal do movimento, é identificada como aquela que possui a magnitude espectral mais forte:

$$f_{\text{dom}} = f_k \quad \text{onde} \quad |X_k| = \max(|X|) \tag{2.4}$$

[1] Direct Current ou, no português, corrente contínua (DC): um termo herdado da eletrônica. Representa a média do sinal ao longo do tempo.

A partir dessa frequência, calcula-se o período do movimento T, ou seja, o intervalo médio entre repetições do ciclo:

$$T = \frac{1}{f_{\text{dom}}} \tag{2.5}$$

Essa análise permite a visualização e quantificação dos padrões cíclicos presentes nos movimentos capturados, facilitando a interpretação da regularidade e do ritmo do movimento.

2.3 Análise de Séries Temporais

Uma série temporal é um conjunto de observações ordenadas no tempo, não necessariamente igualmente espaçadas, que apresentam dependência serial, isto é, dependência entre os valores ao longo dos instantes de tempo. A notação usualmente utilizada para representar uma série temporal é $S_1, S_2, S_3, \ldots, S_T$, onde T indica o tamanho total da série. Esse tipo de dado é amplamente encontrado em fenômenos físicos, biológicos, econômicos e de engenharia, nos quais o comportamento observado ao longo do tempo revela padrões que podem ser estudados e modelados (Gutiérrez, 2003).

A análise de séries temporais refere-se ao estudo de dados coletados ao longo do tempo, onde as observações mantêm uma ordem cronológica, geralmente apresentando dependência entre os valores ao longo dos instantes.

As séries temporais podem ser decompostas em três componentes principais:

1. Tendência: comportamento de longo prazo da série, como crescimento ou declínio.
2. Ciclo: oscilações suaves e irregulares associadas a fatores como atividade econômica.
3. Sazonalidade: variações que ocorrem em intervalos regulares, como dias da semana ou estações do ano.

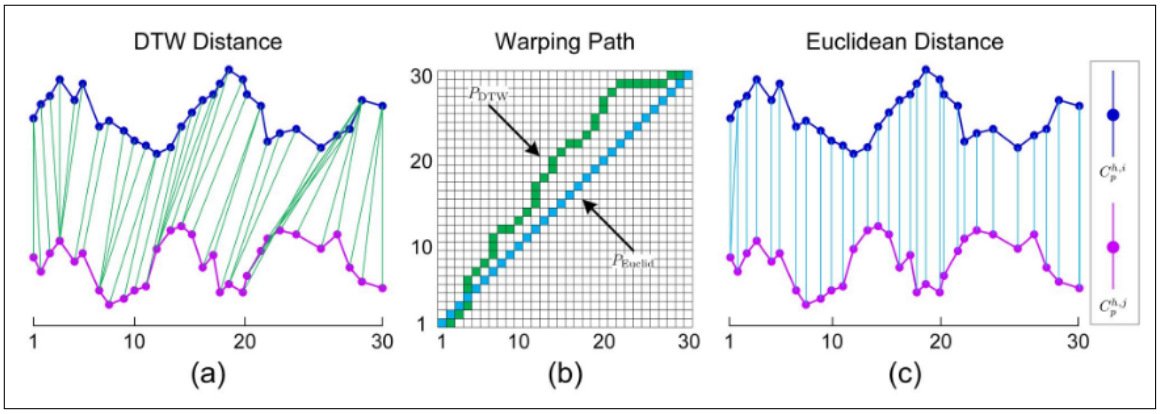
Um conceito fundamental é a estacionariedade, que ocorre quando a média e a variância da série permanecem constantes ao longo do tempo. Séries com tendência ou sazonalidade geralmente não são estacionárias, exigindo técnicas de transformação (como diferenciação ou remoção da média) para torná-las adequadas à modelagem estatística ou computacional (Gutiérrez, 2003).

As análises podem ter dois objetivos principais:

1. Modelagem: entender a estrutura da série e suas relações com outras variáveis.
2. Previsão: estimar valores futuros com base nos dados históricos.

2.3.1 *Dynamic Time Warping*

Figura 5 – (a)Alinhamentos DTW para medir distância/semelhança.(b)Caminhos PDTW (DTW) e PEuclid (Euclidiana).(c)Pareamento via distância Euclidiana.



Fonte: (Li *et al.*, 2019)

O *Dynamic Time Warping* (DTW) é um poderoso algoritmo capaz de medir similaridade entre sequências temporais que podem variar em velocidade ou duração, encontrando o alinhamento ótimo através de uma matriz de custos acumulados $D(i, j)$ (Sakoe; Chiba, 1978).

O algoritmo DTW calcula o alinhamento ótimo entre duas séries temporais $X = (x_1, \ldots, x_n)$ e $Y = (y_1, \ldots, y_m)$, construindo uma matriz de custo acumulado $D(i, j)$, definida recursivamente pela seguinte equação:

$$D(i,j) = d(a_i, b_j) + \min \begin{cases} D(i-1,j), \\ D(i,j-1), \\ D(i-1,j-1) \end{cases} \tag{2.6}$$

Onde:

- $d(a_i, b_j)$ representa a função de distância entre os elementos a_i e b_j, explicado na Seção 2.3.1.1;
- O valor de $D(i, j)$ é calculado como o custo atual somado ao menor custo acumulado entre as três posições vizinhas: *acima* $D(i-1, j)$, *à esquerda* $D(i, j-1)$ e *na diagonal* $D(i-1, j-1)$.

Esse processo garante a construção da trajetória ótima de alinhamento entre as duas sequências, minimizando o custo total.

O algoritmo é particularmente útil para comparar padrões temporais que ocorrem em diferentes momentos, como no reconhecimento de voz, onde frases idênticas pronunciadas

em velocidades distintas apresentam formas de onda similares, mas desalinhadas temporalmente (Warping, 2025; Yadav; Alam, 2018) superando métodos tradicionais baseados em comparação pontual (Sakoe; Chiba, 1978).

2.3.1.1 Funções de Distância

O DTW depende da definição de uma função de distância $d(a_i, b_j)$ entre os elementos a_i e b_j das duas sequências a serem comparadas. A escolha dessa função afeta diretamente o alinhamento e o custo acumulado da trajetória ótima. Abaixo, são apresentadas algumas das métricas mais comuns utilizadas na análise:

1. **Distância Euclidiana ao Quadrado** (*Squared Euclidean Distance*):

$$d(x,y) = \sum_{i=1}^{n} (x_i - y_i)^2 \tag{2.7}$$

   Esta métrica acentua diferenças maiores, pois eleva os desvios ao quadrado, sendo particularmente sensível a variações de escala e outliers [2]. Pequenas discrepâncias podem ser amplificadas, o que a torna adequada para contextos em que grandes desvios devem ser mais penalizados.

2. **Distância Euclidiana** (*Euclidean Distance*):

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.8}$$

   Representa a distância geométrica tradicional entre dois vetores. Assim como a versão ao quadrado, é sensível à escala dos dados, embora em menor grau. Pode ser afetada por valores extremos, mas oferece uma medição equilibrada da similaridade.

3. **Distância Manhattan** (*Manhattan Distance*):

$$d(x,y) = \sum_{i=1}^{n} |x_i - y_i| \tag{2.9}$$

   Calcula a soma das diferenças absolutas entre os elementos, sendo menos sensível a valores extremos e picos isolados. Embora ainda dependa da escala dos dados, apresenta maior robustez a ruídos locais e discrepâncias pontuais do que as métricas anteriores.

[2] Outlier: um dado que se encontra fora do padrão geral de uma distribuição.

2.3.2 Outros Modelos de Análise Temporal

2.3.2.1 Long Short-Term Memory (LSTM)

O modelo Long Short-Term Memory (LSTM) [3] é um algoritmo eficiente para aprendizado sequencial com redes neurais. Redes recorrentes convencionais enfrentam sérias dificuldades para capturar dependências de longo prazo. O LSTM foi projetado para contornar esse gargalo com uma arquitetura especializada que garante a estabilidade do fluxo de erro ao longo do tempo. Ao utilizar mecanismos de controle chamados portas (gates), o algoritmo decide dinamicamente quando armazenar, esquecer ou expor informações, tornando-se extremamente eficiente em tarefas que exigem memória de longo alcance, como reconhecimento de fala, modelagem de linguagem e análise de séries temporais. Sua complexidade computacional por passo de tempo é constante em relação ao número de pesos, o que o torna escalável e viável para aplicações reais com grandes volumes de dados temporais (Hochreiter; Schmidhuber, 1997).

2.3.2.2 Gated Recurrent Unit (GRU)

O modelo Gated Recurrent Unit (GRU) [4] surgiu como uma alternativa eficiente ao modelo LSTM, com o objetivo de oferecer desempenho comparável em tarefas sequenciais, porém com menor complexidade computacional. O algoritmo foi projetado para mitigar os problemas de gradientes que desaparecem ou explodem em redes recorrentes tradicionais, tornando o aprendizado de dependências de longo prazo mais estável. O GRU simplifica a estrutura ao utilizar apenas duas portas (atualização e reinicialização), em contraste com as três do LSTM, o que reduz o número de parâmetros e o tempo de treinamento. Essa estrutura mais enxuta permite ao GRU capturar de forma eficiente informações relevantes ao longo de sequências temporais, sendo amplamente adotado em tarefas como modelagem de linguagem, previsão de séries temporais e reconhecimento de voz. Seu desempenho competitivo aliado à simplicidade arquitetural faz do GRU uma escolha prática e poderosa em cenários onde recursos computacionais são limitados ou onde é necessário um treinamento mais rápido (Cho *et al.*, 2014).

[3] Long Short-Term Memory, no português Memória de Longo e Curto Prazo, uma rede recorrente especializada em aprender padrões de longo prazo em dados sequenciais.

[4] Gated Recurrent Unit, no português Unidade Recorrente com Portões, uma variação de redes recorrentes projetada para capturar dependências temporais com menos parâmetros.

2.3.2.3 Modelos AR, MR, ARMA E ARIMA

Os modelos Auto-Regressive (AR), Moving Average (MA) , Auto-Regressive Integrated Moving Average (ARMA e ARIMA) [5] formam uma classe fundamental de métodos estatísticos para modelagem e previsão de séries temporais. O modelo AR descreve uma série como uma regressão linear de seus próprios valores passados, enquanto o MA modela a série como uma combinação linear de erros passados. A combinação desses dois componentes dá origem ao modelo ARMA, adequado para séries estacionárias. Já o modelo ARIMA estende o ARMA ao incorporar uma etapa de diferenciação, tornando-o apto para lidar com séries não estacionárias — comuns em fenômenos do mundo real, como dados econômicos, ambientais e industriais. Esses modelos, descritos em profundidade no clássico "Time Series Analysis: Forecasting and Control", oferecem uma estrutura matemática robusta para capturar padrões temporais e realizar previsões com base em estruturas de dependência ao longo do tempo (Box *et al.*, 2015).

[5] Esses modelos são amplamente utilizados na modelagem de séries temporais, sendo o ARIMA uma extensão importante para lidar com séries não estacionárias.

3 TRABALHOS RELACIONADOS

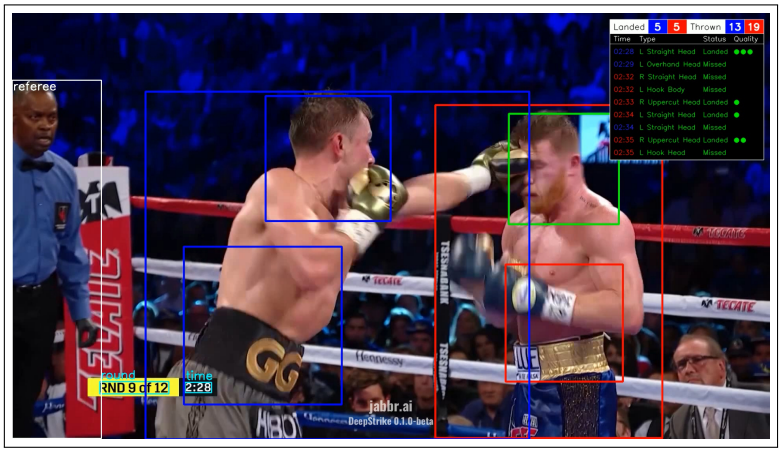
Foi feito um levantamento a fim de analisar as pesquisas mais relevantes que se assemelham à ideia deste projeto. Para identificar esses trabalhos, foram consultadas bases de dados acadêmicas, como Google Acadêmico, utilizando palavras-chave relacionadas ao tema da pesquisa, incluindo "artes marciais"ou "Visão Computacional"ou "estimação de pose"ou "Visão Computacional em artes marciais"ou "Visão Computacional em esportes"ou "estimação de pose e artes marciais"ou "estimação de pose aplicada em esportes"ou "aprendizagem de máquina"ou "algoritmos de classificação supervisionada"ou "classificação com knn"ou "dtw"ou "dtw keypoints"ou "keypoints"ou "yolo"ou "fft"ou "análise de periodicidade"ou "muay thai"ou "dtw sport"ou "keypoints sports"ou "tecnologia no esporte".

3.1 *Jabbr.ai*

Jabbr.ai é um *startup* de tecnologia com a promessa de construir o futuro dos esportes de combate. Após dois anos de pesquisa foi criado o *DeepStrike* (3.1.1), a primeira Inteligência Artificial (IA) de Visão Computacional projetada especificamente para esportes de combate. Em junho de 2023, o CEO Allan Svejstrup foi convidado para ministrar uma palestra sobre o *Jabbr* no *workshop* de esportes da *Computer Vision and Pattern Recognition Conference* (CVPR) (Jabbr.ai, 2024).

3.1.1 DeepStrike

Figura 6 – Detecção pelo *DeepStrike*.



Fonte: (Jabbr.ai, 2024)

O *DeepStrike* é uma tecnologia que rastreia os lutadores e o juiz, distinguindo-os com precisão. Conforme representado na Figura 6, o sistema detecta a cabeça e o torso dos atletas dentro das *bounding boxes*, permitindo visualizar o contato de um golpe aplicado pelo adversário. Além disso, utiliza OCR para reconhecer o *round* atual e o tempo de ação decorrido.

A ferramenta funciona com qualquer entrada de vídeo, seja profissional, amador ou de *sparring*, podendo ser utilizada com *smartphones* ou câmeras convencionais, tanto em dispositivos portáteis quanto em tripés. O *DeepStrike* realiza a geração automática de estatísticas, fornece *insights* personalizados e mede mais de 50 parâmetros para cada lutador.

Entre os principais parâmetros analisados estão: o número total de socos acertados por cada lutador (Acertos); os socos de alto impacto, que são aqueles que acertaram de forma limpa com força e efeito visível; e o total de socos lançados. O sistema também calcula a porcentagem de pressão sobre o oponente, considerando indicadores como levar o adversário às cordas ou ao *corner*, avançar fazendo o oponente recuar, e permanecer lutando em distâncias curtas ou médias.

3.2 *High Performance Moves Recognition and Sequence Segmentation Based on Key Poses Filtering*

Nessa pesquisa Vicente *et al.* (2016) aborda a análise de movimentos de atletas de *Taekwondo* de alto rendimento, tradicionalmente realizada através de gravações de vídeo com avaliação humana adicional. No entanto, esse processo manual é propenso a erros e consome uma quantidade significativa de tempo. Nesse contexto, o presente trabalho propõe uma metodologia automatizada para auxiliar atletas na melhoria de seu desempenho, através do reconhecimento e segmentação automática de movimentos em sequências de golpes durante o treinamento. Os resultados obtidos podem ser utilizados pelos treinadores para uma análise mais precisa e detalhada da performance do atleta.

A metodologia adotada nesse estudo baseia-se na filtragem de poses-chave, onde é selecionado um pequeno conjunto de quadros para o processo de aprendizagem. Para isso, Vicente *et al.* (2016) emprega o modelo discriminativo *Latent-Dynamic Conditional Random Field* (LDCRF), que é capaz de capturar dinamicamente as subestruturas intrínsecas e extrínsecas dos movimentos.

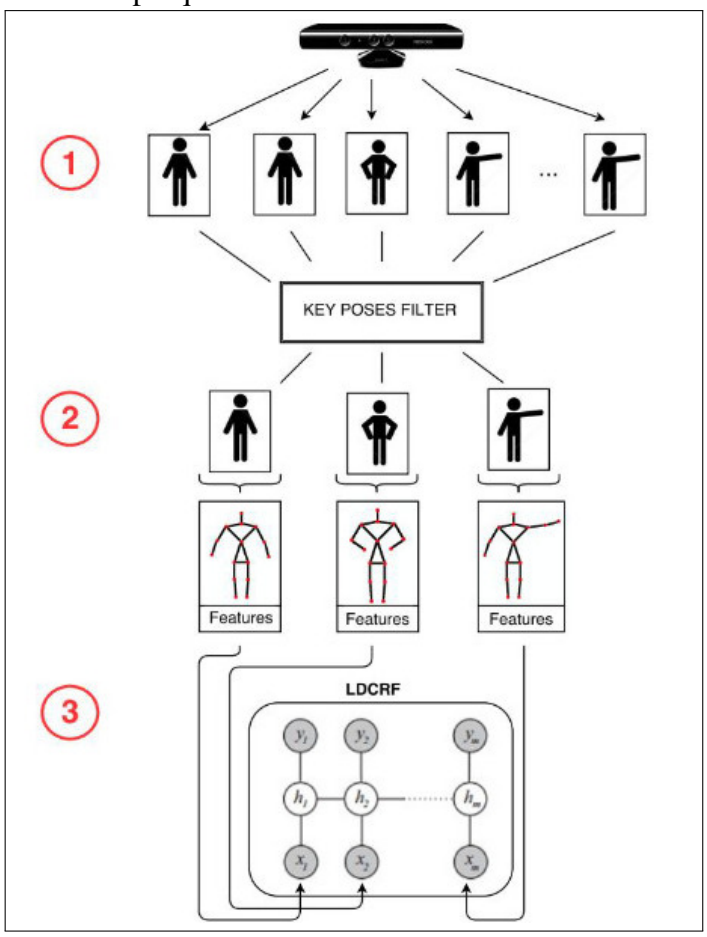
Um desafio enfrentado na representação de movimentos rápidos é a necessidade de uma grande quantidade de quadros. No entanto, esse processo pode gerar um grande número de

quadros semelhantes, o que pode retardar o reconhecimento e diminuir a taxa de classificação devido ao ruído nos dados. Para contornar esse problema, Vicente *et al.* (2016) propõe o método de filtragem com base em poses-chave, que gera um conjunto conciso de poses discriminativas capazes de representar um movimento. Esse processo de amostragem adaptativa permite uma representação mais simples dos movimentos, utilizando menos dados na etapa de aprendizagem.

O primeiro objetivo é realizar um aprendizado supervisionado utilizando um conjunto de treinamento composto por uma sequência de quadros $\{f_1, \ldots, f_n\}$ e seus respectivos rótulos de movimentos $\{y_1, \ldots, y_n\}$, para treinar o modelo LDCRF.

Esta abordagem de aprendizagem é representada na Figura 7, onde Vicente *et al.* (2016) busca reconhecer e segmentar sequências não rotuladas que contêm execuções de diferentes movimentos.

Figura 7 – Representação gráfica da metodologia aplicada na pesquisa



Fonte: (Vicente *et al.*, 2016)

Sua metodologia é descrita em três etapas principais:

1. Extração de Pontos-Chave (3.2.1): Cada sequência treinada passa pelo processo de extração

de pontos-chave, onde são identificados os pontos-chave discriminativos que representam os movimentos.

2. Rotulação e Filtragem de Pontos-Chave (3.2.2): Os pontos-chave de cada sequência treinada são rotulados e filtrados, visando selecionar apenas os mais relevantes para o processo de aprendizagem.
3. Treinamento LDCRF (3.2.3): O modelo LDCRF é treinado e testado utilizando os quadros filtrados, representados por seus vetores de características calculadas de Φ. Este modelo é fundamental para reconhecer e segmentar os movimentos nas sequências não rotuladas.

3.2.1 Extração de Poses-Chaves

Para extrair as poses-chave, o método inicialmente representa um registro de gestos (vários gestos realizados com uma pequena pausa entre cada um) como uma curva em um espaço de alta dimensão, assumindo que entre cada execução de gesto, o usuário realiza pequenas pausas. Esses intervalos de pausa podem ser identificados por intervalos de alta curvatura, enquanto os gestos são identificados por intervalos de baixa curvatura, o que permite uma segmentação robusta dos gestos. Este processo é realizado através de um esqueleto estimado em profundidade.

Após a segmentação dos movimentos, as poses-chave podem ser extraídas usando uma estratégia que segue:

1. A primeira e a última pose de cada gesto são marcadas como poses principais. Se um movimento começar e terminar na mesma pose, a pose que mais difere das poses inicial e final também será adicionada.
2. Se forem encontrados movimentos com a mesma representação, então poses discriminantes serão adicionadas a esses conjuntos de movimentos até que todas as representações se tornem únicas.

As poses devem ser treinadas previamente: primeiro, é definido o grupo de movimentos que é desejada analisar e, em seguida, é submetido ao processo de extração das poses. A informação do esqueleto é composta por:

- Articulações: Coordenadas 3D de cada uma das 20 articulações do corpo.
- Ângulos de articulações: 9 ângulos zenitais θ, 8 ângulos azimutais ϕ das articulações superiores e inferiores do corpo.
- O vetor com o valor das poses-chave é dado por: $\Phi(x_j) = (J_1, \ldots, J_{20}, \theta_1, \ldots, \theta_9, \phi_1, \ldots, \phi_8)^T$

3.2.2 Processo de Filtragem

Após a conclusão da etapa 3.2.1, foi obtido um conjunto de poses $K = \{k_1, k_2, \ldots, k_m\}$, agrupando os pontos-chave extraídos dos movimentos. O autor pegou a sequência de quadros $\{f_1, \ldots, f_n\}$ e mapeou cada um dos pontos-chaves do conjunto de poses K, usando o classificador *K-Nearest Neighbors* (KNN) combinado com o limite de similaridade de pose $\varepsilon = 2\pi$ para classificar o quadro f_i. Caso não haja uma pose atribuída a um quadro, será retornada uma pose inválida.

No caso de uma execução de movimento a 30 *Frames Per Second* (FPS) haverá pouca variação de pose do corpo entre uma sequência de quadros contínuos. Assim, este processo irá atribuir a mesma pose-chave a uma sequência de quadros contínuos até que a pose de um determinado quadro se torne o mais semelhante a outra pose-chave. No entanto, é possível escolher o quadro mais adequado desta sequência contínua que possui o mesmo rótulo, para ser o quadro representativo desse intervalo.

No final do processo é gerado um conjunto de quadros $X = \{x_1, x_2, \ldots, x_m\}$ que pode ser associado a um rótulo de movimentos, gerando o conjunto de rótulos de movimentos $Y = \{y_1, y_2, \ldots, y_m\}$. A estratégia de seleção de quadros é simples, mas eficaz. Dos quadros com o mesmo rótulo de poses-chave, foi escolhido o primeiro da sequência. Isso reduz significativamente a quantidade de informação armazenada em cada gravação, em vez de salvar a informação do esqueleto em cada quadro.

3.2.3 Treinamento e Teste de Modelo

O processo de filtragem reduz a quantidade de informações armazenadas e o tempo de processamento, contribuindo para a eficiência do reconhecimento. Para cada movimento realizado pelos atletas, foram extraídos 17 ângulos (θ, ϕ) das articulações superiores e inferiores por quadro a uma taxa de 30 FPS. Utilizando o modelo LDCRF, é possível mapear os quadros filtrados para seus respectivos movimentos de *Taekwondo*. O treinamento do LDCRF gera um modelo que estima a subestrutura da sequência de movimentos, representado por um vetor de variáveis ocultas. Este modelo estima o rótulo mais provável para uma nova sequência de poses-chave filtradas, maximizando o modelo condicional.

3.3 *The application of improved DTW algorithm in sports posture recognition*

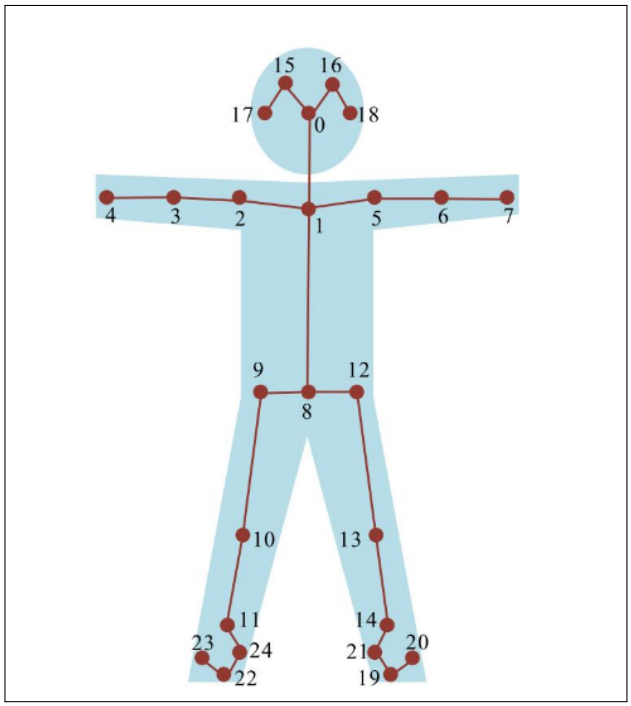
Neste artigo, Niu (2024) propõe um modelo inovador para reconhecimento de posturas em esportes, com validação experimental em movimentos de Tai Chi. A abordagem combina três técnicas fundamentais:

- *DTW (Dynamic Time Warping)* para comparação de séries temporais de pontos articulares
- *KNN (K-Nearest Neighbors)* para classificação das posturas
- *PCA (Principal Component Analysis)* para redução de dimensionalidade

3.3.1 Metodologia

O processo inicia com a detecção de 24 pontos-chave do esqueleto humano utilizando o modelo OpenPose, conforme a Figura 8.

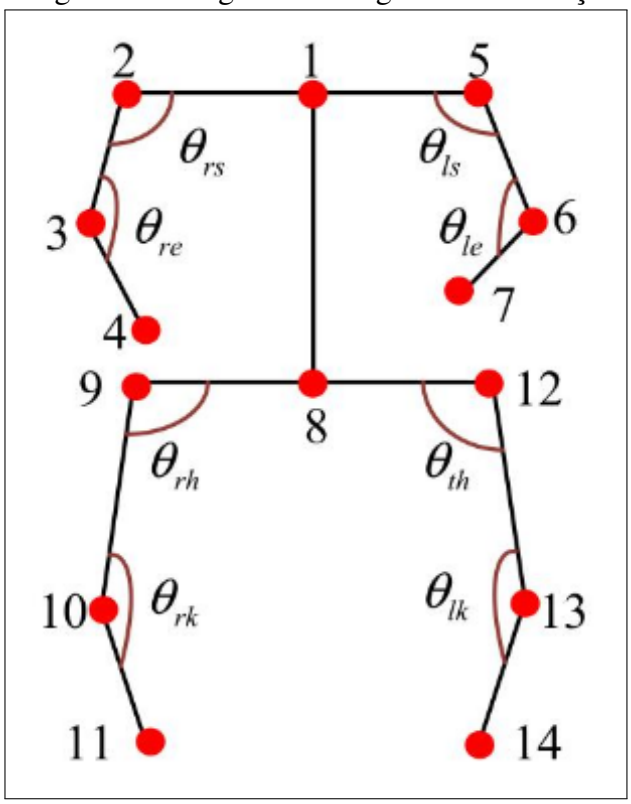
Figura 8 – Modelo de esqueleto OpenPose.



Fonte: (Niu, 2024)

As coordenadas (x, y) das articulações são extraídas e normalizadas. Então são selecionadas as juntas mais importantes para representar poses de Tai-Chi. A Figura 9 representa ângulos articulares, considerando 8 juntas principais.

Figura 9 – Diagrama do ângulo da articulação.



Fonte: (Niu, 2024)

O DTW tradicional é aprimorado pela introdução do KNN, fazendo uso da métrica de distância Euclidiana $d(A_i, B_i) = \sqrt{\sum_{\omega=1}^{N}(A_{i\omega} - B_{i\omega})^2}$, na qual é obtido o melhor caminho de menor distância.

3.3.2 Resultados Experimentais

Os testes utilizaram dois conjuntos de dados: o *Weizmann Dataset* com 90 vídeos de 10 posturas e um dataset próprio com 1.400 vídeos de 7 movimentos de Tai Chi. Cada sequência de vídeo tem uma taxa de quadros de 25 fps, uma resolução de 160 × 120 e uma duração média de 30 segundos. Os resultados mostraram redução de 74% na dimensionalidade com PCA e acurácia máxima de 89% com k=5 no KNN. Posturas como *"White Crane Spreads its Wings"* alcançaram precisão superior a 90%, comprovando a eficácia do método.

3.4 Comparativo entre trabalhos

O Quadro 1 apresenta uma análise detalhada das funcionalidades dos trabalhos relacionados (Jabbr.ai, 2024; Vicente *et al.*, 2016; Niu, 2024) em comparação com o trabalho

proposto, com foco nas aplicações de Visão Computacional.

Quadro 1 – Comparativo entre os trabalhos relacionados e o trabalho proposto.

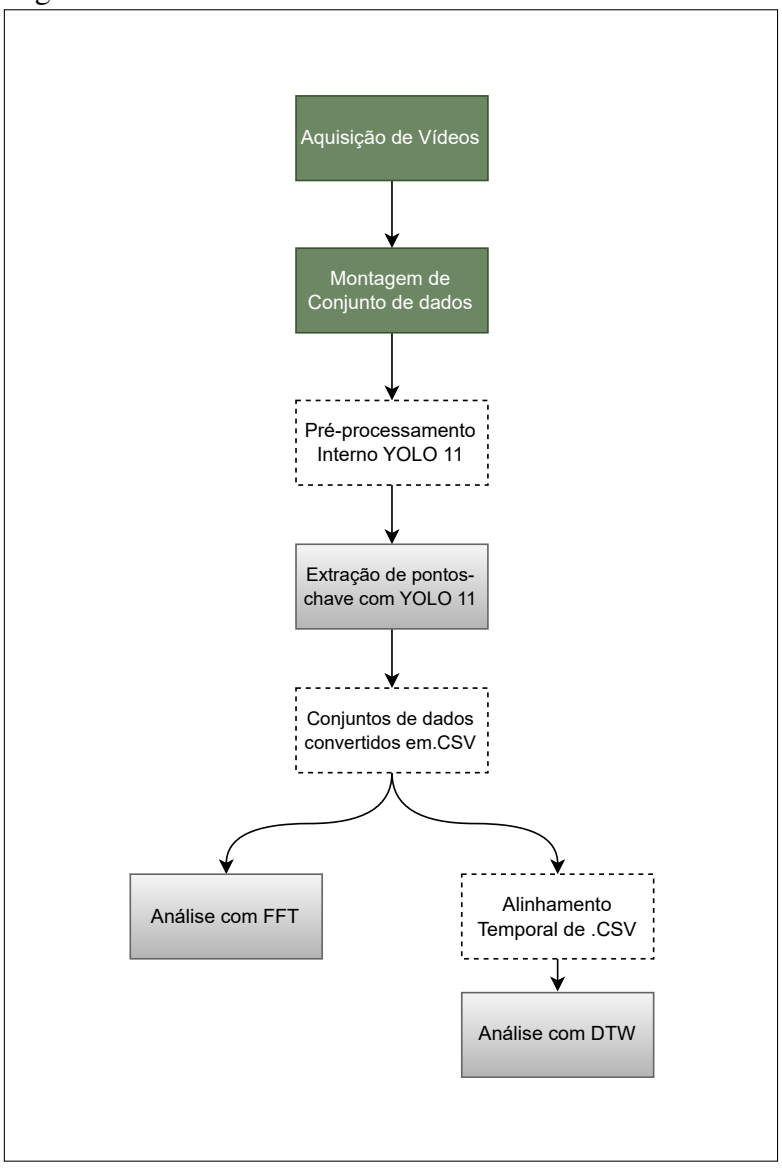
Comparativo entre trabalhos	**(Jabbr.ai, 2024)**	**(Vicente *et al.*, 2016)**	**(Niu, 2024)**	**Trabalho proposto**
Open Source			X	X
Conjunto de dados aberto			X	X
Conjunto de dados próprio		X		X
Aplicável em tempo real				
Receber vídeos gravados de entrada	X	X	X	
Geração de estatísticas	X			
Análise interpretativa dos dados	X			
Contagem automática de pontos	X			
Identificar padrões de movimento	X	X	X	X
Identificar sequências de golpes	X	X		
Extração de pontos-chave			X	X
Utilizar pontos-chave para treinamento de modelo		X	X	X
Utilizar modelo de aprendizagem supervisionada		X	X	
Prever resultados	X			X
Utilizar DTW para análise			X	X
Auxiliar no treinamento de atletas	X	X	X	X

Fonte: Elaborado pelo autor (2025).

4 METODOLOGIA

Para extrair as informações desejadas a partir da análise de vídeos de treinamento, foi realizada uma pesquisa aprofundada em trabalhos científicos sobre análise determinística e estimação de pose humana aplicadas ao esporte, com foco em artes marciais. A Figura 10 ilustra a linha de execução desta análise. O código usado no projeto de pesquisa foi escrito na linguagem Python, com auxílio de algumas bibliotecas [1] e produzido em dois ambientes de execução, o Visual Studio Code e o Google Colab. Nesse último, foi feito uso do acelerador de *hardware*, *GPU T4*, fornecido gratuitamente pela empresa para etapas de extração e análise.

Figura 10 – Fluxo Geral



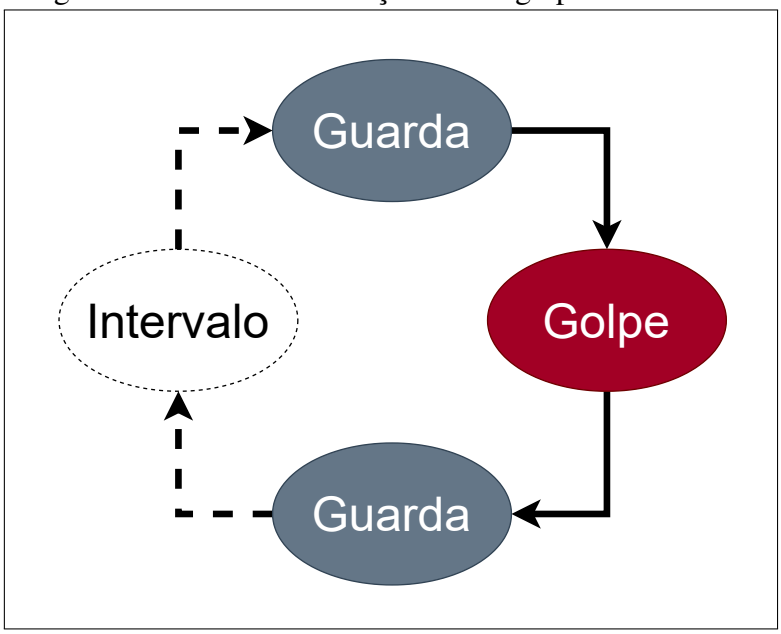
Fonte: Elaborado pelo autor (2025).

[1] OpenCV-Python: processamento de imagens; DTW-Python: cálculo de similaridade temporal; Ultralytics: modelos YOLO para visão computacional; NumPy e Pandas: manipulação de dados e arrays.

4.1 Aquisição de Vídeos de Golpes

Para esta análise, foram coletados vídeos contendo a execução das oito armas do Muay Thai, nas posturas de luta ortodoxa e canhota, para serem usados como dados de entrada em um algoritmo de análise determinística. Todo o conteúdo é de autoria própria e de outros atletas. Os vídeos foram gravados utilizando a câmera de um iPhone 11 [2]. Um golpe completo em qualquer um dos vídeos do conjunto de dados é representado por um ciclo de execução, conforme a figura 11.

Figura 11 – Ciclo de execução de um golpe



Fonte: Elaborado pelo autor (2025).

4.1.1 Montagem do Conjunto de Dados

Dois *datasets* foram montados para realizar as análises:

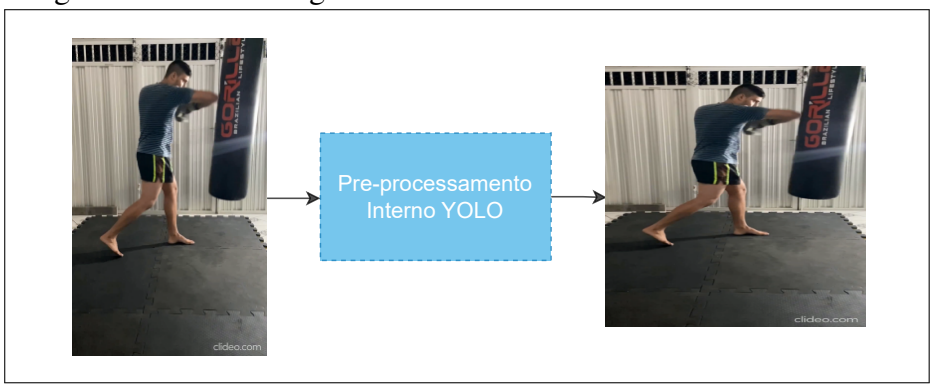
1) O primeiro é o conjunto de testes, que possui 69 vídeos, cada vídeo com duração entre 00:50, 01:20 e 04:00. Os vídeos de menor duração contêm entre 30 e 50 execuções contínuas de golpes, gerando entre 1000 e 4000 quadros, os maiores até 7000 quadros. O conjunto de dados foi dividido por diretórios contendo o nome dos golpes para o lado direito e esquerdo, sendo executados na postura destra. As classes representadas são jab, direto, cruzado, gancho, cotovelada circular, cotovelada reta, chute alto, chute frontal, chute baixo e joelhada.

[2] Gravados de frente e de lado, com enquadramento do corpo inteiro no modo retrato, em resolução 1080p a 30 FPS.

2) O segundo é o conjunto de referência, onde foi usada a ferramenta clideo (2025) para extrair amostras contendo uma única execução bem definida representando as classes. Após salvar cada golpe em seu diretório, foi decidido gerar golpes no lado canhoto, aplicando a técnica de *flip* horizontal [3] da biblioteca OpenCV para gerar a execução do golpe na postura de luta canhota (cruzado-esquerdo-destro e cruzado-equerdo-canhoto, por exemplo). Assim, finalizando a organização dos diretórios dos golpes, separados por lado (direito e esquerdo) e por postura (destra e canhota). Para uniformizar a duração dos vídeos e garantir consistência temporal entre as amostras, os vídeos originais que possuem aproximadamente foram replicados em loop para atingir aproximadamente 1 minuto de duração, gerando, então, a postura de luta canhota. O processo foi realizado em lote, utilizando a biblioteca OpenCV, que permitiu capturar os quadros do vídeo original e regravá-los, gerando o conjunto de dados de referência (executado em *loop* [4]) finalizado com 108 vídeos.

4.2 Extração de Pontos-Chave

Figura 12 – Vídeo original X Vídeo redimensionado

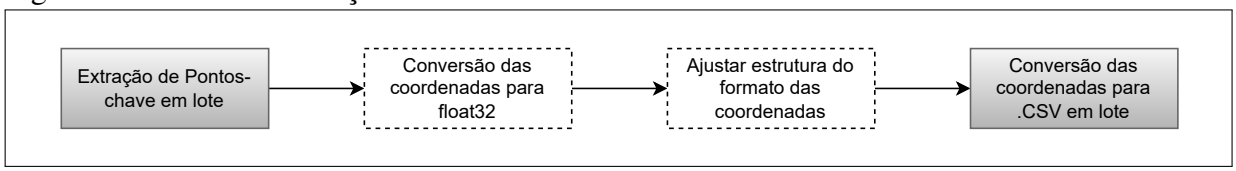


Fonte: Elaborado pelo autor (2025).

A Figura 12 ilustra o uso da técnica de redimensionamento interno [5] no YOLO. Este pré-processamento de dados visa padronizar o tamanho para manter a consistência dos vídeos obtidos, permitindo que a análise seja feita de forma eficaz. Caso o valor do tamanho não seja passado como argumento, os dados serão pré-processados com tamanho padrão de 640x640.

[3] Flip horizontal: Inverter a imagem no sentido horizontal.
[4] Loop: estrutura de repetição.
[5] Vídeos na resolução 1080p (1920 x 1080 pixels) para 640 x 640 pixels.

Figura 13 – Fluxo de Extração



Fonte: Elaborado pelo autor (2025).

Com os conjuntos de dados em formato de vídeo organizados, foi realizada a etapa de inferência para estimar a pose humana nos vídeos. Essa etapa foi realizada em lote, usando o YOLO 11 [6]. Este modelo retorna, para cada quadro do vídeo, um conjunto de pontos-chave representando articulações [7] do corpo humano em coordenadas 2D [x, y], possuindo tipo de dados torch.tensor [8] com estrutura **video-keypoints[pessoa][frame][articulação][coord]**.

Os dados foram convertidos para o tipo float32, com o objetivo de reduzir o uso de memória e garantir compatibilidade com métodos de análise temporal e modelos de aprendizado de máquina. E também, como cada vídeo possui apenas uma pessoa, a estrutura original foi simplificada para **video-keypoints[frame][articulação][coord]**, removendo a camada intermediária redundante.

Após o pré-processamento, as coordenadas foram armazenadas em arquivos .csv, organizados em pastas correspondentes aos tipos de golpes. Cada linha do arquivo representa um frame, e cada coluna corresponde a uma articulação, com as coordenadas formatadas como vetor de caracteres contendo pontos (x, y).

4.2.1 Ferramentas da Etapa

4.2.1.1 Pré-processamento dos pontos-chave com Numpy

O NumPy é uma biblioteca open source poderosa para computação numérica em Python, oferecendo arrays [9] N-dimensionais rápidos e eficientes, além de funções matemáticas avançadas, geração de números aleatórios, álgebra linear e transformadas de Fourier. Sua sintaxe de alto nível é acessível a usuários de todos os níveis, e seu núcleo, escrito em C, garante alto desempenho. Compatível com diversas plataformas, incluindo GPUs e bibliotecas de arrays esparsos, o NumPy é amplamente utilizado e mantido por uma comunidade ativa e colaborativa

[6] YOLO (You Only Look Once) versão 11: Modelo de Visão Computacional que oferece diversas tarefas.
[7] Mapeadas na Figura 3.
[8] Tensor: Estrutura multidimensional de dados (matriz, vetor, escalar).
[9] Array: Estrutura de dados que armazena vários elementos do mesmo tipo

(Harris *et al.*, 2020).

O pré-processamento dos pontos-chave extraídos, realizado entre as etapas de extrair e salvar (Figura 13), busca padronizar o tipo de dados e otimizar a estrutura. 1) As coordenadas foram convertidas para o tipo float32 para reduzir o consumo de memória e otimizar leituras posteriores desses dados. 2) A estrutura da saída do modelo foi espremida (*np.squeeze*), removendo a camada redundante que contém a quantidade de pessoas detectadas.

4.2.1.2 YOLO11 Pose

A *Ultralytics* é a empresa responsável pelo desenvolvimento, melhoria e distribuição do código aberto do YOLOv3 ao YOLOv12. Além disso, é uma biblioteca de código aberto para Python, oferecendo uma ampla gama de algoritmos complexos de visão computacional e aprendizagem profunda. A técnica de Visão Computacional foi a estimação de pose humana, uma das tarefas realizadas pelo YOLO11 [10], capaz de identificar pontos específicos em uma imagem. Estes pontos são normalmente nomeados de pontos-chave e podem significar várias partes de um objeto, como articulações, pontos de referência ou características específicas. Normalmente, estes pontos-chave são representados por uma matriz de coordenadas representadas em 2D ou 3D [11]. (Jocher *et al.*, 2024). O Quadro 2 mostra as principais características desta versão utilizada.

Os modelos de pose são pré-treinados pelo *dataset Microsoft Common Objects in Context* (MS-COCO) por padrão, ou por um conjunto de dados personalizado, e os resultados gerados por esta tarefa são um conjunto de pontos-chave de um objeto na imagem junto de seus respectivos valores de confiança (Jocher *et al.*, 2024).

Quadro 2 – Comparativo entre YOLOv10, YOLOv11 e YOLOv12

Característica	**YOLOv10 (yolo10, 2024)**	**YOLO11 (yolo11, 2024)**	**YOLO12 (yolo12, 2025)**
Lançamento	2024	**2024 (atual)**	2025
Destaque	Sem NMS	**+22% eficiência**	Atenção por regiões
Precisão (mAP)	38.5-54.4	**39.5-54.7**	40.6-55.2
Velocidade (ms)	1.8-10.7	**1.5-11.3**	1.6-11.8
Tamanho do Modelo	2.3-29.5M	**-22% params vs v8**	2.6-59.1M
Arquitetura	CSPNet	**Backbone refinado**	R-ELAN + FlashAttention
Melhor para	Edge devices	**Aplicações gerais**	Visão computacional avançada
Exportação para Edge	TensorRT, OpenVINO, CoreML	**TensorRT, ONNX**	TensorRT, ONNX com FlashAttention
GPU Mínima	T4	**T4 (FP16)**	Turing+ (FlashAttention)
Licença	AGPL-3.0	**AGPL/Enterprise**	AGPL-3.0
Destaque Técnico	Head 1-para-1	**Otimização holística**	Atenção espacial

Fonte: Elaborado pelo autor (2025).

[10] YOLO (You Only Look Once) versão 11: Modelo de Visão Computacional que oferece diversas tarefas.
[11] Coordenadas 2D são no formato [x, y], e as 3D incluem visibilidade: [x, y, visible].

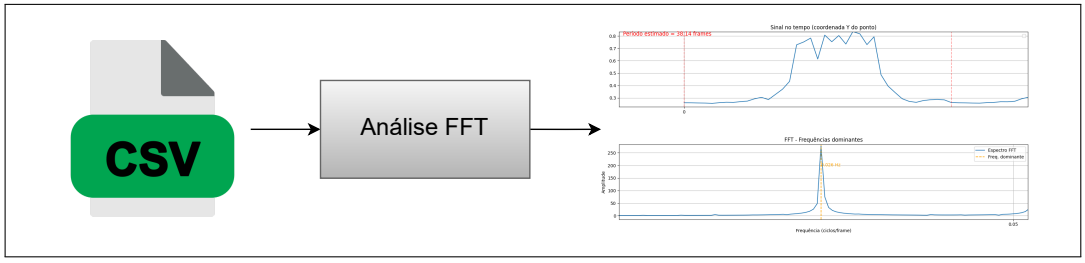
4.2.1.3 Salvar pontos-chave com Pandas

O Pandas é uma biblioteca open source para Python, licenciada sob BSD, que fornece estruturas de dados eficientes e ferramentas fáceis de usar para análise e manipulação de dados. Amplamente utilizada em ciência de dados, ela lida especialmente bem com dados tabulares e séries temporais, oferecendo funcionalidades como leitura e escrita de arquivos (como CSV e Excel), seleção e filtragem de dados, criação de colunas derivadas, geração de estatísticas resumidas, manipulação de texto, junção de tabelas, reorganização de estruturas e tratamento de dados temporais (pandas, 2025).

Para facilitar o armazenamento e a análise dos dados extraídos, foi implementado um método de conversão dos pontos-chave para estruturar os dados em um *DataFrame* [12]. Durante a conversão, os diretórios de saída são criados automaticamente, e cada CSV [13] é salvo com o índice representando a sequência temporal quadro a quadro. Esse processo foi realizado em lote para todos os vídeos, possibilitando uma análise posterior mais estruturada e automatizada dos movimentos.

4.3 Análise de períodos com Transformada Rápida de Fourier (FFT)

Figura 14 – Análise com FFT



Fonte: Elaborado pelo autor (2025).

Para obter o ritmo de execução dos movimentos, foi aplicada a Transformada Rápida de Fourier sobre as amostras de pontos-chave extraídos. Com o objetivo de identificar a frequência dominante associada à repetição do golpe, a partir dos deslocamentos registrados ao longo do tempo.

A função implementada permite calcular a frequência dominante para uma articulação específica — sendo esta o ponto de contato referente ao golpe executado — considerando

[12] Dataframe: Estrutura tabular
[13] CSV: Comma Separated Value, no português, Valores Separados por Vírgulas.

separadamente os eixos X, Y e também a *magnitude* [14] — o período obtido na magnitude é o indicador do ciclo completo do golpe —. Para cada um desses sinais:

1. O sinal é centralizado pela subtração da sua média, eliminando o descolamento (Equação 2.2).
2. A FFT é aplicada ao sinal, obtendo-se o espectro de frequências (Equação 2.1).
3. É calculada a magnitude espectral, onde considera-se apenas a metade positiva do espectro, excluindo o componente de frequência zero (DC) (Equação 2.3).
4. A frequência com maior amplitude (pico espectral)[15] é identificada como frequência dominante (f_{dom}) (Equação 2.4), e o período T correspondente é calculado pela inversa de f_{dom} (Equação 2.5).
5. Os resultados são visualizados em dois gráficos:
   - Sinal temporal com marcações do período de execução dos ciclos estimados.
   - Espectro de frequência, com destaque para a frequência dominante.

Desse modo é possível calcular e comparar as estimativas de período obtidas para cada eixo separadamente, bem como para a magnitude combinada dos deslocamentos. Tal análise é importante para verificar a consistência dos padrões cíclicos entre os movimentos no sentido horizontal (eixo x) e no sentido vertical (eixo y).

4.3.1 Ferramentas da Etapa

4.3.1.1 NumPy

Conforme citado na Seção 4.2.1.1, esta etapa também faz uso da biblioteca *NumPy* (Harris *et al.*, 2020).

O sinal é convertido para o domínio da frequência, por meio da função *np.fft.fft()* que implementa a Transformada de Fourier de maneira eficiente. Essa abordagem também permite identificar a frequência dominante correspondente ao pico mais alto, obtido com *np.abs()*, e extraído usando *np.argmax()*.

[14] Neste contexto, magnitude refere-se ao deslocamento total da articulação no plano, calculado a partir da combinação dos eixos X e Y, oferecendo uma medida integrada da movimentação.

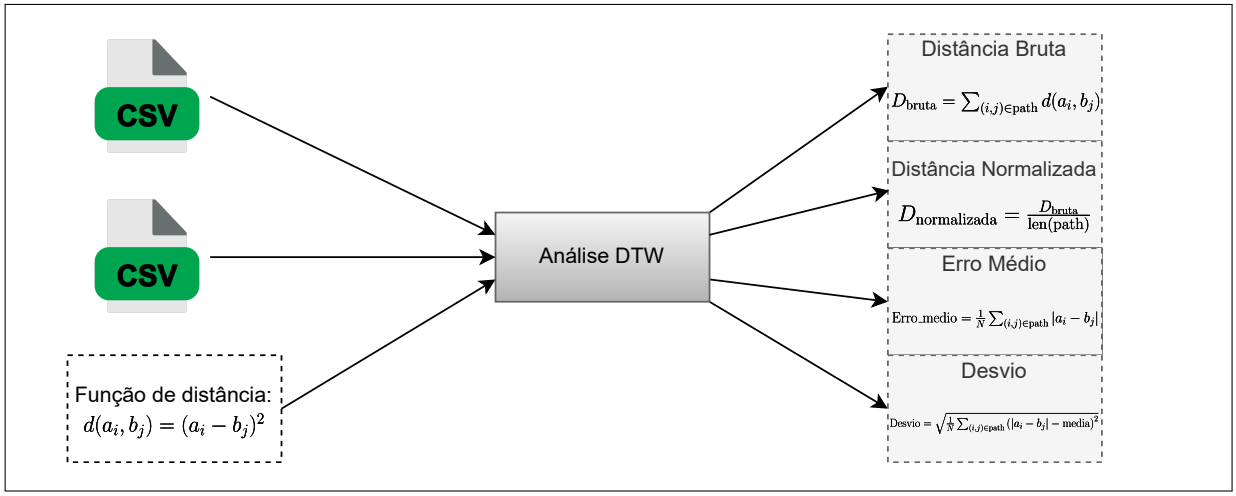
[15] Frequência com magnitude espectral mais forte.

4.4 Alinhamento Temporal manual nos arquivos. CSV

Esta etapa é simples e direta. Os conjuntos de dados criados passaram por um ajuste manual nos índices que representam o tempo — isto é, na quantidade de linhas dos arquivos CSV[16]. Para cada golpe, os arquivos com maior duração (mais linhas) foram truncados para igualar a quantidade de amostras dos arquivos menores correspondentes. Esse alinhamento visa garantir que os conjuntos de referência e teste apresentem a mesma quantidade de amostras para cada golpe, com o objetivo de reduzir os valores retornados no cálculo da distância entre as duas sequências temporais, conforme será discutido na Seção 4.5.

4.5 Análise de distâncias com DTW

Figura 15 – Fluxo de uso do DTW



Fonte: Elaborado pelo autor (2025).

Essa etapa consiste no uso da técnica Dynamic Time Warping (DTW) [17]. O algoritmo foi executado em lote, com o objetivo de analisar e comparar duas sequências temporais de pontos-chave extraídos dos vídeos com o YOLO 11 (4.2). Os dados foram pré-processados conforme detalhado na Seção 4.4, preservando variações angulares, velocidade e execução dos movimentos.

Além das amostras, uma métrica de distância (2.3.1.1) também foi usada como argumento no algoritmo DTW. Nesta pesquisa, foi usada a métrica de distância Euclidiana ao Quadrado, que apresentou melhores resultados entre as comparações realizadas entre amostras.

[16] CSV: Comma Separated Values, ou Valores Separados por Vírgulas.
[17] No português, Distorção Temporal Dinâmica.

Esse processo é aplicado em todas as articulações (Figura 3), convertidas de vetores bidimensionais para vetores unidimensionais [18]. As comparações foram realizadas percorrendo todas as combinações entre modelos de referência e modelos de teste para calcular a menor distância entre as séries temporais.

O modelo retorna automaticamente o caminho de alinhamento ótimo, que consiste em pares de coordenadas i, j, representando a correspondência entre os pontos da série 1 e da série 2. Esse caminho atua como um "mapa"que indica como as sequências foram deformadas temporalmente para alcançar o melhor alinhamento. A partir desse caminho, são computadas duas métricas principais pelo próprio algoritmo: a distância bruta DTW e a distância normalizada, conforme as Equações 4.2 e 4.3, respectivamente.

Para refinar a análise da semelhança entre os movimentos, foi calculado manualmente duas métricas adicionais: o erro médio ponto a ponto (Equação 4.4), que indica o erro médio de correspondência entre as duas sequências; e o desvio padrão dos erros (Equação 4.5), em que um desvio alto indica que alguns pontos estão muito mal alinhados, mesmo que o erro médio seja baixo. Essas métricas fornecem outra perspectiva sobre a qualidade do alinhamento.

4.5.1 Métricas de Distância

A biblioteca utilizada permite aplicar DTW com métricas personalizadas (Seção 2.3.1.1), preservando o caminho de alinhamento e calculando distâncias acumuladas e normalizadas. A distância euclidiana ao quadrado mostrou-se mais eficaz na distinção entre os golpes, sendo escolhida como métrica principal. Abaixo estão as métricas testadas:

Tabela 1 – Comparação das Métricas de Distância com diferentes limiares

Métrica	**Expressão Matemática**	**Limiar**	**Porcentagem**
Euclidiana ao Quadrado	$\sum(x_i - y_i)^2$	0.01	57.58%
		0.03	86.36%
		0.05	96.97%
Euclidiana	$\sqrt{\sum(x_i - y_i)^2}$	0.01	4.55%
		0.03	36.36%
		0.05	46.97%
Manhattan	$\sum \lvert x_i - y_i \rvert$	0.01	4.55%
		0.03	36.36%
		0.05	46.97%

Fonte: Elaborado pelo autor (2025).

[18] Achatamento do vetor de coordenadas com formato (n_quadros, 17, 2) para (n_quadros, 34,).

Na avaliação da métrica mais adequada para este problema, foram testados limiares intermediários (0.005, 0.01, 0.03) para a Distância Normalizada, considerados valores moderadamente permissivos, conforme a Tabela 1. Esses limiares foram definidos para as diferentes métricas de distância (Euclidiana ao Quadrado, Euclidiana e Manhattan), com base no princípio de que valores menores de distância indicam maior similaridade entre os movimentos comparados. Dessa forma, foi possível avaliar qual métrica apresentou melhor desempenho sobre o conjunto total de amostras. A Tabela 1 apresenta a porcentagem de amostras cuja distância ficou abaixo dos limiares estabelecidos para cada métrica avaliada.

Para quantificar o desempenho em cada fórmula de distância, foi calculada a porcentagem de bons resultados, dada pela equação:

$$\text{Porcentagem de bons resultados} = \frac{N_{\text{bons}}}{N_{\text{total}}} \times 100 \tag{4.1}$$

Onde:

- $Nbons$ é o número de amostras cuja métrica selecionada está abaixo do limiar definido.
- $Ntotal$ é o número total de amostras analisadas.

Desse modo, conforme apresentado na Tabela 1, observa-se que as distâncias Euclidiana e Manhattan apresentaram desempenhos idênticos, com percentuais de acerto consideravelmente mais baixos em comparação à distância Euclidiana ao Quadrado. Esta última apresentou resultados substancialmente superiores com base nos limiares definidos.

4.5.2 Métricas de Saídas

A saída final é composta por quatro métricas principais:

1. Distância DTW bruta, que representa o custo total acumulado do alinhamento:

$$D_{\text{bruta}} = \sum_{(i,j)\in \text{path}} d(a_i, b_j) \tag{4.2}$$

2. Distância Normalizada, que ajusta esse valor pela extensão do caminho ótimo, permitindo comparação entre sequências de tamanhos distintos:

$$D_{\text{normalizada}} = \frac{D_{\text{bruta}}}{\text{len(path)}} \tag{4.3}$$

3. Erro médio ponto a ponto, média das diferenças absolutas entre os pontos alinhados, calculado após a obtenção do caminho de alinhamento:

$$\text{Erro médio} = \frac{1}{N} \sum_{(i,j)\in \text{path}} |a_i - b_j| \tag{4.4}$$

4. Desvio padrão, que expressa a variabilidade desses erros ao longo do tempo:

$$\text{Desvio} = \sqrt{\frac{1}{N} \sum_{(i,j)\in \text{path}} \left(|a_i - b_j| - \text{Erro médio} \right)^2} \tag{4.5}$$

4.5.3 Ferramentas da Etapa

4.5.3.1 dtw-python

Este projeto faz uso da biblioteca *dtw-python*, uma implementação em código aberto do algoritmo *Dynamic Time Warping*. O DTW é uma técnica poderosa para comparar séries temporais que podem ter variações na velocidade ou no ritmo, sendo muito usada em áreas como análise de movimentos, reconhecimento de padrões e mineração de dados temporais. Ele permite alinhar duas sequências de forma flexível, calculando tanto a distância entre elas quanto o melhor caminho de alinhamento (dtw, 2025).

5 RESULTADOS

Neste capítulo, são apresentados e discutidos os resultados obtidos nos experimentos realizados. Os dados foram organizados em tabelas e gráficos para facilitar a análise e a comparação entre os diferentes golpes, lados, articulações e métricas avaliadas.

5.1 Resultado FFT

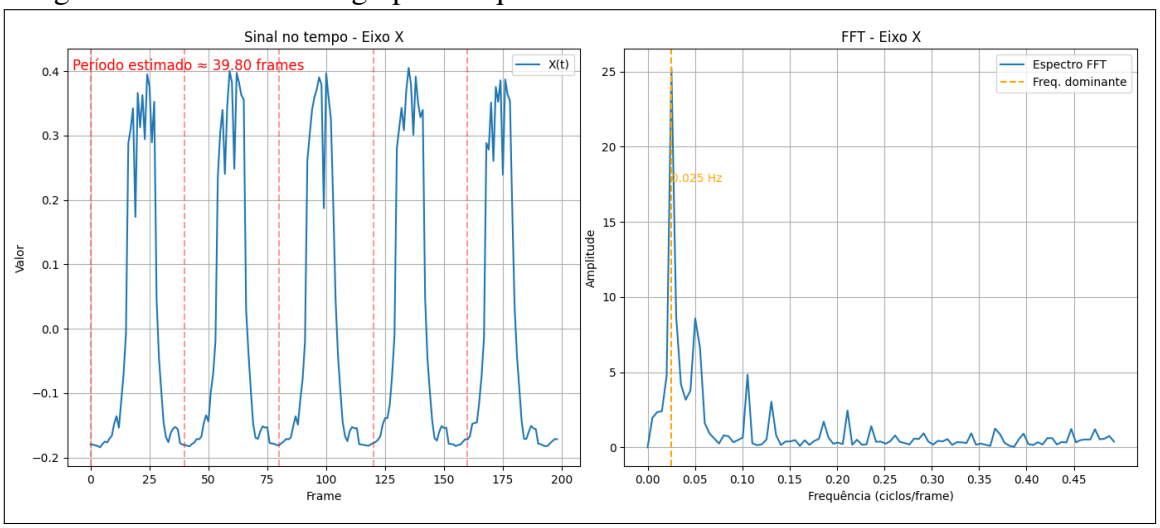
Para validar a consistência dos movimentos nos conjuntos de dados criados, foi realizada uma análise temporal focada em uma articulação específica, representando o principal ponto de contato do golpe. Essa análise permitiu identificar a frequência dominante e o período médio dos ciclos dos golpes.

Embora essa análise de frequência não componha diretamente a pipeline principal do processamento, sua aplicação foi fundamental como ferramenta complementar de inspeção visual. A avaliação dos gráficos temporais e espectrais contribuiu para verificar a consistência dos golpes registrados nos vídeos, oferecendo uma perspectiva adicional sobre a qualidade dos movimentos analisados.

5.1.1 Conjunto de dados de Referência

5.1.1.1 Analise Horizontal (Eixo X)

Figura 16 – Ciclo de um golpe e frequência dominante no eixo X - dados de referência



Fonte: Elaborado pelo autor (2025).

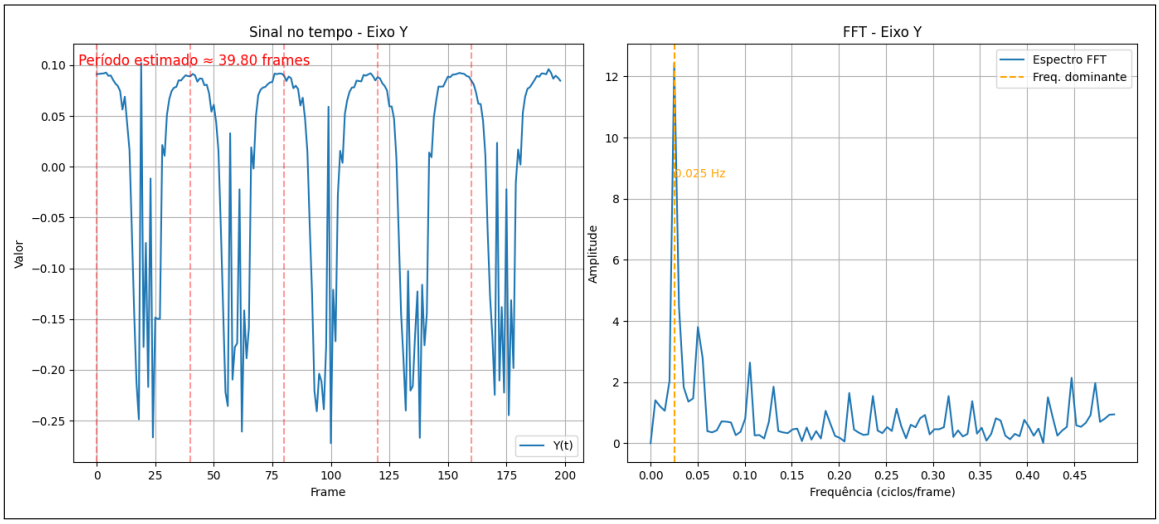
A Figura 16 mostra a análise do deslocamento da articulação ao longo do eixo X,

que corresponde ao movimento horizontal. No gráfico da esquerda, observa-se um padrão cíclico bem definido, com picos espaçados de forma regular, indicando uma execução estável do movimento, indicando que que há uma repetição consistente do gesto ao longo do tempo. O ciclo de execução do golpe foi detectado a cada 39.80 quadros.

O gráfico da direita apresenta o espectro de frequência desse deslocamento horizontal. Nota-se um pico dominante em aproximadamente 0,025 Hz[1], evidenciando que o movimento ocorre de maneira rítmica, com baixa presença de ruídos e variações não periódicas.

5.1.1.2 Analise Vertical (Eixo Y)

Figura 17 – Ciclo de um golpe e frequência dominante no eixo Y - dados de referência



Fonte: Elaborado pelo autor (2025).

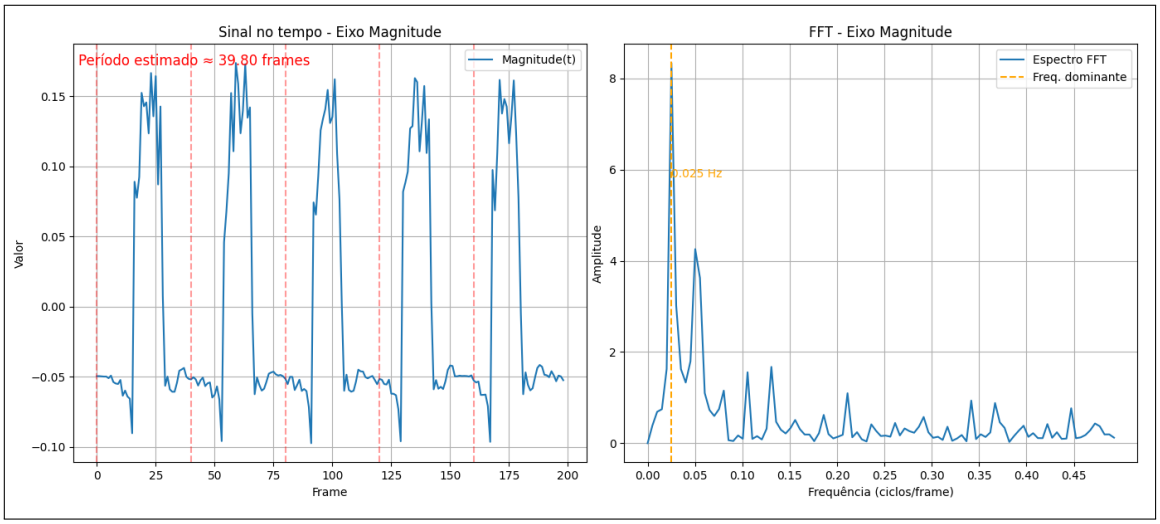
A Figura 17 apresenta a análise do deslocamento ao longo do eixo Y, que representa o movimento vertical da articulação. O gráfico da esquerda mostra oscilações suaves e regulares, com um padrão de repetição bem definido. O período estimado também gira em torno de 39.80 quadros, que embora não esteja detectando um ciclo completo de um golpe, o padrão de repetição observado ainda indica sincronia entre os movimentos verticais e horizontais.

No gráfico da direita, o espectro de frequência dominante o mesmo valor encontrado na Seção 5.1.1.1. Reforçando a regularidade do golpe ao longo do tempo, essa consistência sugere uma boa execução técnica presente no conjunto de referência.

[1] Hz: unidade de medida de frequência

5.1.1.3 Analise da Magnitude Vetorial (X + Y)

Figura 18 – Ciclo de um golpe e frequência dominante da magnitude vetorial do movimento - dados de referência



Fonte: Elaborado pelo autor (2025).

A Figura 18 mostra a análise da magnitude vetorial [2] do deslocamento. Mostrando um padrão de repetição cíclico evidente, com picos de alturas e intervalos similares. O ciclo de um golpe foi detectado no período final é de 39.80 quadros, em que cada pico apresenta forma aproximadamente simétrica. Esse comportamento reforça a qualidade de execução dos golpes no conjunto de dados de referência.

Já o gráfico da direita apresenta um espectro de frequência com pico dominante também em 0,025 Hz, indicando que a maior parte da energia está concentrada em uma única frequência. Isso sugere que o movimento é executado de forma consistente e sem variações significativas, características esperadas de golpes bem cadenciados.

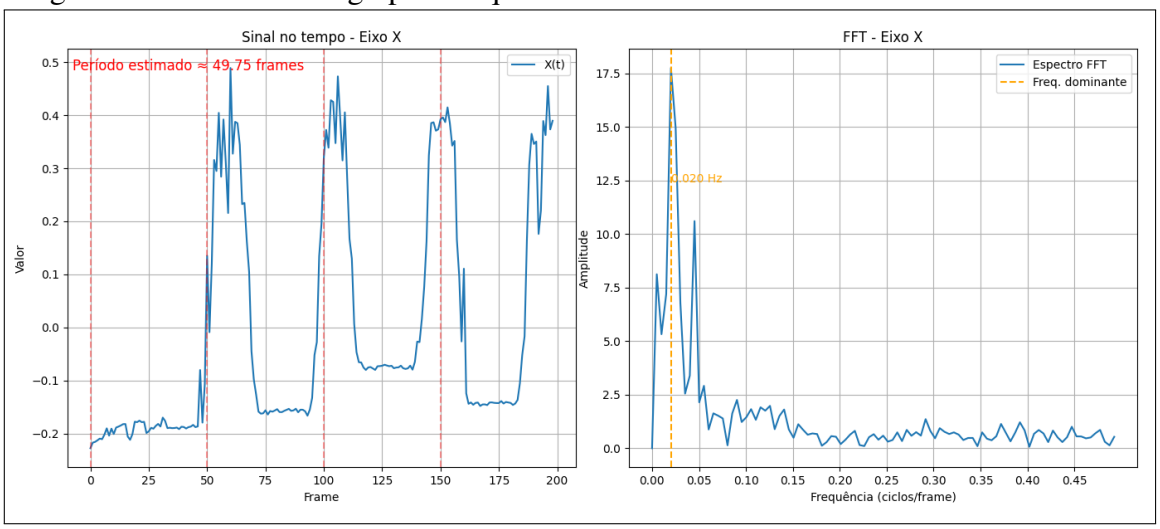
5.1.2 Conjunto de dados de Teste

5.1.2.1 Análise Horizontal (Eixo X)

A Figura 19 apresenta a análise do movimento ao longo do eixo X, correspondente ao deslocamento horizontal da articulação, com base nos dados de teste. O gráfico à esquerda revela um padrão cíclico perceptível, embora possuam picos de altura e intervalo variáveis. Isso indica que, apesar das variações observadas, há uma regularidade subjacente na execução

[2] Neste contexto, magnitude refere-se ao deslocamento total da articulação no plano, calculado a partir da combinação dos eixos X e Y, oferecendo uma medida integrada da movimentação.

Figura 19 – Ciclo de um golpe e frequência dominante no eixo X — dados de teste



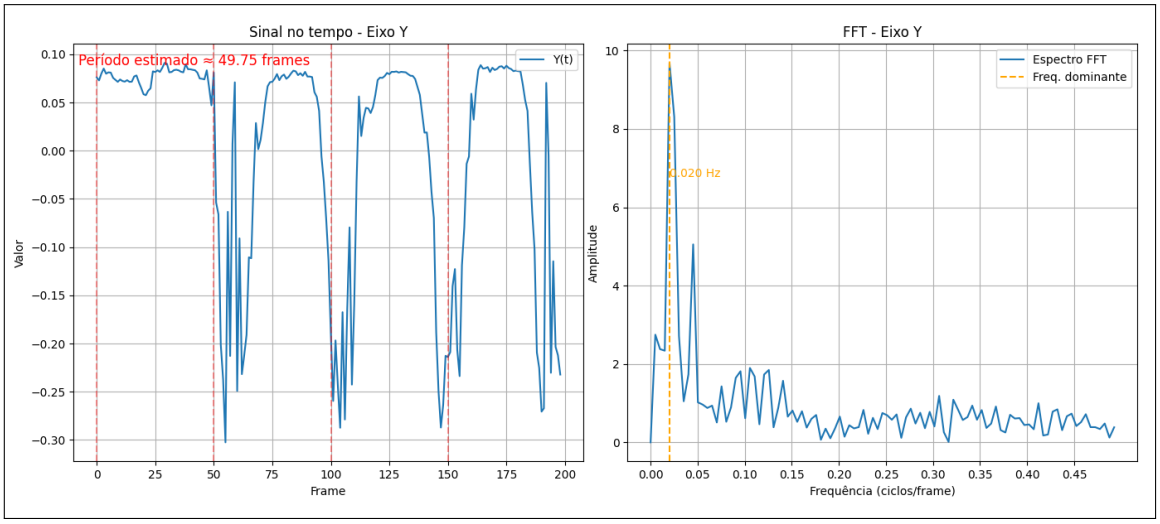
Fonte: Elaborado pelo autor (2025).

do movimento. O período estimado é de aproximadamente 49.75 quadros, na qual o ciclo de execução de um golpe não foi detectado com precisão devido as inconsistências presentes na amostra.

No gráfico da direita, identifica-se uma frequência dominante em torno de 0,020 Hz. Contudo, a presença de ruídos e de picos secundárias com menor amplitude sugere certa instabilidade no ritmo do movimento, possivelmente associada a movimentos desnecessários ou inconsistências durante a execução do golpe.

5.1.2.2 Análise Vertical (Eixo Y)

Figura 20 – Ciclo de um golpe e frequência dominante no eixo Y — dados de teste



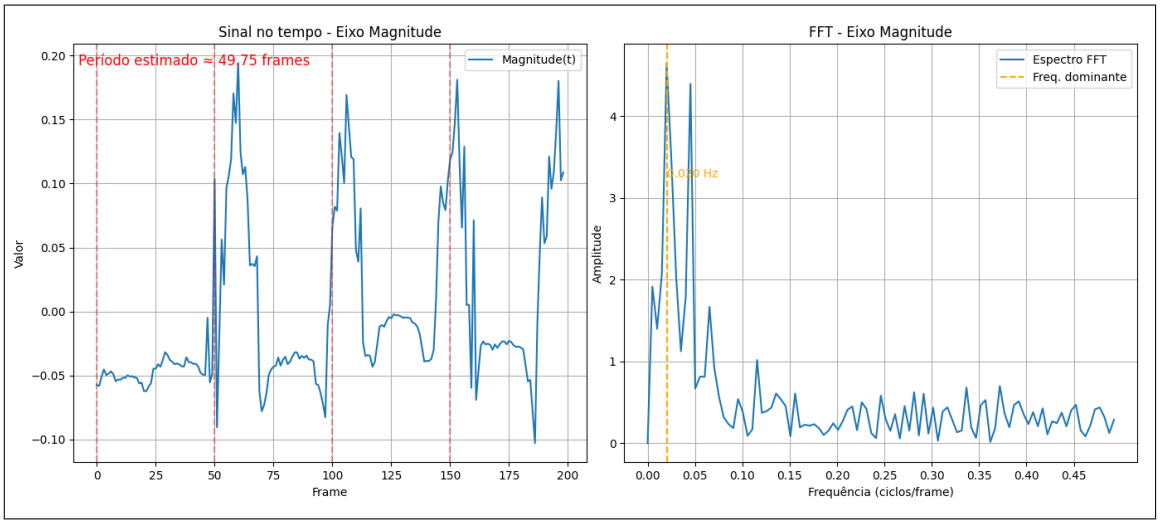
Fonte: Elaborado pelo autor (2025).

A Figura 20 apresenta a variação do deslocamento vertical da articulação ao longo do tempo, ou seja, ao longo do eixo Y. No gráfico da esquerda, observa-se um padrão com picos mais variados e intervalos irregulares entre os ciclos, indicando maior variabilidade na execução do movimento. O período estimado se mantém próximo de 49.75 quadros, na qual o ciclo do golpe não foi detectado com precisão devido o padrão de repetição desregular.

No gráfico da direita, o maior pico possui 0.020 Hz [3], indicando a frequência dominante do movimento vertical. Também aparecem picos secundários, que evidencia uma maior dispersão espectral. Esse comportamento pode ser consequência de variações na execução do movimento, perda de regularidade na execução e ruídos durante a gravação

5.1.2.3 Análise da Magnitude Vetorial (X + Y)

Figura 21 – Ciclo de um golpe e frequência dominante da magnitude vetorial do movimento — dados de teste



Fonte: Elaborado pelo autor (2025).

A Figura 21 apresenta a evolução da magnitude vetorial [4] do movimento. No gráfico da esquerda, que representa o golpe no domínio do tempo, nota-se picos com amplitudes crescentes, intervalos oscilado e formato assimétrico. O período estimado final é 49,75 quadros. O padrão repetitivo, indica menor uniformidade entre os ciclos de um golpe.

No gráfico da direita, a frequência dominante continua centrada em 0,020 Hz. No entanto, nota-se há presença de várias frequências secundárias que ameaçam esse valor domi-

[3] Hz: unidade de medida de frequência.

[4] Neste contexto, magnitude refere-se ao deslocamento total da articulação no plano, calculado a partir da combinação dos eixos X e Y, oferecendo uma medida integrada da movimentação.

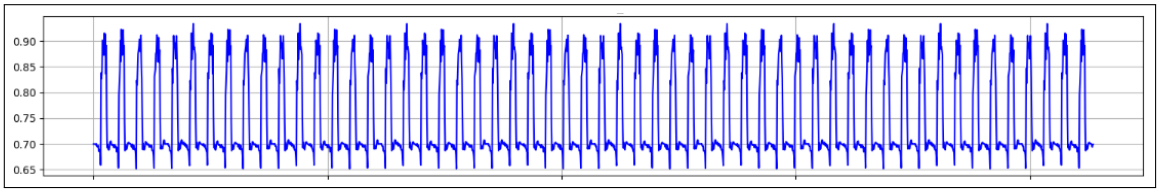
nante, reforçando a ideia de inconsistência na execução do golpe e indicando forte presença de ruídos nas amostras de teste.

5.2 Resultado DTW

5.2.1 Análise

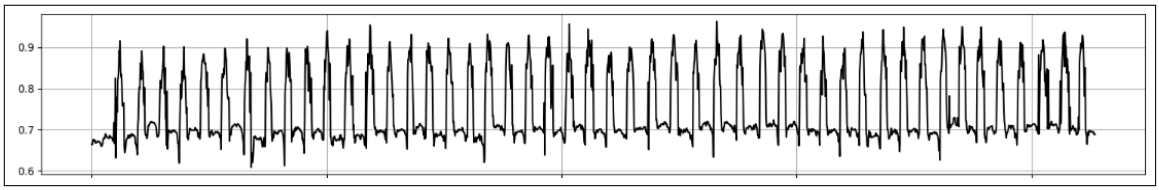
As amostras foram submetidas a um processo de alinhamento temporal (Seção 4.4) visando reduzir a distância entre as amostras de referência e teste, minimizando a influência de variações temporais. Assim, características como cadência, ângulo e controle de execução passam a ser os principais responsáveis pelas diferenças observadas no alinhamento.

Figura 22 – Comportamento da articulação do tornozelo direito ao longo do tempo em uma amostra de referência de um chute alto direito na postura destra



Fonte: Elaborado pelo autor (2025).

Figura 23 – Comportamento de uma articulação do tornozelo direito ao longo do tempo em uma amostra de teste de um chute alto direito na postura destra

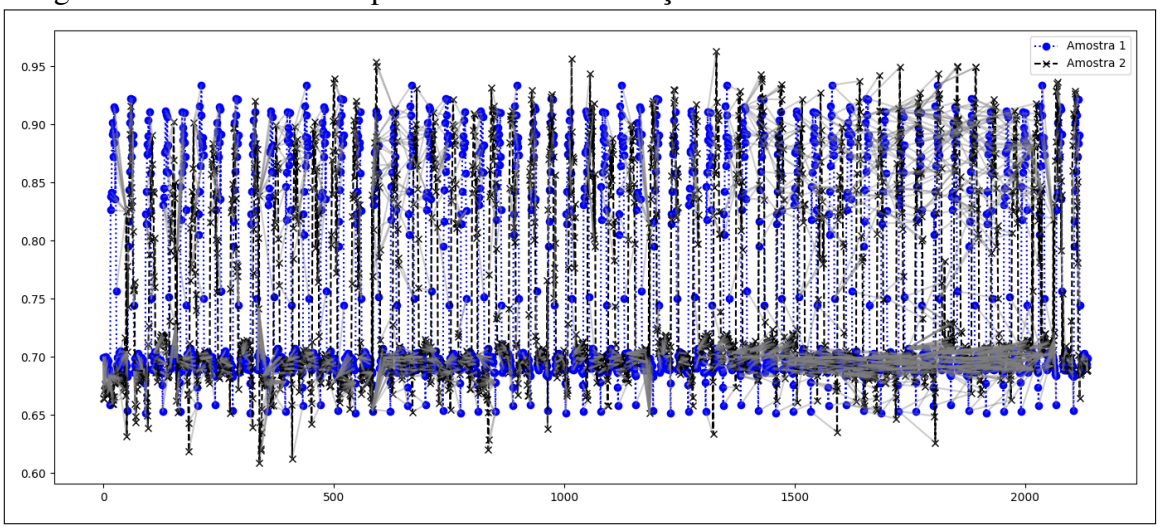


Fonte: Elaborado pelo autor (2025).

Com as séries temporais do movimento usadas como argumentos de entrada, o algoritmo foi computado para realizar o cálculo da distância entre as amostras analisadas. A Figura 24 apresenta o resultado do alinhamento temporal entre as sequências consideradas. Os pares de pontos entre as amostras foram ajustados para manter a correspondência entre os padrões de movimento, mesmo diante de algumas variações na execução. As linhas de conexão indicam que, apesar das diferenças observadas entre as amostras, o algoritmo conseguiu alinhar segmentos com estrutura similar, minimizando a distância global.

A Figura 25 mostra o caminho de alinhamento DTW da articulação referente ao ponto de contato do golpe entre duas amostras. A linha verde indica os pares de pontos correspondentes.

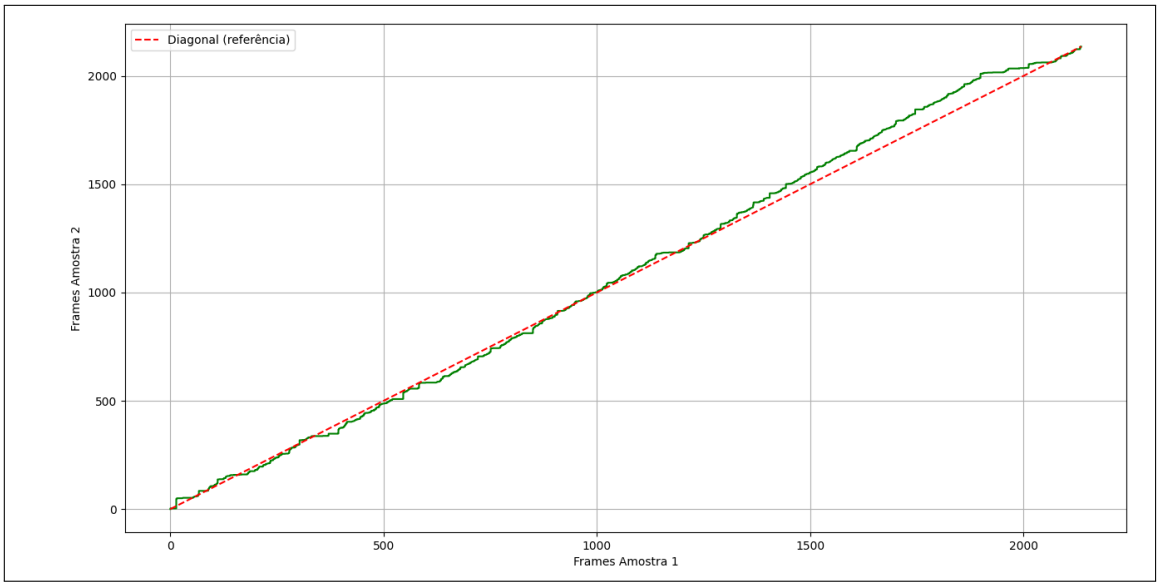
Figura 24 – Distância computada de uma articulação entre duas amostras analisadas



Fonte: Elaborado pelo autor (2025).

A proximidade desse caminho em relação à diagonal de referência sugere alta similaridade entre as sequências, enquanto desvios indicam distorções temporais na execução do movimento. Essas diferenças podem ser atribuídas a variações na cadência, amplitude ou controle motor.

Figura 25 – Caminho de alinhamento DTW entre duas amostras



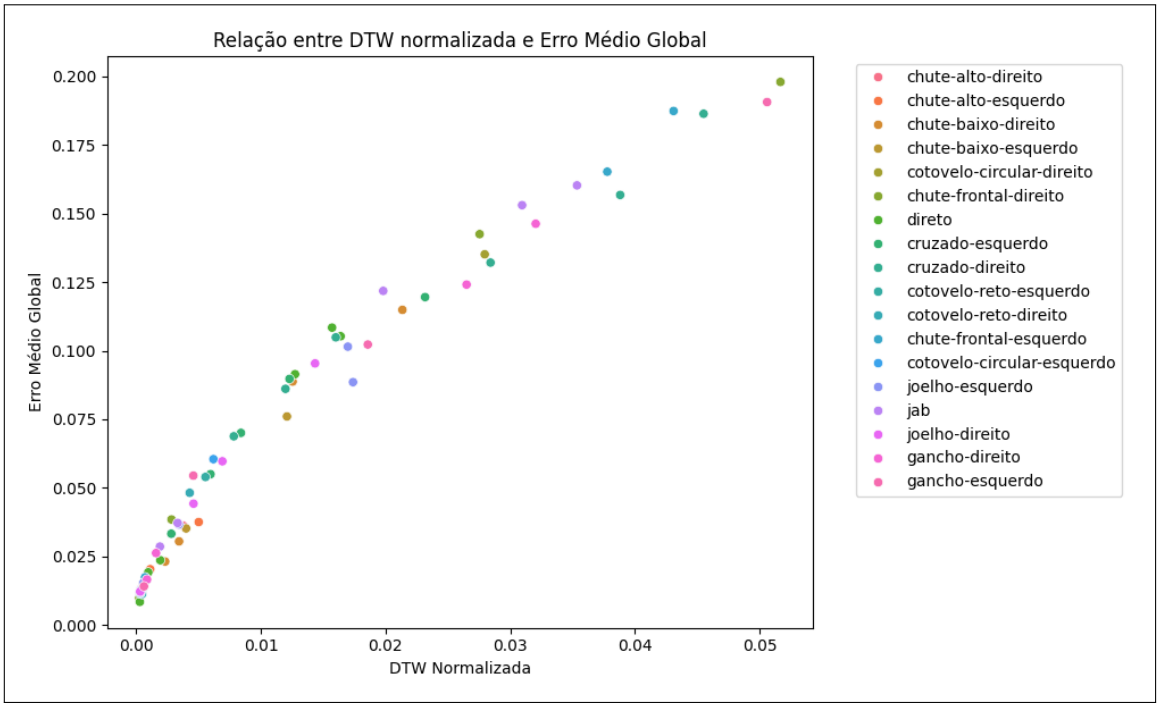
Fonte: Elaborado pelo autor (2025).

5.2.2 Análise Quantitativa

A Figura 26 apresenta os pontos distribuídos ao longo de uma curva crescente, formando uma linha suave que indica que, à medida que a distância normalizada aumenta, o erro médio também se eleva. O fator de qualidade dessas métricas é que, quanto mais próximo de

zero, melhor — ou seja, menores valores de distância e erro médio indicam maior similaridade. Parte dos pontos está concentrada na região inferior esquerda, sugerindo alto grau de similaridade entre as amostras de referência e teste, enquanto a outra parte se dispersa da metade para a parte superior. Pode-se concluir que os pontos aglomerados correspondem às comparações com o lado destro do conjunto de referência, enquanto os mais dispersos referem-se ao lado canhoto.

Figura 26 – Gráfico de dispersão; Distância normalizada X Erro médio

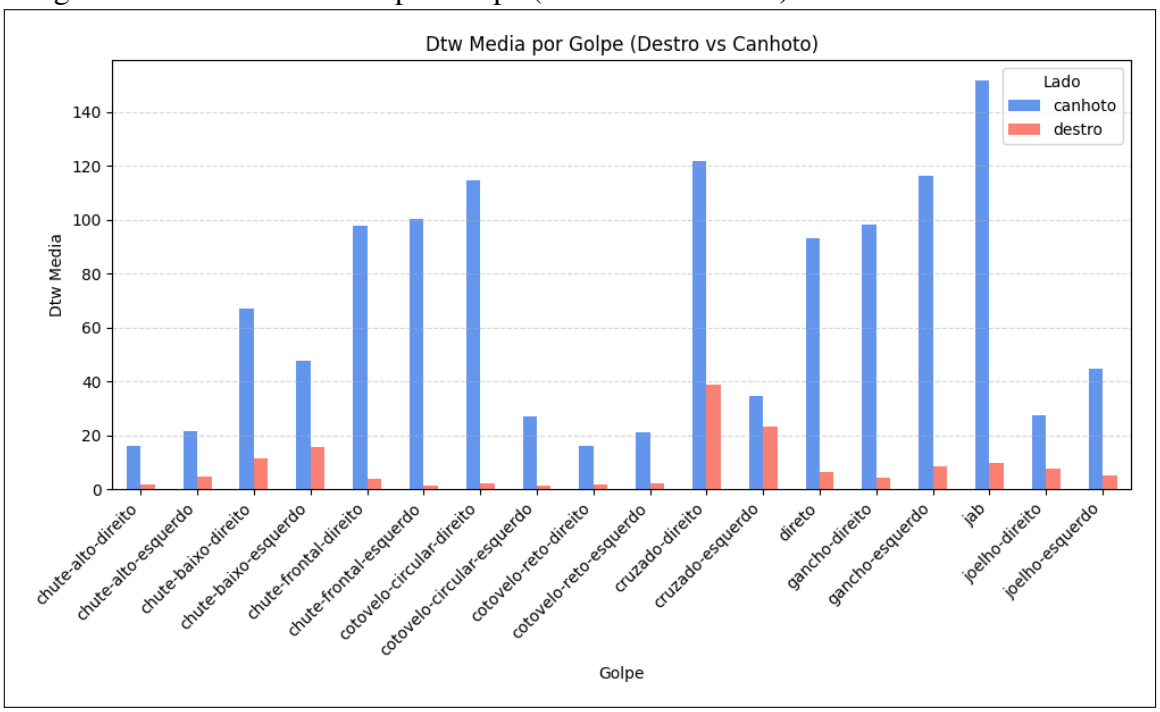


Fonte: Elaborado pelo autor (2025).

As Figuras 27, 28 e 29 apresentam o comportamento das amostras em relação às métricas avaliadas: distância bruta, distância normalizada e erro médio. Como já mencionado, o conjunto de testes foi gravado com a postura destra como dominante, o que resulta na média de valores mais baixos para essa postura. Isso indica que as sequências executadas na postura destra apresentam boa similaridade entre os movimentos de referência e os de teste.

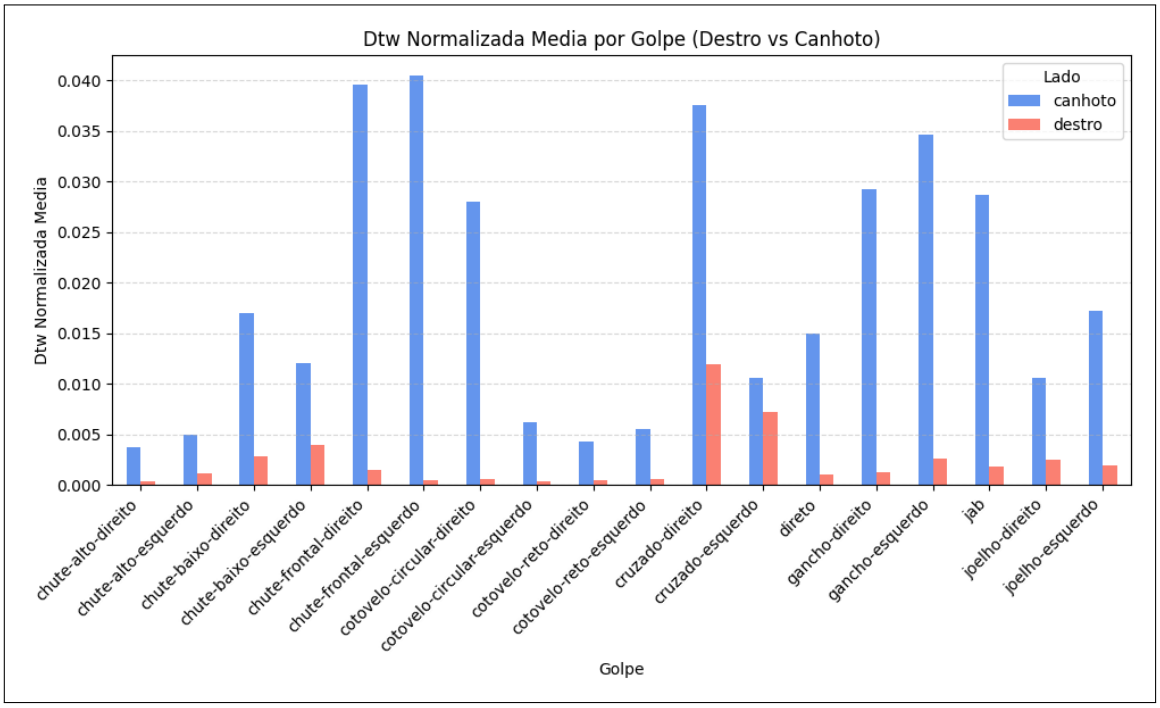
A Figura 30 apresenta os valores de desvio para cada golpe. Os valores de desvio global tendem a ser mais elevados do que os observados em outras métricas analisadas. Isso ocorre porque o desvio global representa a dispersão dos dados, revelando possíveis inconsistências ou instabilidades nos ciclos de execução dos golpes.

Figura 27 – Distância Bruta por Golpe (Destro vs Canhoto)



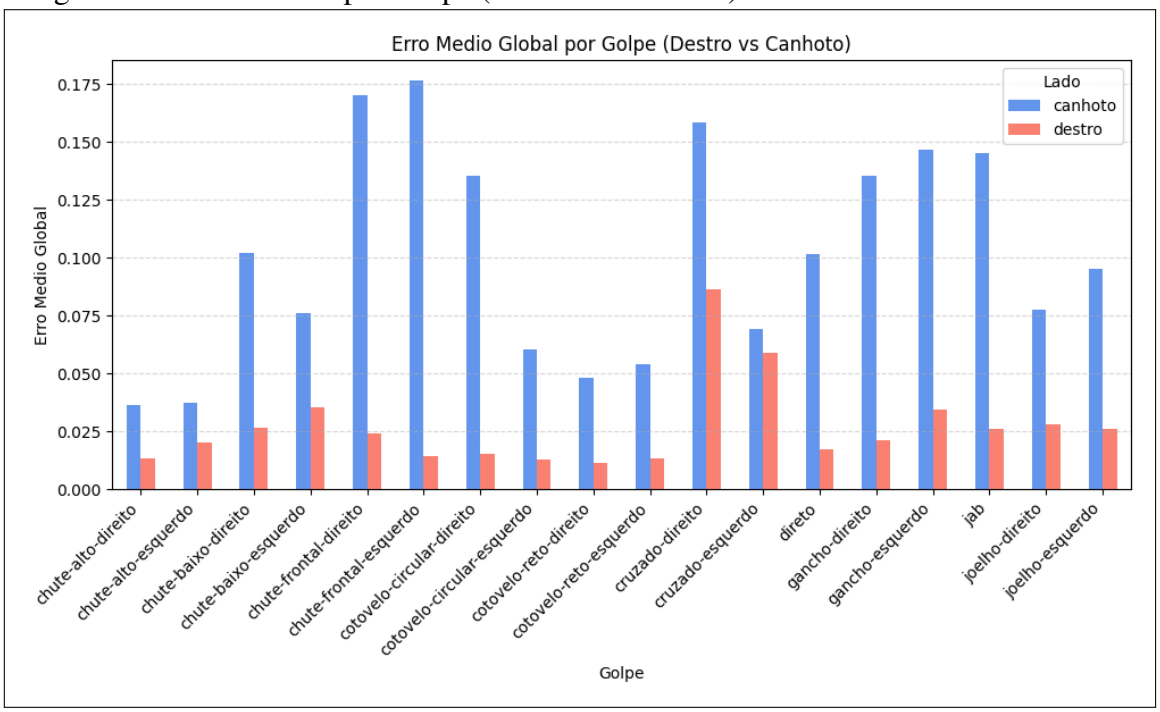
Fonte: Elaborado pelo autor (2025).

Figura 28 – Distância Normalizada por Golpe (Destro vs Canhoto)



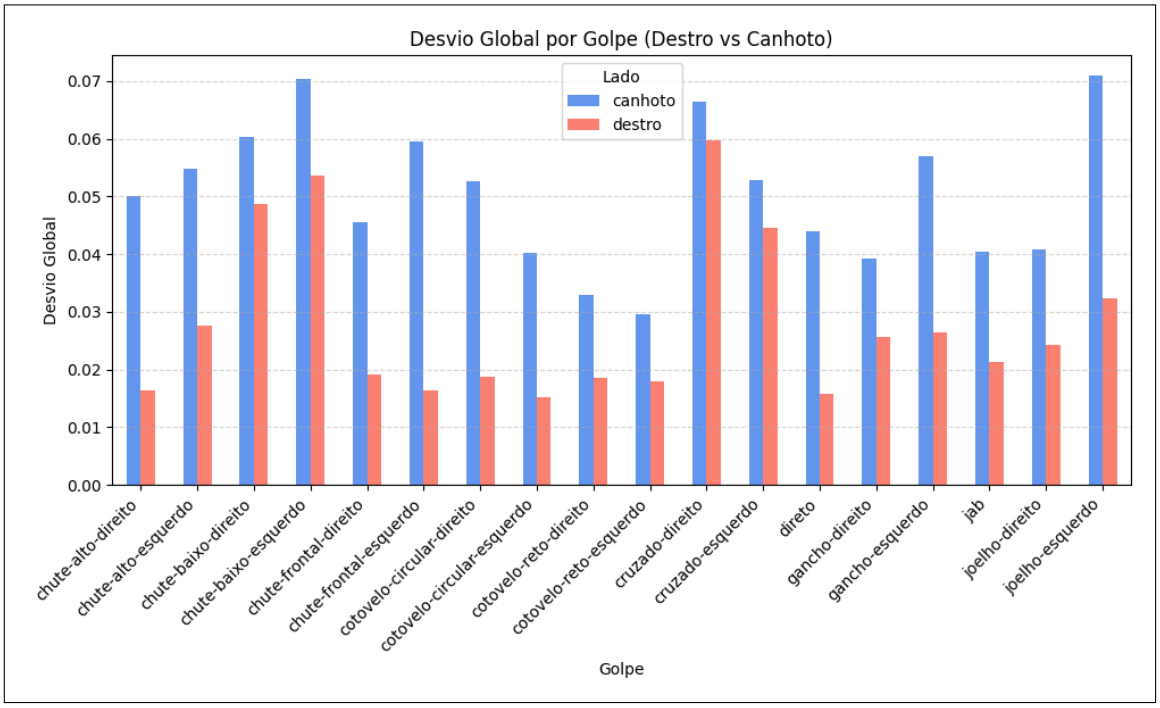
Fonte: Elaborado pelo autor (2025).

Figura 29 – Erro Médio por Golpe (Destro vs Canhoto)



Fonte: Elaborado pelo autor (2025).

Figura 30 – Desvio Global por Golpe (Destro vs Canhoto)



Fonte: Elaborado pelo autor (2025).

Ao se adotar um limiar de 0.005 para a distância normalizada, apenas 46,97% das amostras foram classificadas como suficientemente similares, conforme definido na Equação 4.1. Esse percentual abaixo de 50% indica que a maior parte dos movimentos analisados apresentou variações superiores ao tolerado por esse critério rigoroso. Tal resultado era esperado, dado que as amostras de teste naturalmente incorporam ruídos e variações na execução, o que eleva a distância mesmo em movimentos tecnicamente corretos. Portanto, o fato de menos da metade das amostras satisfazerem o limiar não caracteriza falha do modelo, mas evidencia que o valor adotado é bastante restritivo para contextos práticos. A Figura 31 apresenta a proporção de classificações positivas obtidas com quatro diferentes valores de limiar: 0.005, 0.01, 0.03 e 0.05.

Figura 31 – Porcentagem de melhores resultados entre limiares 0.005, 0.01, 0.03 e 0.05.)

Métrica	Expressão Matemática	Limiar	Porcentagem
Euclidiana ao Quadrado	$\sum(x_i - y_i)^2$	0.005	46.97%
		0.01	57.58%
		0.03	86.36%
		0.05	96.97%

Fonte: Elaborado pelo autor (2025).

A Tabela 2 apresenta os melhores resultados globais para cada tipo de golpe comparado, considerando as posturas destra e canhota. De forma geral, os valores de distância e erro médio tendem a ser menores para a postura destra (conforme os dados utilizados na Seção 4.1.1.1). Ou seja, dentre as comparações realizadas entre os conjuntos de dados de referência e teste, esses resultados indicam maior similaridade entre as amostras.

Os golpes executados na postura destra apresentam uma variabilidade na distância bruta entre 0.5922 e 25.3449, sendo essas as menores distâncias computadas pelo DTW. Já na postura canhota, os valores variam entre 9.1078 e 114.7682. Embora os valores sejam mais elevados, o modelo DTW ainda foi capaz de identificar similaridade em algumas comparações entre amostras destras x canhotas, considerando o conjunto de testes ser executado na postura destra.

Golpes como o chute-frontal-direito e o joelho-direito, na postura destra, apresentaram os menores valores de distância e erro, sugerindo execuções mais consistentes e próximas das amostras de referência. Por outro lado, movimentos como o cotovelo-circular-direito e o jab, na postura canhota, exibiram maiores distâncias e variações, indicando maior dissimilaridade na execução e instabilidade nos ciclos temporais dos golpes.

Tabela 2 – Médias dos melhores resultados filtrados por distância normalizada para cada classe

Golpe	Lado	DTW Bruto	DTW Norm.	Erro Médio	Desvio
chute-alto-direito	destro	1.7746	0.000415	0.013331	0.016461
chute-alto-direito	canhoto	16.1430	0.003779	0.036173	0.050081
chute-alto-esquerdo	destro	4.7939	0.001122	0.020338	0.027654
chute-alto-esquerdo	canhoto	21.4065	0.005011	0.037541	0.054850
chute-baixo-direito	destro	9.1949	0.002321	0.023134	0.045406
chute-baixo-direito	canhoto	49.6851	0.012540	0.088767	0.051136
chute-baixo-esquerdo	destro	15.8419	0.003998	0.035210	0.053591
chute-baixo-esquerdo	canhoto	47.8841	0.012086	0.076042	0.070446
chute-frontal-direito	destro	0.5922	0.000240	0.009816	0.012039
chute-frontal-direito	canhoto	68.0700	0.027536	0.142529	0.045479
chute-frontal-esquerdo	destro	0.7783	0.000315	0.011484	0.014722
chute-frontal-esquerdo	canhoto	93.3737	0.037773	0.165299	0.065970
cotovelo-circular-direito	destro	2.2908	0.000558	0.015122	0.018816
cotovelo-circular-direito	canhoto	114.7682	0.027945	0.135170	0.052724
cotovelo-circular-esquerdo	destro	1.5800	0.000363	0.012682	0.015214
cotovelo-circular-esquerdo	canhoto	26.9057	0.006189	0.060475	0.040190
cotovelo-reto-direito	destro	1.7594	0.000467	0.011257	0.018521
cotovelo-reto-direito	canhoto	16.1397	0.004288	0.048195	0.032852
cotovelo-reto-esquerdo	destro	2.0457	0.000537	0.013496	0.017974
cotovelo-reto-esquerdo	canhoto	21.1356	0.005552	0.053965	0.029630
cruzado-direito	destro	25.3449	0.007823	0.068805	0.053487
cruzado-direito	canhoto	92.0468	0.028410	0.132153	0.066396
cruzado-esquerdo	destro	2.6599	0.000821	0.016380	0.023972
cruzado-esquerdo	canhoto	9.1078	0.002811	0.033277	0.036526
direto	destro	1.8045	0.000289	0.008429	0.012708
direto	canhoto	79.4581	0.012734	0.091457	0.042951
gancho-direito	destro	2.9026	0.000864	0.016570	0.023641
gancho-direito	canhoto	89.0076	0.026490	0.124132	0.040314
gancho-esquerdo	destro	2.0752	0.000618	0.014085	0.019415
gancho-esquerdo	canhoto	62.3907	0.018569	0.102272	0.051297
jab	destro	1.7332	0.000328	0.011896	0.012853
jab	canhoto	104.5719	0.019805	0.121829	0.041209
joelho-direito	destro	0.9784	0.000314	0.012192	0.012807
joelho-direito	canhoto	17.9841	0.006906	0.059696	0.044892
joelho-esquerdo	destro	1.4757	0.000567	0.015316	0.019425
joelho-esquerdo	canhoto	44.1840	0.016968	0.101494	0.059427

Fonte: Elaborado pelo autor (2025).

6 CONCLUSÕES E TRABALHOS FUTUROS

A Visão Computacional mostra-se uma tecnologia extremamente promissora, capaz de impactar significativamente o cenário das artes marciais. Neste trabalho, foi conduzida uma análise detalhada da execução dos golpes do Muay Thai, a partir da extração e análise de pontos-chave extraídos de vídeos de treinamento. A metodologia proposta envolveu a aquisição cuidadosa de vídeos em diferentes posturas, o pré-processamento dos dados para garantir padronização temporal e espacial, além da aplicação de técnicas robustas de análise temporal.

O uso do YOLO11 como principal ferramenta para extração dos pontos-chave proporcionou a obtenção de dados precisos e consistentes das articulações do corpo humano em 2D, viabilizando uma análise mais objetiva e estruturada dos movimentos. A aplicação da Transformada Rápida de Fourier (FFT) possibilitou identificar padrões cíclicos e frequências dominantes nos golpes, contribuindo para uma melhor compreensão do ritmo e da cadência das execuções. O uso do Dynamic Time Warping (DTW) mostrou-se a abordagem mais eficiente para comparação de sequências temporais, permitindo distinguir variações entre diferentes tipos de golpes. Adotando um limiar rigoroso de similaridade, foi possível identificar que 46,97% das amostras comparadas apresentaram alto grau de proximidade com as referências estabelecidas.

Como perspectivas para trabalhos futuros, sugere-se utilizar os resultados obtidos nesta pesquisa como argumentos para o desenvolvimento de modelos baseados em inteligência artificial, visando uma análise automatizada, mais rápida e com maior precisão. Espera-se que tais modelos possam ser aplicados tanto em vídeos de treinamento em tempo real quanto em combates reais, onde a presença de ruídos e variações nos dados é significativamente maior. Também é proposto expandir o conjunto de dados com mais diversidade de atletas, utilizar estimativa de pose 3D para capturar profundidade e dinâmica dos movimentos, e implementar sistemas capazes de fornecer feedback em tempo real durante treinamentos e competições.

REFERÊNCIAS

BALLARD, D. H.; BROWN, C. M. : Computer vision. Prentice Hall Professional Technical Reference, 1982. Disponível em: https://www.scribd.com/doc/79984210/Ballard-D-and-Brown-C-M-1982-Computer-Vision. Acesso em: 14 abr. 2024.

BOX, G. E.; JENKINS, G. M.; REINSEL, G. C.; LJUNG, G. M. : Time series analysis: forecasting and control. John Wiley & Sons, 2015. Disponível em: https://download.e-bookshelf.de/download/0003/8810/69/L-G-0003881069-0007953902.pdf. Acesso em: 01 jul. 2025.

CHO, K.; MERRIËNBOER, B. V.; GULCEHRE, C.; BAHDANAU, D.; BOUGARES, F.; SCHWENK, H.; BENGIO, Y. : Learning phrase representations using rnn encoder-decoder for statistical machine translation. **arXiv preprint arXiv:1406.1078**, 2014.

CLIDEO. : clideo. 2025. https://clideo.com/pt. Acesso em: 18 nov. 2024.

DINIZ, P. S.; SILVA, E. A. da; NETTO, S. L. : Processamento digital de sinais- : Projeto e análise de sistemas. Bookman Editora, 2014. Disponível em: https://pdfcoffee.com/processamento-digital-de-sinais-projeto-e-analise-de-sistemas-pdf-free.html. Acesso em: 01 jul. 2025.

DRUMOND, J. G. d. F. : Tecnologia e esporte: perspectivas bioéticas. **Bioethikos**, v. 5, n. 4, p. 411–418, 2011.

DTW python. : python-dtw. 2025. https://pypi.org/project/dtw-python/. Acesso em: 10 jul. 2025.

FANG, H.-S.; LI, J.; TANG, H.; XU, C.; ZHU, H.; XIU, Y.; LI, Y.-L.; LU, C. : Alphapose: Whole-body regional multi-person pose estimation and tracking in real-time. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 2022.

GONZALEZ, R. C.; WOODS, R. E. : Processamento de imagens digitais. Englewood Cliffs, NJ, USA: Editora Blucher, 2000. Disponível em: https://www.cl72.org/090imagePLib/books/Gonzales,Woods-Digital.Image.Processing.4th.Edition.pdf.

GUPTA, V.; PATIL, C. : Human Pose Estimation using Keypoint RCNN in PyTorch. 2021. https://learnopencv.com/human-pose-estimation-using-keypoint-rcnn-in-pytorch/. Acesso em: 12 jul. 2025.

GUTIéRREZ, J. L. C. : Monitoramento da instrumentação da barragem de corumbá i por redes neurais e modelos de box & jenkins. Dissertação (Mestrado) – Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio), Rio de Janeiro, 2003. Programa de Pós-Graduação em Engenharia Civil, Mestrado em Ciências de Engenharia Civil, Área de Concentração: Geotecnia. Disponível em: https://www.maxwell.vrac.puc-rio.br/colecao.php?strSecao=especifico&nrSeq=4244@1.

HARRIS, C. R.; MILLMAN, K. J.; WALT, S. J. van der; GOMMERS, R.; VIRTANEN, P.; COURNAPEAU, D.; WIESER, E.; TAYLOR, J.; BERG, S.; SMITH, N. J.; KERN, R.; PICUS, M.; HOYER, S.; KERKWIJK, M. H. van; BRETT, M.; HALDANE, A.; RÍO, J. F. del; WIEBE, M.; PETERSON, P.; GÉRARD-MARCHANT, P.; SHEPPARD, K.; REDDY, T.; WECKESSER, W.; ABBASI, H.; GOHLKE, C.; OLIPHANT, T. E. : Array programming with NumPy. **Nature**, Springer Science and Business Media LLC, v. 585, n. 7825, p. 357–362, set. 2020. Disponível em: https://doi.org/10.1038/s41586-020-2649-2.

HECKBERT, P. : Fourier transforms and the fast fourier transform (fft) algorithm. **Computer Graphics**, v. 2, n. 1995, p. 15–463, 1995.

HOCHREITER, S.; SCHMIDHUBER, J. : Long short-term memory. **Neural computation**, MIT press, v. 9, n. 8, p. 1735–1780, 1997.

INSAFUTDINOV, E.; PISHCHULIN, L.; ANDRES, B.; ANDRILUKA, M.; SCHIEKE, B. : Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In: . European Conference on Computer Vision (ECCV), 2016. Disponível em: http://arxiv.org/abs/1605.03170. Acesso em: 01 jun. 2025.

JABBR.AI. : Automatic Punch Stats Content Generation. 2024. https://jabbr.ai/. Acesso em: 28 abr. 2024.

JOCHER, G.; Q, B.; MUNAWAR, R.; CHAURASIA, A.; Q, L. : Pose estimation - ultralytics yolo8. 2024. https://docs.ultralytics.com/pt/tasks/pose/. Acesso em: 17 abr. 2024.

KENDALL, A.; GRIMES, M.; CIPOLLA, R. : Posenet: A convolutional network for real-time 6-dof camera relocalization. 2016. Disponível em: https://arxiv.org/abs/1505.07427.

LI, Y.; LIU, R. W.; LIU, Z.; LIU, J. : Similarity grouping-guided neural network modeling for maritime time series prediction. **IEEE Access**, Ieee, v. 7, p. 72647–72659, 2019.

MENDES-NEVES, T.; MEIRELES, L.; MENDES-MOREIRA, J. : A survey of advanced computer vision techniques for sports. 2023.

NIU, C. : The application of improved dtw algorithm in sports posture recognition. **Systems and Soft Computing**, Elsevier, v. 6, p. 200163, 2024.

PANDAS. : pandas. 2025. https://pandas.pydata.org/docs/index.html. Acesso em: 10 de julho de 2025.

SAKOE, H.; CHIBA, S. : Dynamic programming algorithm optimization for spoken word recognition. **IEEE transactions on acoustics, speech, and signal processing**, IEEE, v. 26, n. 1, p. 43–49, 1978.

SANTOS, R. F. dos; LEITE, T. L. do C.; LIMA, B. N.; MANESCHY, M. de S.; JUNIOR, R. S. de A.; ALMEIDA, K. da S.; PASSOS, R. P.; JUNIOR, G. d. B. V. : Capacidades físicas na prática do muay thai. **Revista CPAQV–Centro de Pesquisas Avançadas em Qualidade de Vida| Vol**, v. 13, n. 3, p. 2, 2021.

SIMON, T.; JOO, H.; MATTHEWS, I.; SHEIKH, Y. : Hand keypoint detection in single images using multiview bootstrapping. In: . Computer Vision and Pattern Recognition Conference (CVPR), 2017. Disponível em: https://doi.org/10.48550/arXiv.1704.07809. Acesso em: 19 dez. 2023.

SOLEM, J. E. : Programming computer vision with python: Tools and algorithms for analyzing images. "O'Reilly Media, Inc.", 2012. Disponível em: https://www.oreilly.com/library/view/programming-computer-vision/9781449341916/. Acesso em: 02 jun. 2024.

SUN, K.; XIAO, B.; LIU, D.; WANG, J. : Deep high-resolution representation learning for human pose estimation. In: . Computer Vision and Pattern Recognition (CVPR), 2019. Disponível em: https://arxiv.org/abs/1902.09212. Acesso em: 20 dez. 2024.

SZELISKI, R. : Computer vision: algorithms and applications. Springer Nature, 2022. Disponível em: https://szeliski.org/Book/. Acesso em: 15 jun. 2024.

VICENTE, C. M. de S.; NASCIMENTO, E. R.; EMERY, L. E. C.; FLOR, C. A. G.; VIEIRA, T.; OLIVEIRA, L. B. : High performance moves recognition and sequence segmentation based on key poses filtering. In: IEEE. 2016. p. 1–8. Disponível em: https://homepages.dcc.ufmg.br/~erickson/publications/vicente_wacv2016.pdf. Acesso em: 14 jan. 2024.

WARPING, D. T. : Dynamic Time Warping. 2025. https://medium.com/@markstent/dynamic-time-warping-a8c5027defb6/. Acesso em: 08 mai.2025.

XU, H.; BAZAVAN, E. G.; ZANFIR, A.; FREEMAN, W. T.; SUKTHANKAR, R.; SMINCHISESCU, C. : Ghum & ghuml: Generative 3d human shape and articulated pose models. In: . IEEE, 2020. p. 6184–6193. Disponível em: https://ieeexplore.ieee.org/document/9157563. Acesso em: 15 jun. 2025.

YADAV, M.; ALAM, M. A. : Dynamic time warping (dtw) algorithm in speech: a review. **International Journal of Research in Electronics and Computer Engineering**, v. 6, n. 1, p. 524–528, 2018.

YOLO10. : YOLO VERSÃO 10. 2024. https://docs.ultralytics.com/pt/models/yolov10/. Acesso em: 01 jul. 2025.

YOLO11. : YOLO VERSÃO 11. 2024. https://docs.ultralytics.com/pt/models/yolo11/. Acesso em: 01 jul. 2025.

YOLO12. : YOLO VERSÃO 12. 2025. https://docs.ultralytics.com/pt/models/yolo12/. Acesso em: 01 jul. 2025.

ZHENG, C.; WU, W.; CHEN, C.; YANG, T.; ZHU, S.; SHEN, J.; KEHTARNAVAZ, N.; SHAH, M. : Deep learning-based human pose estimation: A survey. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 56, n. 1, aug 2023. ISSN 0360-0300. Disponível em: https://doi.org/10.1145/3603618.