

# TREINE CERTO: APLICAÇÃO PARA ANÁLISE DA CORRETEDE DE EXERCÍCIOS FÍSICOS UTILIZANDO APRENDIZADO DE MÁQUINA

Luiz Gustavo Klitzke, Aurélio Faustino Hoppe – Orientador, Diego Rafael Eising –  
Coorientador

Curso de Bacharel em Ciência da Computação  
Departamento de Sistemas e Computação  
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brasil

lgklitzke@furb.br, aureliof@furb.br, diegoskel@hotmail.com

**Resumo:** Com o recente aumento da adesão à prática de atividades físicas, cresce também o número de praticantes inexperientes treinando por conta própria ou sem acompanhamento adequado. Nesse contexto, a execução incorreta de exercícios torna-se comum, elevando o risco de lesões e comprometendo os resultados. Considerando que o acompanhamento profissional nem sempre é acessível, este artigo apresenta uma aplicação desenvolvida para realizar, automaticamente, a classificação da execução de exercícios físicos e sua correteza, utilizando processamento de imagens, técnicas de visão computacional e aprendizado de máquina. Para isso, foram utilizados recursos da biblioteca OpenPose, que permite a extração de pontos-chave do corpo humano a partir de vídeos, servindo como entrada para uma rede neural híbrida, composta por um modelo de Rede Convolutiva de Grafos (GCN) temporal, combinado com uma rede Long Short-Term Memory (LSTM). O modelo foi treinado com um conjunto de 430 vídeos, capturados e classificados sob a supervisão de profissionais especializados. A aplicação alcançou uma acurácia de 99,66% nos dados de validação e foi avaliada experimentalmente por meio de um formulário aplicado a voluntários com perfis e níveis de experiência variados, evidenciando sua aceitação, usabilidade intuitiva e precisão nos resultados.

**Palavras-chave:** Exercícios físicos. Análise de correteza. Classificação. OpenPose. Rede Neural Convolutiva de Grafos. Long Short-Term Memory.

## 1 INTRODUÇÃO

Segundo Camargo e Añez (2020), a Organização Mundial da Saúde recomenda que a prática regular de atividade física é essencial para a prevenção de doenças não transmissíveis, doenças cardiovasculares e diversos tipos de câncer, além de beneficiar a saúde mental, prevenindo declínios cognitivos e sintomas de depressão e ansiedade. A mesma organização recomenda entre 150 e 300 minutos semanais de atividade física moderada.

De acordo com dados obtidos pela Pesquisa Nacional de Saúde de 2019 (Instituto Brasileiro de Geografia e Estatística, 2021) e evidenciados por Silva (2021), 40,3% dos brasileiros em geral e 59,7% dos idosos não atendem tais recomendações. No entanto, o Ministério da Saúde (2022), ressalta que houve um aumento no número de adultos que praticam atividades físicas por pelo menos 150 minutos semanais em seu tempo livre, de 30,3% em 2009 para 36,7% em 2021.

Brito *et al.* (2021) destacam o crescimento dos estabelecimentos de treinos funcionais e musculação no Brasil, que já ocupa a segunda posição mundial em número de locais e a terceira em faturamento. O aumento da oferta facilita a entrada de novos praticantes, mas também ressalta a importância do acompanhamento profissional especializado para o direcionamento e acompanhamento durante a prática. Muitas pessoas se machucam ao realizarem exercícios para os quais não receberam treinamento, ou não possuem o gesto motor para realizá-lo de forma adequada, e isso pode afetar a saúde e a preparação física (Mallmann, 2020).

Segundo Amadio e Serrão (2011), a análise física dos movimentos humanos é a área do estudo da biomecânica, envolvida na identificação de parâmetros mecânicos capazes de permitir que os movimentos sejam realizados de forma adequada e segura e de influenciar o rendimento esportivo e a melhora da qualidade de vida. O acompanhamento de um profissional na execução de exercícios garante que sejam realizados de maneira correta, exemplificando e corrigindo a postura em busca de uma prática segura e eficiente (Tecnofit, 2024). Porém, o acompanhamento por um profissional especializado não é acessível para a maioria dos brasileiros, com custos que variam entre R\$ 80,00 e R\$ 200,00 por hora em grandes capitais, como São Paulo, Rio de Janeiro e Brasília (Neon, 2024).

Cuesta-Vargas, Galán-Mercant, Williams (2010) indicam que os métodos tradicionais para coleta de dados sobre posição e movimento do corpo humano, embora amplamente utilizados, apresentam limitações em termos de mobilidade, custo e precisão dos dados em ambientes não controlados. Nesse cenário, o avanço da visão computacional trouxe uma transformação significativa, permitindo a análise de movimentos com maior acessibilidade e aplicabilidade em ambientes diversos. Ferramentas como a OpenPose, apresentada por Boesch (2023), destacam-se por sua capacidade de detectar até

135 pontos-chave do corpo humano com precisão e em tempo real, utilizando apenas imagens ou vídeos capturados por câmeras convencionais, sem a necessidade de equipamentos especializados de alto custo.

Segundo Boesch (2023), a avaliação via OpenPose expande o potencial de análise de movimentos em contextos variados, incluindo reabilitação, ergonomia, esportes e atividades físicas em geral. Na prática esportiva, por exemplo, a OpenPose oferece possibilidades como a análise detalhada de técnicas de movimento, identificação de falhas na execução de exercícios e prevenção de lesões por meio de *feedback* imediato aos atletas. Além disso, sua aplicabilidade em tempo real torna a ferramenta ideal para treinos monitorados remotamente, permitindo que profissionais supervisionem os movimentos de seus alunos mesmo à distância.

Outro ponto destacado por Boesch (2023) é a escalabilidade da OpenPose, que pode ser empregada tanto em ambientes laboratoriais quanto em cenários cotidianos, como academias, residências ou espaços públicos. Isso democratiza o acesso à avaliação biomecânica, permitindo que indivíduos de diferentes níveis de especialização tenham acesso a uma análise detalhada de seus movimentos, algo anteriormente disponível apenas em laboratórios ou centros esportivos de alto desempenho.

Adicionalmente, a OpenPose possibilita a integração com modelos de aprendizado de máquina, como Redes Convolucionais de Grafos (GCNs), permitindo a criação de sistemas automatizados capazes de classificar ou corrigir movimentos com base em padrões previamente treinados, ampliando ainda mais seu uso em aplicações práticas. Dessa forma, a avaliação via OpenPose não apenas supera limitações dos métodos tradicionais, mas também abre novas possibilidades para a análise de movimentos de forma acessível, precisa e em diversos contextos.

Diante deste contexto, este trabalho apresenta o desenvolvimento de uma aplicação *desktop* de processamento de imagens que analise a execução de exercícios físicos a partir de vídeos capturados por câmeras comuns, fornecendo *feedback* aos usuários com o intuito de aprimorar a eficácia e a segurança dos treinos. Os objetivos específicos são: (i) desenvolver um modelo de aprendizado de máquina híbrido, combinando Redes Convolucionais de Grafos (GCNs) e Redes Long Short-Term Memory (LSTM), para avaliar a correção da postura e do movimento durante exercícios físicos. A partir dos pontos-chave do corpo humano extraídos pela biblioteca OpenPose, o modelo considerará a estrutura espacial das articulações e a dinâmica temporal dos movimentos, analisando vídeos capturados por câmeras convencionais; (ii) identificar os principais fatores que influenciam a precisão do modelo de aprendizado de máquina na avaliação da técnica de execução de exercícios físicos e sua aplicabilidade com usuários reais; (iii) desenvolver uma interface de usuário intuitiva e amigável para a aplicação *desktop* que integre o modelo de aprendizado de máquina, permitindo que os usuários visualizem e ajustem sua técnica de execução de exercícios físicos.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo está subdividido em quatro seções. A seção 2.1 aborda a análise de movimentos e postura durante exercícios físicos, apresentando as origens e métodos utilizados nessa área de estudo. A seção 2.2 apresenta o funcionamento da biblioteca OpenPose, detalhando sua arquitetura baseada em redes convolucionais de múltiplos estágios e o processo de estimativa de poses humanas. A seção 2.3 se aprofunda na definição e exemplos de aplicação de redes convolucionais de grafos. Por fim, a seção 2.4 apresentará os trabalhos correlatos identificados durante a revisão bibliográfica, contextualizando a pesquisa atual em relação ao estado da arte.

### 2.1 ANÁLISE DE MOVIMENTOS E CORREÇÃO DE EXERCÍCIOS FÍSICOS

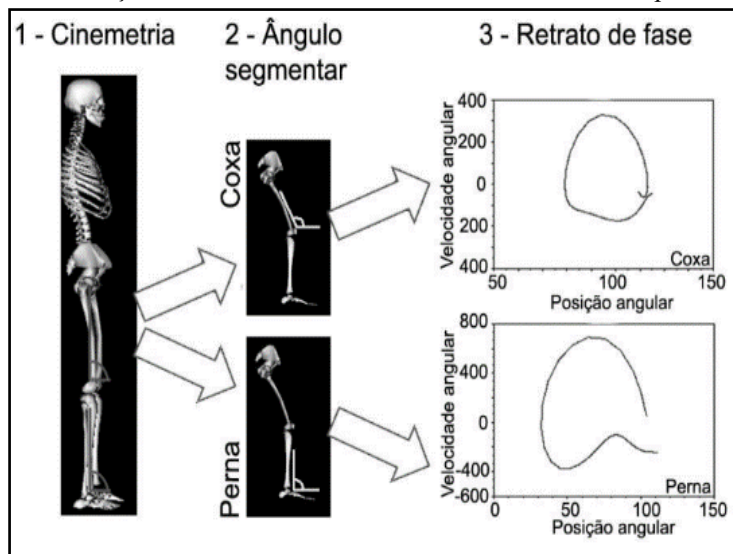
Segundo Caspersen *et al.* (1985), "exercício físico" não é meramente um sinônimo de "atividade física". Embora esses termos sejam frequentemente usados de forma intercambiável e compartilhem elementos em comum, como o movimento corporal e o gasto energético mensurável, o exercício físico distingue-se como uma subcategoria específica. Ele é caracterizado por ser uma atividade física intencionalmente planejada, estruturada e repetitiva, com o objetivo de melhorar ou manter um ou mais componentes da aptidão física (Caspersen *et al.*, 1985).

Juntamente com a prática de exercícios, surge o ramo de estudos focado em sua análise, conhecido como biomecânica. Amadio e Serrão (2011), caracterizam a biomecânica como uma área de estudo multidisciplinar, que utiliza conceitos fundamentais da Física unidos a conhecimentos da Anatomia e da Fisiologia, disciplinas que delimitam as características estruturais e funcionais do aparelho locomotor humano. Segundo Bartlett (2014), o estudo e aplicação dos conceitos da biomecânica traz uma gama de benefícios não apenas para atletas de alto rendimento, mas para qualquer praticante que busque melhorar sua performance e reduzir o risco de acidentes e lesões relacionadas à execução errônea do movimento.

O estudo e a fundamentação teórica do estudo do movimento depende fortemente da coleta e análise de dados da execução dessas atividades. A análise biomecânica do movimento humano é operacionalizada a partir da adoção daquelas que são reconhecidas como as suas quatro grandes áreas de investigação: a cinemetria, a dinamometria, a eletromiografia e a antropometria (Amadio *et al.*, 1999). Os autores ainda apontam que cada uma dessas áreas de investigação coleta e analisa dados sobre um aspecto do movimento: a dinamometria tem como objetivo principal a determinação das forças que produzem o movimento, a eletromiografia estuda os músculos a partir da captação dos eventos elétricos vinculados

à contração muscular, a antropometria descreve, a partir de técnicas experimentais e ou analíticas, as características físicas dos segmentos corporais e a cinemetria objetiva e analisa os dados da posição, do deslocamento, da velocidade e da aceleração dos segmentos e do próprio corpo humano, exemplo que pode ser visto na Figura 1.

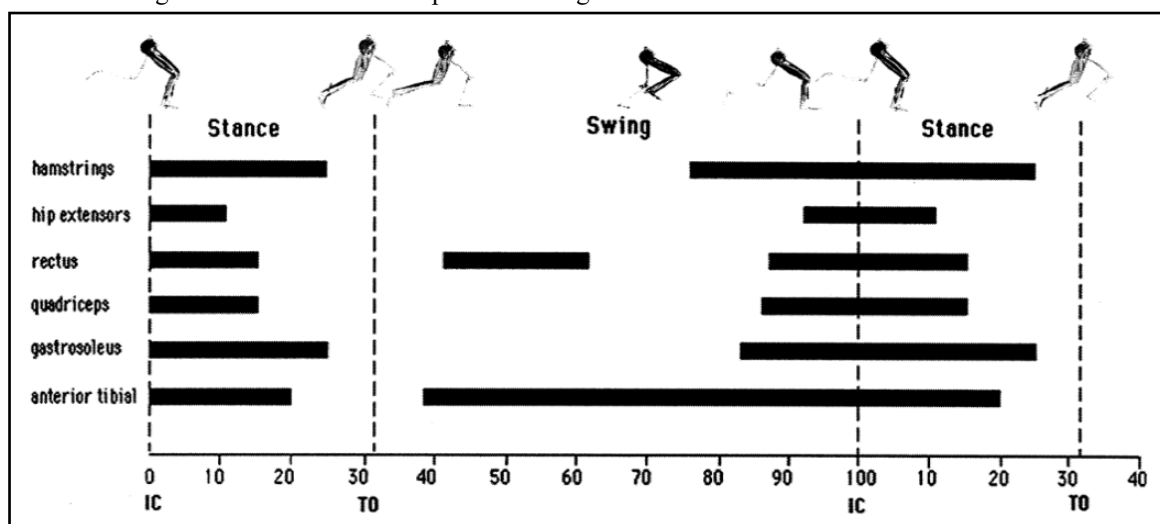
Figura 1 – Exemplo da extração de dados do movimento através da cinemetria para estudo de caminhada.



Fonte: Haupenthal, Pereira e Michaelsen (2010).

Embora exista uma forte relação entre a biomecânica e estudos focados na performance da execução de esportes, essa não é sua única área de estudo. Segundo Amadio e Serrão (2011), a investigação de parâmetros relacionados à locomoção humana é um dos aspectos mais estudados pela biomecânica, com estudos e considerações creditados à pensadores clássicos como Aristóteles e Leonardo da Vinci e expandidos amplamente com a aplicação dos conceitos da física de Isaac Newton pelos irmãos Weber no século XIX. Segundo Novacheck (1998), o crescimento e a popularização de várias modalidades de corrida a partir da década de 1960 resultaram em um aumento significativo nos estudos biomecânicos, voltados tanto para a melhoria de desempenho quanto para a redução de lesões. Esses estudos continuam até os dias atuais, com processos amplamente documentados. A Figura 2 exemplifica alguns dos dados coletados durante a análise de uma corrida, utilizando técnicas de eletromiografia.

Figura 2 – Dados coletados por eletromiografia de um indivíduo durante uma corrida.



Fonte: Novacheck (1998).

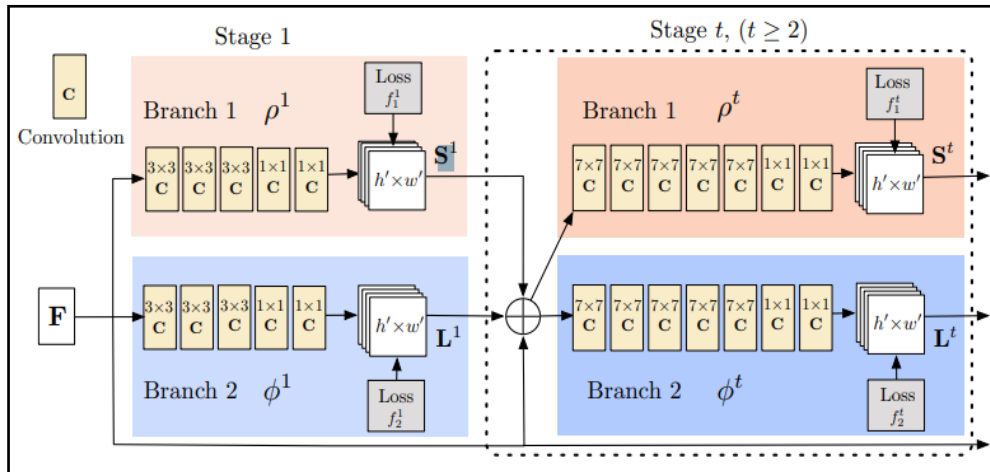
De acordo com Espitia-Mora, Vélez-Guerrero e Callejas-Cuervo (2024), os avanços tecnológicos têm expandido os métodos disponíveis para a captura de dados de movimento do corpo humano. Esses métodos variam desde configurações complexas, como múltiplas câmeras calibradas para criar um ambiente tridimensional, até trajes de captura de movimento, bem como soluções mais simples, como acelerômetros, giroscópios e outros sensores vestidos durante a prática da atividade. No entanto, como apontado por Singh *et al.* (2022), esses métodos apresentam limitações, como altos custos de aquisição, curta duração de bateria e restrições de mobilidade do usuário.

## 2.2 OPENPOSE

A utilização de técnicas de visão computacional e aprendizado de máquina tem promovido avanços significativos na análise automatizada de movimentos humanos. Uma dessas soluções amplamente difundidas é a biblioteca OpenPose, desenvolvida por membros do Carnegie Mellon Perceptual Computing Lab, que permite a estimativa em tempo real das poses de múltiplos indivíduos a partir de vídeos ou imagens. Segundo Cao *et al.* (2018), o OpenPose destaca-se por adotar uma abordagem *bottom-up*, no qual os pontos-chave do corpo, presente no vídeo, são inicialmente detectados e depois associados às pessoas presentes na imagem, diferentemente das abordagens convencionais que primeiro identificam as pessoas no vídeo e realizam múltiplos processos de estimativa de pose individuais.

Cao *et al.* (2018) descrevem que o procedimento se inicia com a extração de características visuais por meio de uma Rede Neural Convolutiva (*Convolutional Neural Network – CNN*) baseada na arquitetura VGG-19, que gera um mapa de *features* utilizado como entrada para uma segunda rede convolutiva de múltiplos estágios, composta por duas ramificações paralelas. A Figura 3 ilustra a arquitetura desta rede de múltiplos estágios. Em cada estágio, a primeira ramificação gera os mapas de confiança das partes do corpo, enquanto a segunda calcula os campos de afinidade entre essas partes. Os resultados de ambos os estágios, juntamente com as *features* extraídas da imagem original, são combinados e alimentam o estágio seguinte. Esse processo iterativo permite refinar progressivamente as estimativas, resultando em maior precisão na detecção da pose.

Figura 3 – Representação da arquitetura de rede do OpenPose.



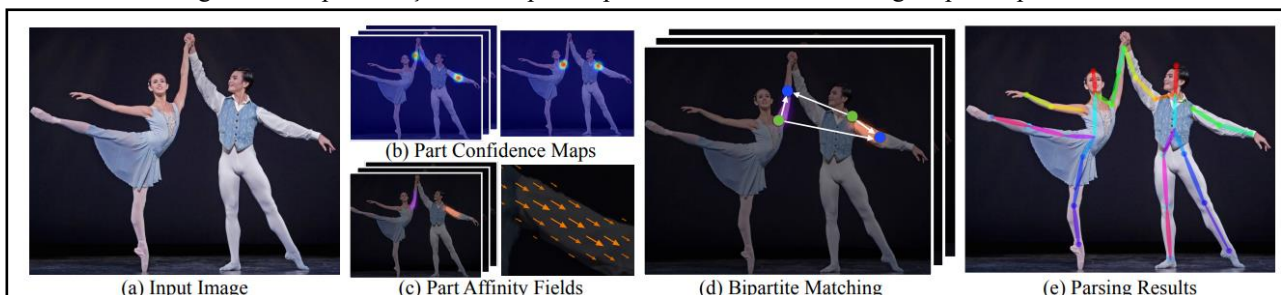
Fonte: Cao *et al.* (2018).

Segundo Cao *et al.* (2018), a primeira ramificação produz os Mapas de Confiança (*Part Confidence Maps*): imagens bidimensionais que indicam a probabilidade de cada parte do corpo (como cotovelo ou joelho) estar localizada em uma posição específica da imagem, com cada parte sendo representada em um canal distinto. A detecção dos pontos correspondentes ao ombro e cotovelo esquerdos de duas pessoas pode ser observada no item (b) da Figura 4.

A segunda ramificação gera os Campos de Afinidade de Partes (*Part Affinity Fields – PAFs*): campos vetoriais que codificam a direção e a associação entre pares de partes do corpo, permitindo a identificação de quais pontos pertencem a um determinado indivíduo. Essa relação pode ser visualizada no item (c) da Figura 4, onde os pontos previamente detectados são agrupados para formar articulações coerentes por pessoa.

Após várias etapas de refinamento, a rede produz as versões finais dos mapas de confiança e dos PAFs. Esses dados são então organizados em uma estrutura de grafo e submetidos a um algoritmo de inferência, que segmenta os pontos em múltiplos grafos e extrai a configuração final das poses de cada indivíduo presente na imagem. Esse processo é ilustrado no item (d) da Figura 4, e o resultado pode ser visto no item (e).

Figura 4 – Representação das etapas do processamento de uma imagem pelo OpenPose.



Fonte: Cao *et al.* (2018).

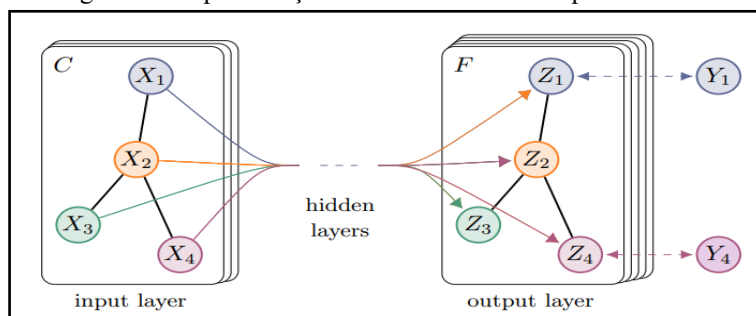
Cao *et al.* (2018) afirmam que a adoção dessa estratégia iterativa em múltiplos estágios, com supervisão intermediária a cada ciclo, permite à rede melhorar progressivamente a qualidade das estimativas., apresentando desempenho e precisão superiores a métodos anteriores, sendo capaz de operar em tempo real, mesmo em contextos com múltiplas pessoas e ambientes não controlados.

### 2.3 REDES CONVOLUCIONAIS DE GRAFOS

Wu *et al.* (2021) caracterizam as Redes Convolucionais de Grafos (*Graph Convolutional Network* - GCN) como uma extensão das redes convolucionais tradicionais, projetadas para operar com dados estruturados como grafos, podendo representar conjuntos de elementos interconectados com relações complexas e interdependências, sem possuir necessariamente uma organização regular como imagens ou sequências lineares. Essa característica torna as GCNs particularmente adequadas para representar e processar relações complexas entre elementos conectados, como redes sociais, estruturas moleculares, ou sequências de poses humanas ao longo do tempo.

Kipf e Welling (2016) apresentam uma definição base para o processo de propagação em uma GCN onde cada nó do grafo atualiza seus pesos e vetor de características através do uso de uma função de ativação sobre a agregação de informações de seus vizinhos conectados, baseando-se na estrutura de adjacência do grafo. Esse modelo de propagação permite que a rede capture relações locais e globais entre os nós do grafo, mesmo em estruturas irregulares. Na Figura 5, pode-se ver a representação de uma GCN com múltiplas camadas para aprendizado semissupervisionado, com  $C$  representando uma camada de entrada e  $F$  uma camada de saída, tendo múltiplas camadas ocultas entre elas. A estrutura do grafo conta com nós representados por  $X_i$  e  $Z_i$ , as arestas ligando-os representadas por linhas pretas e os rótulos de classificação dos nós que os possuem representados por  $Y_i$ .

Figura 5 – Representação de uma GCN de múltiplas camadas.

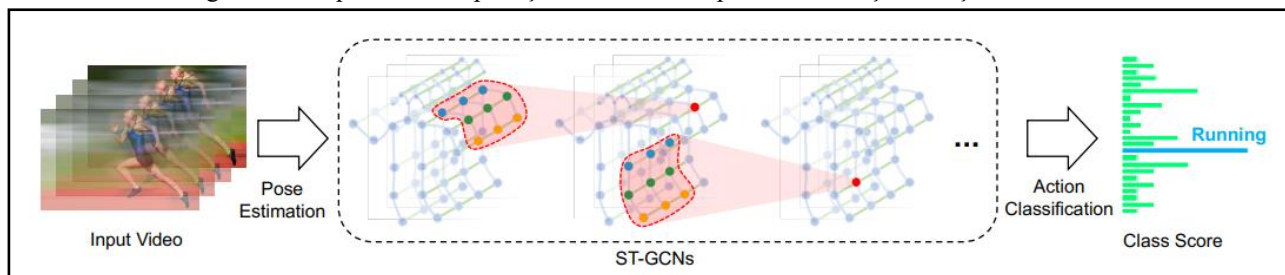


Fonte: Kipf e Welling (2016).

Essa arquitetura de rede tem sido amplamente utilizada em tarefas de classificação de nós, predição de ligações, detecção de comunidades e aplicações mais específicas como em tarefas de visão computacional envolvendo estimativas de pose humana. Um exemplo é o trabalho desenvolvido por Yan, Xiong e Lin (2018), no qual os autores propuseram o uso de uma GCN para reconhecimento de ações humanas a partir da análise de dados posicionais dos pontos-chave do corpo extraídos de vídeos. O modelo é estruturado de modo que cada ponto do corpo seja representado como um nó do grafo, conectado a seus vizinhos anatômicos (como o joelho direito ao pé direito ou o ombro esquerdo ao pulso esquerdo), além de ser ligado ao mesmo ponto nos quadros anteriores e posteriores da sequência temporal. Essa estrutura permite a rede capturar não apenas a configuração espacial dos membros, mas também a evolução temporal do movimento ao longo da execução.

Conforme ilustra a Figura 6, o processo inicia com a divisão do vídeo de entrada em múltiplos quadros. Cada quadro é submetido à estimativa de pose por meio da biblioteca OpenPose, resultando nas coordenadas espaciais de pontos-chave do corpo humano. Em seguida, esses dados são organizados conforme a estrutura apresentada anteriormente e fornecidos como entrada para uma GCN, que processa as informações e classifica a sequência de movimento em uma das classes previamente definidas durante o treinamento da rede, com base na pontuação de afinidade mais elevada entre as categorias disponíveis.

Figura 6 – Arquitetura da aplicação de uma GCN para identificação de ações humanas.



Fonte: Yan, Xiong e Lin (2018).

## 2.4 TRABALHOS CORRELATOS

Nesta seção, apresenta-se uma revisão sistemática da literatura existente sobre análise da corretude de exercícios físicos envolvendo aprendizado de máquina e uma síntese dos trabalhos correlatos. Para realizar uma busca sistemática de trabalhos correlatos com o tema proposto, foram utilizados os portais Google Scholar e Portal CAPES como fontes de pesquisa. Os termos de busca escolhidos foram “exercise”, “image”, “recognition”, “movement”, “training”, “body”, “pose” e “estimation”, combinados de forma estratégica para identificar artigos relevantes a algum dos objetivos principais do projeto. A busca considerou apenas trabalhos publicados a partir de 2020, resultando em aproximadamente 355 artigos no Portal CAPES e 86.000 artigos no Google Scholar.

Os artigos identificados foram analisados em detalhe, considerando a relevância para a proposta de aplicação e o número de citações. Além disso, utilizou-se a plataforma Litmaps para auxiliar na identificação de trabalhos adicionais aos já encontrados. Após a análise, foram selecionados oito trabalhos que abordam a análise de movimentos do corpo humano por meio do processamento de vídeos, conforme apresentado no Quadro 1.

Quadro 1 – Síntese dos trabalhos correlatos selecionados

Assunto	Filtro	Referência
Video based exercise recognition and correct pose detection	“exercise” AND “image” AND “recognition”	Rangari <i>et al.</i> (2022)
3D Pose Based Feedback For Physical Exercises	“exercise” AND “image” AND “recognition”	Zhao <i>et al.</i> (2022)
Automatic recognition and assessment of physical exercises from RGB images	“exercise” AND “image” AND “recognition”	Pham <i>et al.</i> (2022)
Recognition technology of human body movement behavior in fitness exercise based on transfer learning	“movement” AND “exercise” AND “body” AND “image”	Ke <i>et al.</i> (2022)
Fast and robust video-based exercise classification via body pose tracking and scalable multivariate time series classifiers	“movement” AND “exercise” AND “body” AND “image”	Singh <i>et al.</i> (2022)
Accuracy of Hidden Markov Models in Identifying Alterations in Movement Patterns during Biceps-Curl Weight-Lifting Exercise	“movement” AND “exercise” AND “body” AND “image” AND “training”	Peres <i>et al.</i> (2022)
Human Body Pose Estimation and Applications	“body” AND “pose” AND “estimation”	Amrutha, Prabu, Paulose (2021)
Markerless Motion Tracking With Noisy Video and IMU Data	“body” AND “movement” AND “tracking”	Shin, Li, Halilaj (2023)

Fonte: elaborado pelo autor.

Dentre os trabalhos correlatos apresentados foram selecionados três que apresentam características mais semelhantes com os principais objetivos propostos no projeto, tendo suas principais características sintetizadas nos quadros a seguir. O primeiro trabalho (Quadro 2) descreve uma aplicação que identifica exercícios físicos através de vídeos em tempo real e os apresenta ao usuário em uma interface Web (Pham *et al.*, 2022). O segundo trabalho (Quadro 3) é um *software* capaz de classificar exercícios através da estimativa de pontos do corpo humano processados de um vídeo (Rangari *et al.*, 2022). Por fim, o terceiro trabalho (Quadro 4) é um estudo que analisa a viabilidade do uso de vídeos para reconhecimento e caracterização de exercícios físicos em tempo real utilizando o desenvolvimento militar com barra como exemplo (Singh *et al.*, 2022).

Quadro 2 – Automatic recognition and assessment of physical exercises from RGB images

Referência	Pham <i>et al.</i> (2022)
Objetivos	Auxiliar praticantes inexperientes de exercício através do reconhecimento em tempo real de exercícios físicos pela análise de imagens RGB e interface Web para interação com usuário.
Principais funcionalidades	Processamento de vídeos RGB em tempo real, classificação da qualidade de exercícios baseada em ângulos entre pontos-chaves do corpo humano comparados com exemplos de um banco de dados e visualização dos dados em ambiente Web.
Ferramentas de desenvolvimento	Linguagem de programação Python com bibliotecas numPy, TensorFlow e pyside2, a API MediaPipe Pose e recursos computacionais da Google Cloud.
Resultados e conclusões	Resultados experimentais apontam uma acurácia de 98,33% e a utilidade da ferramenta desenvolvida para aprendizados de exercícios, porém, o banco de dados deve ser expandido.

Fonte: elaborado pelo autor.

Pham *et al.* (2022) disponibilizaram um sistema de reconhecimento e avaliação de exercícios físicos baseado em imagens RGB. Utilizando o MediaPipe para a estimativa de pontos-chave do corpo humano através de vídeos, a sequência de dados de posição tridimensional é extraída e uma rede neural no modelo DD-Net é implementada para reconhecer ações em tempo real. O sistema calcula tanto pontuações baseadas em quadros do vídeo quanto em sequências, utilizando a técnica *Dynamic Time Warping* (DTW) para avaliar o desempenho ao longo de múltiplos quadros. Segundo Pham *et al.* (2022), o *framework* é projetado para funcionar em câmeras comuns de smartphones e laptops, e alcança uma precisão média de 98,33% para reconhecimento de ações em testes com três exercícios populares.

Quadro 3 – Video based exercise recognition and correct pose detection

Referência	Rangari <i>et al.</i> (2022).
Objetivos	Estimar a posição tridimensional das articulações do corpo humano através de imagens bidimensionais e sua classificação em um banco de dados de exercícios.
Principais funcionalidades	Classificar um exercício sendo executado através de uma imagem RGB.
Ferramentas de desenvolvimento	Linguagem de programação Python com as bibliotecas OpenPose e TensorFlow.
Resultados e conclusões	O modelo de classificação proposto obteve uma acurácia de 97% em determinar qual tipo de exercício está sendo executado.

Fonte: elaborado pelo autor.

Rangari *et al.* (2022) propuseram um sistema de reconhecimento de exercícios baseado na análise de vídeos, capaz de identificar e avaliar a correteza postural durante prática de exercícios físicos. Utilizando dados obtidos pela estimativa de coordenadas de 18 articulações do corpo humano obtidas por câmeras RGB, os autores propõem um modelo de rede neural que classifica os movimentos e avalia a correteza das posturas em tempo real, sem a necessidade de sensores vestíveis ou outros meios externos de captura de dados. Para isso, organizou-se um conjunto de dados que inclui vídeos de exercícios de prancha com posturas corretas e incorretas, resultando em uma precisão de 97,01%, superior a métodos existentes baseados em sensores, segundo Rangari *et al.* (2022).

Quadro 4 – Fast and robust video-based exercise classification via body pose tracking and scalable multivariate time series classifiers

Referência	Singh <i>et al.</i> (2022).
Objetivos	Classificar a correteza da execução do exercício de desenvolvimento militar com barra através da análise de vídeo e redes neurais para aplicação em fisioterapia e análise de performance esportiva sem o uso de sensores físicos.
Principais funcionalidades	Acompanhar múltiplas partes do corpo através de um vídeo RGB e classificar seus movimentos ao longo do vídeo dentre grupos classificados pelos autores com uma rede neural.
Ferramentas de desenvolvimento	Linguagem de programação Python, com as bibliotecas OpenPose, SlowFast e TensorFlow Lite.
Resultados e conclusões	Foi alcançada uma acurácia média de 87% na classificação do exercício proposto, juntamente com a constatação da viabilidade e robustez do uso de um modelo similar.

Fonte: elaborado pelo autor.

O artigo de Singh *et al.* (2022) apresenta o BodyMTS, um sistema para classificação de exercícios físicos utilizando vídeos e estimativa de pontos-chave do corpo humano, com foco na classificação da correteza do exercício de desenvolvimento militar com barra. A abordagem proposta transforma vídeos em séries temporais multivariadas, rastreando pontos corporais com o OpenPose e classificando-as com uma rede neural de arquitetura SlowFast. Segundo Singh *et al.* (2022), o BodyMTS alcança 87% de acurácia ao classificar a correteza do exercício, com menor tempo de execução e requisitos de infraestrutura que outras propostas mais convencionais.

### 3 DESCRIÇÃO DA APLICAÇÃO

Para melhor entendimento do desenvolvimento da aplicação, este capítulo foi subdividido em quatro seções. A seção 3.1 abordará a extração das informações posicionais dos pontos-chave do corpo humano a partir de vídeos RGB, o processo de captura de dados para o *dataset* e sua representação em um formato *JavaScript Object Notation* (JSON). A seção 3.2 abordará o desenvolvimento de um modelo de Rede Convolucional de Grafos (GCN) temporal capaz de normalizar e processar os dados extraídos de vídeos e caracterizá-los entre um determinado grupo de exercícios informados, bem como os resultados obtidos pelo seu treinamento. Na seção 3.3, é demonstrada a aplicação Windows desenvolvida para permitir a interação dos usuários com o modelo através de uma interface visual intuitiva e amigável. E, por fim, a seção 3.4 aborda os resultados de uma pesquisa conduzida para avaliar a usabilidade da aplicação, com a colaboração de voluntários de diferentes níveis de experiência na área de educação física.

#### 3.1 CAPTURA DE DADOS POSICIONAIS DO CORPO HUMANO

Esta seção descreve o processo desenvolvido para a extração de dados posicionais dos pontos-chave do corpo humano a partir de vídeos RGB, bem como os procedimentos adotados para a coleta e organização dos dados utilizados na construção do conjunto de treinamento da rede neural. Os quadros 5 e 6 apresentam os requisitos funcionais e não funcionais levantados para essa etapa.

Quadro 5 – Requisitos Funcionais

RF01 – Extrair os dados de coordenadas de pontos-chave do corpo humano através de um vídeo RGB de entrada.
RF02 – Gerar um vídeo com a representação visual dos pontos e suas conexões anatômicas sobreposta ao vídeo original.
RF03 – Salvar os dados em uma estrutura JSON, contendo os metadados de taxa de quadros por segundo, a resolução original e uma lista com os dados posicionais de cada quadro.

Fonte: elaborado pelo autor



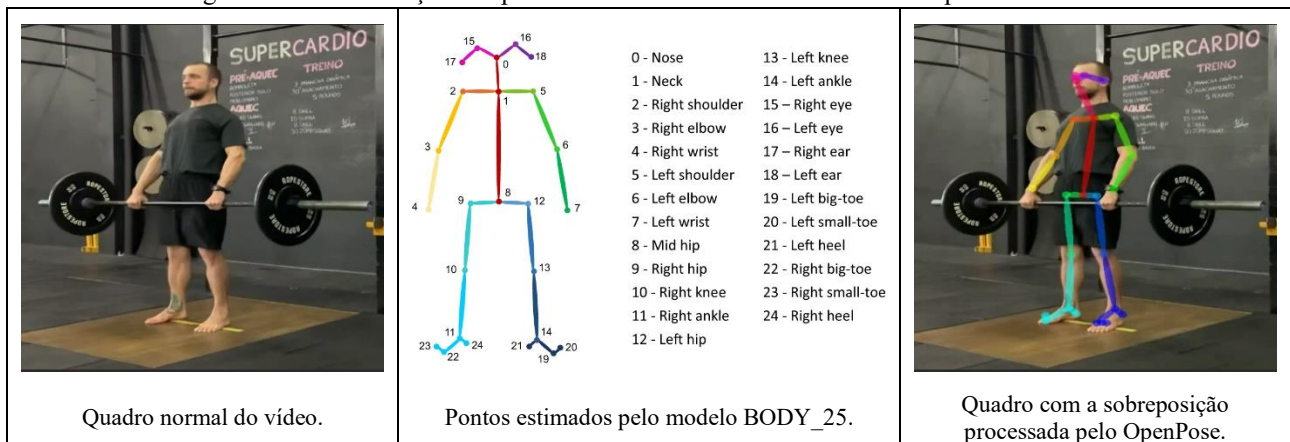
Quadro 6 – Requisitos Não Funcionais

RNF01 – Utilizar o ambiente de desenvolvimento Visual Studio Code.
RNF02 – Utilizar a linguagem Python para desenvolvimento.
RNF03 – Utilizar a API do OpenPose para extração dos dados dos vídeos de entrada.
RNF04 – Utilizar a biblioteca OpenCV para manipulação de quadros e metadados dos vídeos.
RNF05 – Utilizar a biblioteca PyTorch para o desenvolvimento da rede neural.

Fonte: elaborado pelo autor.

A API do OpenPose para Python foi utilizada no processamento inicial dos vídeos e na extração de seus pontos utilizando os recursos de aceleração de GPU em uma placa de vídeo NVIDIA GeForce RTX 3080 Ti. Os parâmetros da API foram configurados para assumir um número máximo de uma pessoa presente no vídeo e utilizar o modelo BODY\_25 (Cao *et al.*, 2018), que extrai as coordenadas posicionais X e Y de 25 pontos do corpo humano como: joelho esquerdo, quadril, ombro direito, cabeça ou mão esquerda. A lista desses 25 pontos, juntamente com uma representação de sua posição no corpo humano, pode ser vista na parte central da Figura 7, conforme apresentado por Zhang *et al.* (2023). Para cada ponto, também é fornecido um valor de confiança, representado por um número real entre 0 e 1, indicando a probabilidade de que suas coordenadas tenham sido estimadas corretamente.

Figura 7 – Demonstração dos pontos estimados e seu resultado em um quadro de vídeo.



Fonte: criado pelo autor com dados extraídos de Zhang *et al.* (2023).

Com essas configurações, o OpenPose processa cada quadro do vídeo individualmente e retornar um objeto contendo os pontos presentes no modelo e uma matriz que representa o quadro com os pontos e suas conexões anatômicas sobrepostas, como pode ser visto também na Figura 7. O Quadro 7 apresenta o código responsável pela configuração desses parâmetros na classe de wrapper do OpenPose, bem como o objeto Datum, responsável por armazenar os resultados do processamento.

Quadro 7 – Código Python configurando parâmetros do wrapper do OpenPose

```
params = {
    "model_folder"      : "openpose_build/models/",
    "model_pose"        : "BODY_25",
    "net_resolution"    : "-1x368",
    "number_people_max": 1,
    "keypoint_scale"    : 3,
    "disable_blending"  : False,
}

self.opWrapper = op WrapperPython()
self.opWrapper.configure(params)
self.opWrapper.start()

# Inicializar o datum object
self.datum = op.Datum()
```

Fonte: elaborado pelo autor.

Para centralizar essas configurações e facilitar o fluxo de trabalho, criou-se uma classe em Python denominada PoseAnalyzer. Essa classe possui membros para as classes da biblioteca, vistos anteriormente no Quadro 7, e encapsula os métodos e parâmetros da API do OpenPose, sendo responsável por sua configuração inicial e pelo processamento dos vídeos. Seu método principal recebe como entrada o caminho do vídeo original e o diretório de destino onde os dados processados devem ser armazenados. Utilizando a biblioteca OpenCV, o vídeo é aberto a partir do caminho informado, seus quadros são lidos de forma sequencial e processados pelos métodos do OpenPose, obtendo a matriz visual de cada quadro e os dados posicionais estimados para cada ponto do corpo. No Quadro 8, pode-se observar o código responsável



por esse processamento, com a matriz representando o quadro do vídeo é enviada para a função e processada pelos membros do OpenPose, retornando a lista de pontos e a matriz com a sobreposição.

Quadro 8 – Código Python para o processamento de um quadro do vídeo.

```
def process_frame(self, frame):  
  
    # Preparar o frame para processamento  
    self.datum.cvInputData = frame  
    self.opWrapper.emplaceAndPop(op.VectorDatum([self.datum]))  
  
    keypoints = self.datum.poseKeypoints  
    if keypoints is None:  
        return None, frame  
  
    # Utilizar o objeto interno do OpenPose para externar os keypoints no frame  
    annotated_frame = self.datum.cvOutputData  
  
    return keypoints[0], annotated_frame
```

Fonte: elaborado pelo autor.

Esses dados são formatados seguindo o formato JSON, incluindo metadados do vídeo e os dados posicionais extraídos de cada quadro. No nível superior do objeto criado, encontram-se três chaves principais: "fps", "frame\_count" e "resolution", que armazenam, respectivamente, a taxa de quadros por segundo do vídeo analisado, o número total de quadros processados e a resolução da imagem no momento da captura, com as dimensões de largura (*width*) e altura (*height*).

A principal chave do objeto é "frames", que contém uma lista ordenada dos quadros processados. Cada item dessa lista é estruturado como um objeto contendo o número do quadro e um dicionário de pontos corporais denominado "keypoints". Esse dicionário associa nomes dos pontos anatômicos gerados pelo modelo BODY\_25 (como "Nose", "Neck", "LShoulder", "RElbow" e "MidHip") a lista de três valores numéricos reais: os dois primeiros correspondem às coordenadas posicionais X e Y do ponto dentro do quadro, e terceiro representa a confiabilidade da estimativa desse ponto.

No contexto geral, três exercícios clássicos do treinamento de força foram selecionados para esse projeto: o levantamento terra com barra, o supino reto com barra e o agachamento com barra. A escolha fundamenta-se na relevância biomecânica desses movimentos, tanto para o desenvolvimento físico quanto para a reabilitação e prevenção de lesões musculares (Vecchio; Hays, 2018). Com o objetivo de avaliar a corretude da execução desses exercícios, foram identificados os principais critérios técnicos, orientações de execução e erros mais comuns de cada exercício com o suporte do coorientador do projeto, servindo de base para as execuções realizadas na captura dos vídeos. A Figura 8 demonstra um quadro da execução de cada um dos exercícios selecionados, conforme exemplos executados por profissionais da área de educação física e utilizados como base para a captura dos demais vídeos do *dataset*.

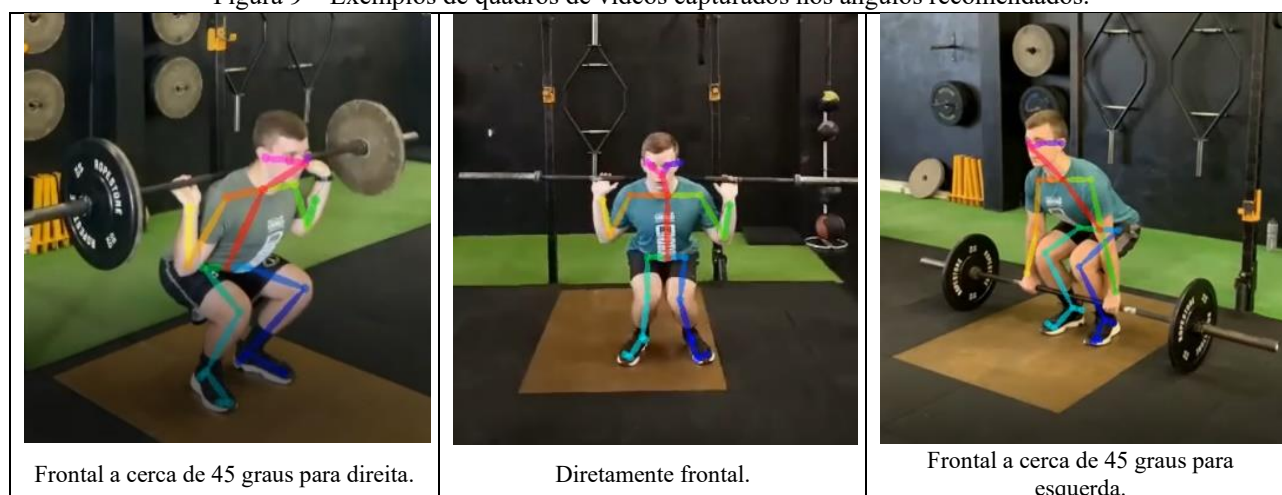
Figura 8 – Imagens de parte da execução dos exercícios selecionados.



Fonte: elaborado pelo autor.

Para minimizar o risco de oclusão de segmentos corporais e garantir maior precisão na detecção dos pontos-chave, os vídeos foram capturados de forma a exibir dois dos planos anatômicos do corpo humano: o plano coronal (através de um ângulo frontal direto) e o plano sagital (ângulos frontais aproximadamente 45 graus à esquerda e à direita do voluntário praticante). Essa abordagem foi adotada com base em testes práticos e nas recomendações de estudos como os de Baldinger, Reimer e Senner (2025), e contribui para ampliar a visibilidade dos membros durante a execução dos exercícios. Exemplos de quadros de diferentes exercícios capturados nesses ângulos podem ser vistos na Figura 9.

Figura 9 – Exemplos de quadros de vídeos capturados nos ângulos recomendados.



Fonte: elaborado pelo autor.

Além de evitar a oclusão de partes do corpo, cada ângulo pode oferecer vantagens dependendo do que se deseja ter como foco da análise. Por exemplo, para avaliar um agachamento com barra e observar especialmente possíveis desvios de joelhos, como o valgismo, uma perspectiva frontal direta é mais adequada, pois permite visualizar claramente o alinhamento e distância entre quadris, joelhos e tornozelos, facilitando também a identificação de possíveis desvios pela rede neural, como pode ser visto na perspectiva diretamente frontal na Figura 9.

Essas informações serviram como base para múltiplas sessões de captura de vídeos de execuções desses exercícios, realizadas com a participação de oito voluntários de diferentes níveis de experiência e formação em práticas esportivas. As gravações ocorreram em uma academia comercial, ambiente previamente autorizado pelos responsáveis do local para a realização do estudo, com o uso de equipamentos apropriados, incluindo uma barra de levantamento de peso olímpico e um banco de apoio. Cada sessão de captura de vídeos foi supervisionada por ao menos um profissional qualificado da área.

A captura dos vídeos foi realizada por meio de diferentes aparelhos celulares, pertencentes aos próprios participantes, o que resultou em vídeos com variadas resoluções e qualidades de imagem. Os ângulos de captura seguiram as recomendações técnicas previamente descritas, visando maximizar a visibilidade dos pontos corporais relevantes. Contudo, não foram utilizadas marcações fixas de posição para os dispositivos de gravação, em vez disso, os vídeos foram capturados simultaneamente por outros voluntários, a partir de múltiplas perspectivas, enquanto um participante executava o exercício. A Figura 10 ilustra esse processo de coleta de dados durante uma das sessões, na qual um voluntário realiza o exercício de supino reto com barra enquanto outros três o filmam a partir de diferentes ângulos, seguindo as recomendações para evitar oclusões corporais.

Figura 10 – Imagem do processo de captura de múltiplos ângulos de um exercício.



Fonte: elaborado pelo autor.

Os vídeos obtidos foram filtrados e classificados manualmente, totalizando 430 vídeos com média de 15 segundos de duração, distribuídos conforme a Tabela 1, onde cada voluntário é identificado por uma letra para fins de anonimização.

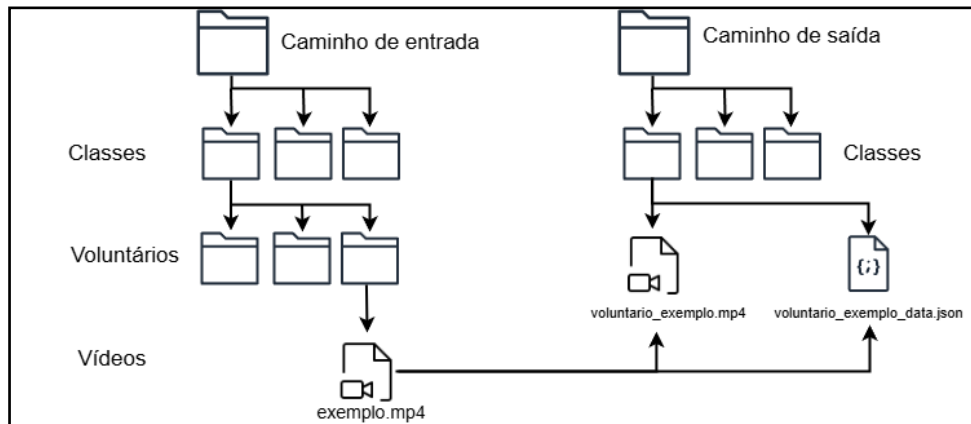
Tabela 1 – Quantidade de vídeos de cada voluntário e exercício.

Exercício Voluntário	Levantamento terra com barra	Levantamento terra com barra com erro	Supino reto com barra	Supino reto com barra com erro	Agachamento com barra	Agachamento com barra com erro
A	6	6	9	11	9	6
B	12	6	10	10	14	4
C	17	5	6	8	11	5
D	4	6	3	6	6	4
E	13	12	17	18	12	12
F	14	12	10	12	8	11
G	9	9	14	8	6	11
H	6	8	2	11	4	7
Total	81	64	71	84	70	60

Fonte: elaborado pelo autor.

Para auxiliar no processamento em larga escala dos vídeos coletados, desenvolveu-se uma classe adicional chamada *VideoProcessor*. Essa classe tem como objetivo automatizar o tratamento de múltiplos arquivos de vídeo, organizando-os conforme o voluntário responsável pela gravação e a respectiva classe de exercício associada. O algoritmo recebe um diretório de entrada e pressupõe que siga uma estrutura hierárquica específica: no primeiro nível encontram-se pastas nomeadas de acordo com as classes de exercício (por exemplo, *Squat*, *Deadlift*), dentro dessas há subpastas que identificam os voluntários, e, finalmente, os arquivos de vídeo localizam-se dentro das pastas de cada voluntário. O algoritmo gera uma estrutura de pasta de saída que reproduz a mesma hierarquia, porém omite o nível que contém os nomes dos voluntários, substituindo-os por prefixos nos nomes dos arquivos gerados. Isso permite preservar a identificação dos participantes e facilita carregar esses dados posteriormente no *dataset*. Essa organização é ilustrada na Figura 11, e posteriormente é utilizada como base para o carregamento dos dados no *dataset* do PyTorch.

Figura 11 – Representação da estrutura de pastas para processamento dos vídeos.



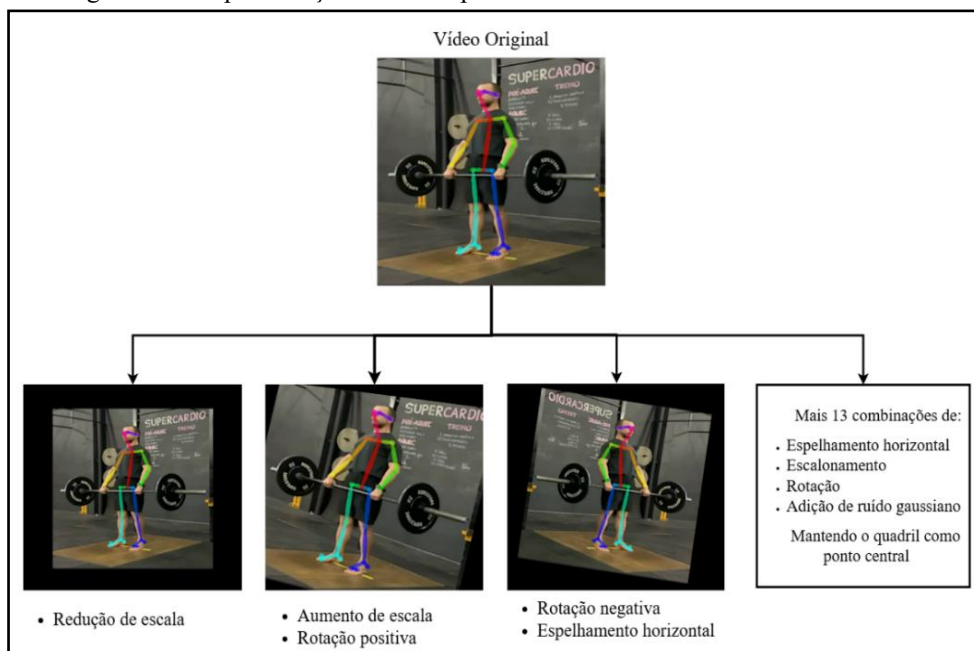
Fonte: elaborado pelo autor.

Para aumentar a variabilidade do conjunto de dados e aprimorar a capacidade de generalização do modelo, foram aplicadas técnicas de *data augmentation* (Shorten e Khoshgoftaar, 2019) diretamente sobre as coordenadas dos pontos corporais extraídos dos vídeos, sem a necessidade de reprocessamento das imagens originais. As transformações implementadas foram: espelhamento horizontal, escalonamento, rotação e adição de ruído gaussiano. O espelhamento horizontal inverte as coordenadas X dos pontos em relação ao eixo vertical da imagem e realiza a substituição automática dos rótulos anatômicos do lado direito para o esquerdo e vice-versa (por exemplo, substituindo o ombro direito pelo ombro esquerdo), de modo a manter a coerência da estrutura corporal refletida.

A transformação de escalonamento aplica uma ampliação ou redução uniforme da pose em torno do MidHip (ponto estimado equivalente ao centro do quadril), simulando variações na distância do praticante em relação à câmera. Já a rotação realiza uma transformação bidimensional em torno do mesmo ponto de referência, aplicando ângulos positivos ou negativos, o que representa pequenas mudanças no enquadramento ou na inclinação da gravação. Por fim, a adição de ruído gaussiano insere pequenas variações aleatórias nas coordenadas X e Y de cada ponto do corpo, simulando imprecisões na detecção ou leves oscilações naturais do movimento durante a execução do exercício. Ao aplicar diferentes combinações dessas transformações, foram obtidos 16 novos arquivos a partir de cada JSON original, totalizando então cerca de 7.310 arquivos. Embora esse processo seja realizado apenas utilizando operações matemáticas sobre a posição

dos pontos já processados, a Figura 12 demonstra uma representação visual do efeito de algumas combinações geradas a partir de um vídeo original.

Figura 12 – Representação visual do processo de aumento de dados de um vídeo.



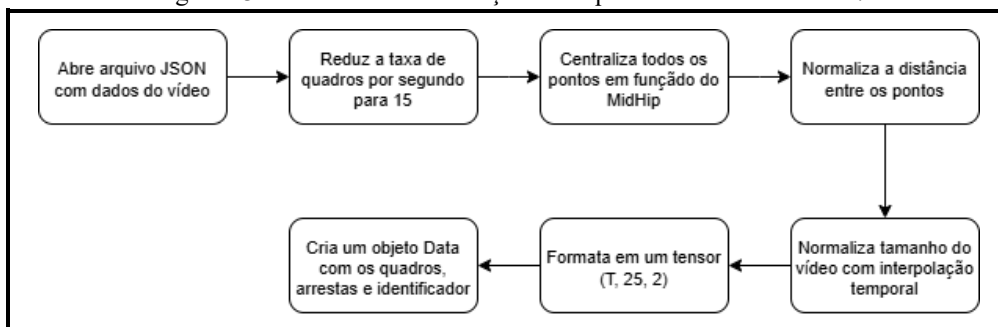
Fonte: elaborado pelo autor.

### 3.2 REDE NEURAL E NORMALIZAÇÃO DE DADOS

Nesta seção será abordada a estrutura de um modelo de Rede Convolucional de Grafos (GCN) temporal e uma classe de *dataset* capaz de normalizar e processar os dados extraídos de vídeos e caracterizá-los entre os tipos de exercícios informados anteriormente, bem como os resultados obtidos pelo seu treinamento. Para que os dados processados anteriormente pudessem ser utilizados, eles precisam ser convertidos para um *dataset* da biblioteca `PyTorch`. Esse processo de preparação é conduzido por uma classe personalizada denominada `ExerciseDataset`, que monta seus dados a partir do caminho de uma pasta base que assume seguir a mesma estrutura de pastas gerada pelo algoritmo de processamento de vídeos e normaliza os arquivos JSON, utilizando as funções demonstradas em maior detalhe no decorrer dessa seção.

Inicialmente, os dados brutos são processados por meio de uma função que executa as seguintes etapas: redução da taxa de quadros por segundo (*Frames Per Second* - FPS), centralização dos pontos em torno do centro do quadril (MidHip), normalização da escala com base na maior distância entre pontos de cada frame e interpolação temporal para garantir que todas as amostras tenham o mesmo comprimento temporal (em número de quadros). Essa uniformidade é fundamental para permitir a comparação entre vídeos com durações diferentes e executados por pessoas com diferentes estaturas e proporções anatômicas. O fluxo de processamento de um arquivo é representado no fluxograma presente na Figura 13.

Figura 13 – Fluxo da normalização do arquivo JSON de um vídeo.



Fonte: elaborado pelo autor.

A taxa de quadros é reduzida para 15 quadros por segundo, por ser um número de quadros que permite um balanço razoável entre o tamanho dos dados e a quantidade de detalhes dos movimentos representados. Essa operação é realizada através do uso da função de geração de vetores distribuídos linearmente da biblioteca `Numpy`, demonstrada no trecho de código do Quadro 9.

Quadro 9 – Código Python para ajuste inicial de taxa de quadros do vídeo.

```
# Método para reduzir a taxa de frames
def downsample_frames(frames, original_fps, target_fps):
    if original_fps <= target_fps:
        return frames

    # Calcula o número de frames após downsampling
    new_length = int(round(len(frames) * target_fps / original_fps))

    # Gera os índices dos frames que serão mantidos
    indices = np.linspace(0, len(frames) - 1, num = new_length, dtype = int)

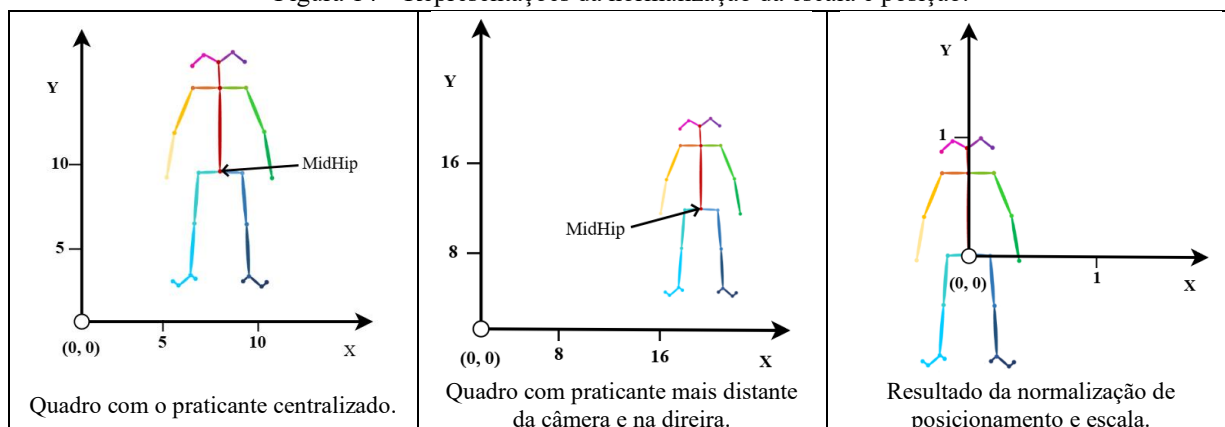
    return [frames[i] for i in indices]
```

Fonte: elaborado pelo autor.

A centralização pelo MidHip visa evitar problemas envolvendo a posição do usuário no vídeo, com o MidHip em si sendo movido para o ponto origem do vídeo (0, 0). A normalização pela maior distância entre pontos de cada quadro padroniza a escala corporal, promovendo maior robustez do modelo a diferentes estaturas do usuário ou distâncias dele da câmera.

A Figura 14 apresenta uma representação visual do processo de normalização aplicado aos pontos corporais. São mostradas três versões dos valores da mesma pose: na primeira, o praticante aparece centralizado e ocupa grande parte do quadro, na segunda está mais distante da câmera e deslocado à direita, enquanto a terceira imagem ilustra como ambas as situações são transformadas após o processo de centralização e normalização. Os pontos são reposicionados de forma que o ponto MidHip passe a ocupar a origem do sistema de coordenadas (0, 0), e as coordenadas X e Y são escaladas proporcionalmente para que a maior distância entre o MidHip e qualquer outro ponto do corpo seja igual a 1. Esse procedimento garante que diferentes posicionamentos e proporções corporais não influenciem negativamente a análise do modelo. Esse processo é executado pelo método `normalize_keypoints`, presente no Quadro 10.

Figura 14 – Representações da normalização da escala e posição.



Fonte: elaborado pelo autor.

Quadro 10 – Código Python responsável pela normalização da distância dos pontos e centralização.

```
def normalize_keypoints(sequence): # sequence: array de shape (frames, 25, 2)

    # Utiliza o midhip de cada frame como ponto de referência
    midhip = sequence[:, B25_R["MidHip"], :]

    # Centraliza todos os pontos em relação ao MidHip
    sequence_centered = sequence - midhip[:, np.newaxis, :]

    # Calcula a maior distância em cada frame para normalizar a escala
    max_dist = np.max(np.linalg.norm(sequence_centered, axis = 2), axis = 1, keepdims = True)
    normalized_sequence = sequence_centered / (max_dist[:, np.newaxis, :] + 1e-8)

    return normalized_sequence
```

Fonte: elaborado pelo autor.

A interpolação linear é então aplicada sobre a lista de quadros de cada vídeo, utilizando recursos da biblioteca NumPy, conforme visto no Quadro 11. Isso faz com que todos os vídeos tenham a mesma duração, definida nesse caso como 15 segundos. Diferentemente do método de redução de quadros, essa interpolação busca adaptar a duração da



seqüência para que todos os exemplos tenham o mesmo número de quadros, gerando quadros artificiais entre os existentes caso necessário.

Quadro 11 – Código Python responsável pela interpolação linear.

```
def temporal_interpolation(sequence, target_len):  
  
    original_len = len(sequence)  
  
    x_original = np.linspace(0, 1, original_len)  
    x_target = np.linspace(0, 1, target_len)  
  
    # Interpolação linear para cada ponto do corpo  
    interpolated_sequence = np.zeros((target_len, sequence.shape[1],  
sequence.shape[2]))  
  
    for joint in range(sequence.shape[1]): # Para cada ponto do corpo  
        for coord in range(sequence.shape[2]): # Para x e y  
            interpolated_sequence[:, joint, coord] = np.interp(x_target, x_original,  
sequence[:, joint, coord])  
  
    return interpolated_sequence
```

Fonte: elaborado pelo autor.

O resultado desse processo é formatado em um *tensor* de dimensões (T, 25, 2), sendo T o número de quadros após a interpolação, 25 correspondendo aos pontos estimados pelo modelo BODY\_25 e 2 representando as coordenadas X e Y de cada ponto. O *tensor* gerado é transformado em um grafo que representa as conexões das poses dos quadros do vídeo, utilizando os métodos providos pela biblioteca PyTorch Geometric. Esse grafo é estruturado de forma que cada quadro do vídeo é representado como um nó, contendo 50 características (as coordenadas X e Y de cada um dos 25 pontos). A estrutura temporal é refletida pelas conexões bidirecionais entre nós consecutivos, com cada quadro sendo conectado com seu predecessor e sucessor.

O código responsável pelo processo de converter essa seqüência de dados normalizada anteriormente para o objeto de *Data* descrito é demonstrado no Quadro 12. Cada objeto representa então o grafo de um vídeo, tendo as características dos T quadros armazenada em X, o identificador numérico da classe em Y e a matriz de adjacência que faz a ligação temporal dos quadros.

Quadro 12 – Código Python para conversão da seqüência de pontos normalizada em um objeto de *Data*.

```
def create_data_object(sequence, label = None):  
  
    target_len = sequence.shape[0]  
  
    # Achata cada frame (cada frame terá 50 features)  
    frame_features = sequence.view(target_len, -1) # shape: (target_len, 50)  
  
    # Cria edge_index conectando o frame i ao i+1 (bidirecional)  
    if target_len > 1:  
        indices = [[i, i + 1] for i in range(target_len - 1)]  
        edge_index = torch.tensor(indices, dtype = torch.long).t().contiguous()  
  
        # Arestas reversas  
        rev_edge_index = edge_index[[1, 0], :]  
        edge_index = torch.cat([edge_index, rev_edge_index], dim = 1)  
  
    else:  
        edge_index = torch.empty((2, 0), dtype = torch.long)  
  
    # Converte a label (string) para inteiro  
    if (label is not None) and (label in LABEL_MAP):  
        y = torch.tensor(LABEL_MAP[label], dtype = torch.long)  
    else:  
        y = torch.tensor(-1, dtype = torch.long)  
  
    # Cria e retorna o objeto Data  
    return Data(x = frame_features, edge_index = edge_index, y = y)
```

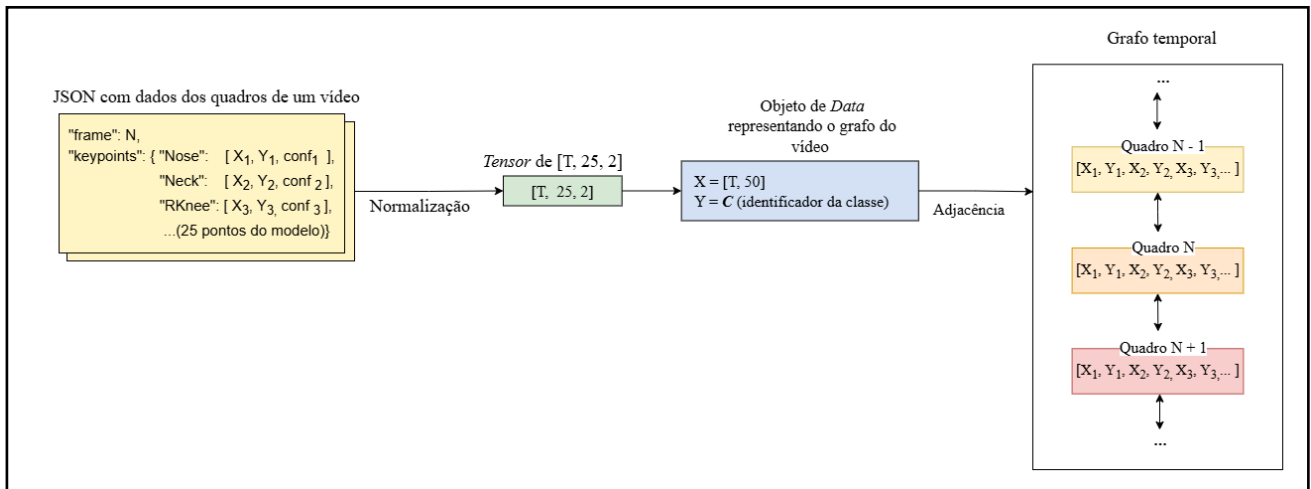
Fonte: elaborado pelo autor.

A Figura 15 representa de forma visual o fluxo de processamento dos dados de um vídeo para a criação do grafo temporal, iniciando pela lista de pontos presente no JSON, passando pelo processo de normalização que resulta em um



Tensor de dimensões (T, 25, 2), que é convertido em um objeto de Data pelo método `create_data_object` e tem sua estrutura de grafo temporal representada com as características e ligações do quadro N, seu predecessor e sucessor.

Figura 15 – Fluxo de processamento dos dados de um vídeo até a estrutura de grafo temporal.

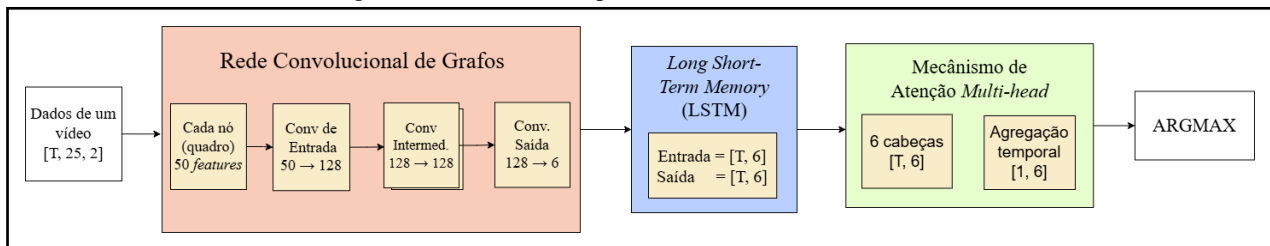


Fonte: elaborado pelo autor.

Para auxiliar no treinamento, desenvolveu-se um algoritmo para criar pastas separando dados entre conjuntos de treinamento e validação, utilizando-se 80% dos dados para treinamento e 20% para validação, além de métodos para salvar os dados já processados em um desses objetos de *dataset* em *cache*, utilizando recursos da biblioteca `Pickle`, para realizar esse processamento e normalização de arquivos JSON em cada iteração de treino da rede.

O modelo proposto para a rede neural se trata de uma Rede Convolutacional de Grafos (GCN) temporal, implementada com os recursos da biblioteca `PyTorch` e `PyTorch Geometric`, seguida por uma rede LSTM e um mecanismo de atenção, conforme apresenta a Figura 16.

Figura 16 – Estrutura de processamento das redes neurais.



Fonte: elaborado pelo autor.

Inicialmente, os dados são processados por uma sequência de camadas convolucionais de grafos, que operam sobre os dados formados com os quadros dos vídeos, conforme abordado na classe do *dataset*. A rede GCN adota três tipos de camadas convolucionais de grafos: a primeira para projeção de entrada, as intermediárias para extração de padrões espaciais e temporais e a última reduz essa representação a um vetor compacto por quadro. Após um processo de testes e ajustes finos, a GCN é composta por três camadas de convolução de grafos: a primeira projeta os vetores de entrada de 50 para um espaço latente de 128 dimensões, a segunda, e única intermediária, mantém essa dimensionalidade de 128 unidades para capturar relações mais complexas entre os nós do grafo, e a terceira reduz essa representação para 6 dimensões por quadro, que serve como entrada para o módulo LSTM subsequente.

Apesar de corresponder ao número de classes de exercício, essa dimensão de saída não representa uma predição, mas sim uma representação que será refinada posteriormente pela LSTM, pelo mecanismo de atenção e pela agregação antes da etapa final de classificação em um tipo de exercício. Esse fluxo é representado em laranja na Figura 16. Cada uma das camadas de convolução da GCN é seguida por uma função de ativação ReLU (*Rectified Linear Unit*), a qual introduz não-linearidades à rede, permitindo que ela aprenda representações mais complexas a partir dos dados de entrada.

Após a aplicação das camadas convolucionais de grafos, os vetores de características resultantes por si só não são suficientes para capturar adequadamente a dinâmica temporal dos movimentos corporais ao longo da sequência de um vídeo, ponto crucial para avaliar sua corretude. Para esse fim, o modelo incorpora um módulo *Long Short-Term Memory* (LSTM), um tipo de Rede Neural Recorrente (*Recurrent Neural Network* - RNN) projetada para lidar com sequências temporais e dependências de longo prazo (Hochreiter; Schmidhuber, 1997), representada em azul na Figura 16. A LSTM é responsável por modelar como os padrões espaciais evoluem ao longo do tempo, aprendendo características

dinâmicas específicas de cada tipo de exercício físico (como o movimento descendente e ascendente de um agachamento ou o padrão do levantamento terra).

Em seguida, os dados de saída da LSTM alimentam um mecanismo de atenção *Multi-Head* (Vaswani *et al.*, 2017) com seis cabeças, permitindo que o modelo aprenda quais partes da sequência temporal são mais relevantes para a classificação, podendo lidar com sequências ruidosas ou com quadros pouco informativos, concentrando a atenção nos momentos mais relevantes da execução do exercício, como o pico do movimento. Por fim, é realizada uma agregação média ao longo da dimensão temporal resultante desse processo de atenção para gerar um vetor com seis posições para cada vídeo, contendo sua proximidade com cada uma das classes. Um algoritmo de extração de argumento máximo (*argmax*) é aplicado nesse vetor, identificando então a classe mais próxima como 1 e definindo-a como a predição do modelo.

Para o processo de treinamento, foi implementada a classe `TemporalGCNClassifier`, que encapsula o modelo, otimizador, função de perda e demais estratégias de regularização e ajustes durante o treinamento. O otimizador utilizado é o Adam, com taxa de aprendizado inicial de 0,001, ajustada posteriormente pela programação de taxa de aprendizado. A função de perda adotada é a de entropia cruzada (*Cross-Entropy Loss*), ponderada por pesos inversamente proporcionais à frequência das classes no conjunto de treinamento, buscando evitar problemas constatados experimentalmente e causados inicialmente pela quantidade de vídeos em cada categoria de exercício, que fazia com que a variação com mais exemplos quase sempre fosse a única identificada pelo modelo. O treinamento foi realizado em lotes (*batch training*), com tamanho fixo de oito amostras por iteração, embaralhando os dados a cada época para garantir maior variabilidade nas combinações apresentadas ao modelo.

Foram adotadas diversas estratégias voltadas à melhoria do desempenho e da capacidade de generalização do modelo, por meio de ajustes finos nos parâmetros ao longo do treinamento. Primeiramente, foi aplicada a técnica de parada antecipada (*early stopping*), que interrompe o treinamento automaticamente caso a perda de validação (*val loss*) não apresente melhora após seis épocas consecutivas. Aplicou-se também uma programação da taxa de aprendizado (*learning rate scheduler*), que reduz a taxa de aprendizado pela metade sempre que a perda de validação não apresenta melhora por quatro épocas consecutivas. No Quadro 13, pode-se ver o trecho de código que realiza a chamada da progressão do *scheduler* através do membro da classe, e monitora a variação de valores de perda, aplicando a parada antecipada caso necessário.

Quadro 13 – Código Python que aplica os ajustes de taxa de aprendizado e controle de parada.

```
# Reduz a taxa de aprendizado se a loss de validação não melhorar
scheduler.step(avg_val_loss)

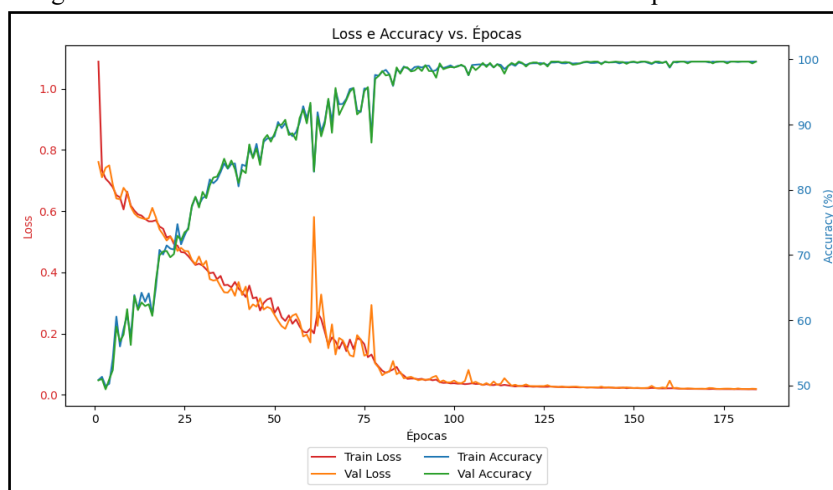
# Early stopping: verifica se a loss de validação melhorou
if avg_val_loss < best_val_loss:
    best_val_loss = avg_val_loss
    early_stop_counter = 0
else:
    early_stop_counter += 1

if early_stop_counter >= self.early_stop_patience:
    print(f"Early stopping ativado: Val Loss não melhorou por {self.early_stop_patience} épocas consecutivas.")
    break
```

Fonte: elaborado pelo autor.

A combinação dessas duas técnicas foi importante para evitar o *overfitting* e permitir que o modelo continuasse sendo aprimorado pelo treinamento, tendo um limite máximo de épocas estipulado em 200, com o melhor modelo percorrendo 184 épocas antes de ser parado. Ao longo do processo de treinamento, os dados de perda e acurácia de treinamento e validação de cada época foram calculados e salvos, permitindo seu acompanhamento em tempo real e sua representação gráfica gerada através da biblioteca `Matplotlib`. O gráfico do treinamento do modelo mais bem sucedido pode ser visto na Figura 17, alcançando uma acurácia de validação de 99,66% e perda de 0,0195 após 184 épocas. Ao final do treinamento, o dicionário de estado do modelo, contendo seus parâmetros e pesos é salvo para poder ser carregado posteriormente.

Figura 17 – Gráfico das métricas de treinamento de cada época do modelo.

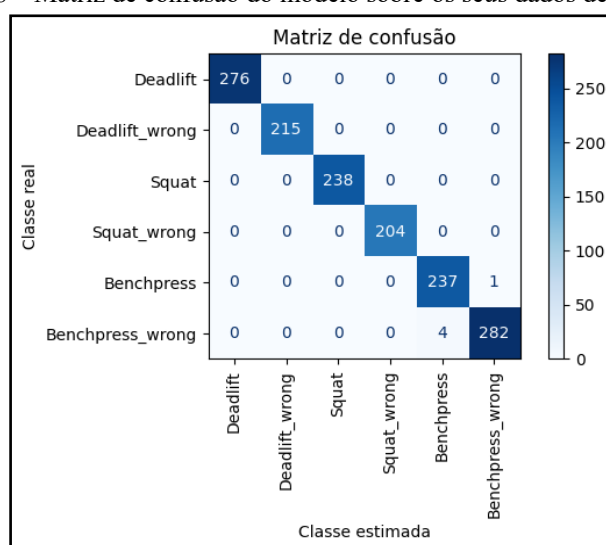


Fonte: elaborado pelo autor.

Ao analisar o gráfico da Figura 17, observa-se que as métricas de perda iniciam em valores elevados e diminuem progressivamente ao longo das épocas, enquanto a acurácia apresenta comportamento inverso, aumentando gradualmente. A velocidade dessa evolução é influenciada pelos mecanismos de ajuste da taxa de aprendizado. Inicialmente, o progresso é mais acentuado, mas, ao identificar-se uma estagnação ou aumento na perda de validação, a taxa de aprendizado é automaticamente reduzida pela metade. Esse ajuste se torna evidente por volta da época 75, quando há um aumento abrupto na perda e uma queda correspondente na acurácia, ocasionando a ativação do ajuste que torna o processo de aprendizado mais gradual. Essa estratégia evita problemas como o *overfitting* e ocorre novamente em outras fases do treinamento, resultando em uma curva de progresso visivelmente mais estável e granular nas épocas finais.

Além disso, é gerada uma lista dos itens que foram classificados erroneamente e a matriz de confusão utilizando a biblioteca *scikit-learn*, a partir dos pesos obtidos pelo treinamento e a base de validação usada, o que possibilitou a análise qualitativa dos erros de classificação e a verificação do desempenho por classe. Pode-se observar na Figura 18 a matriz de confusão gerada pelo modelo. Ao longo do processo de treinamento, essa visualização permitiu analisar que a maioria dos modelos apresentavam erros entre os exercícios, por exemplo, classificando um agachamento com um supino, até que o modelo apresentado nas Figuras 17 e 18 gerou erros apenas entre as variações com e sem erros do supino reto com barra. Na Figura 17, as classes do modelo foram representadas por termos em inglês, sendo "Deadlift" e "Deadlift\_wrong" para execuções corretas e incorretas do levantamento terra com barra, "Squat" e "Squat\_wrong" para o agachamento com barra, e "Benchpress" e "Benchpress\_wrong" para o supino reto com barra.

Figura 18 – Matriz de confusão do modelo sobre os seus dados de validação.

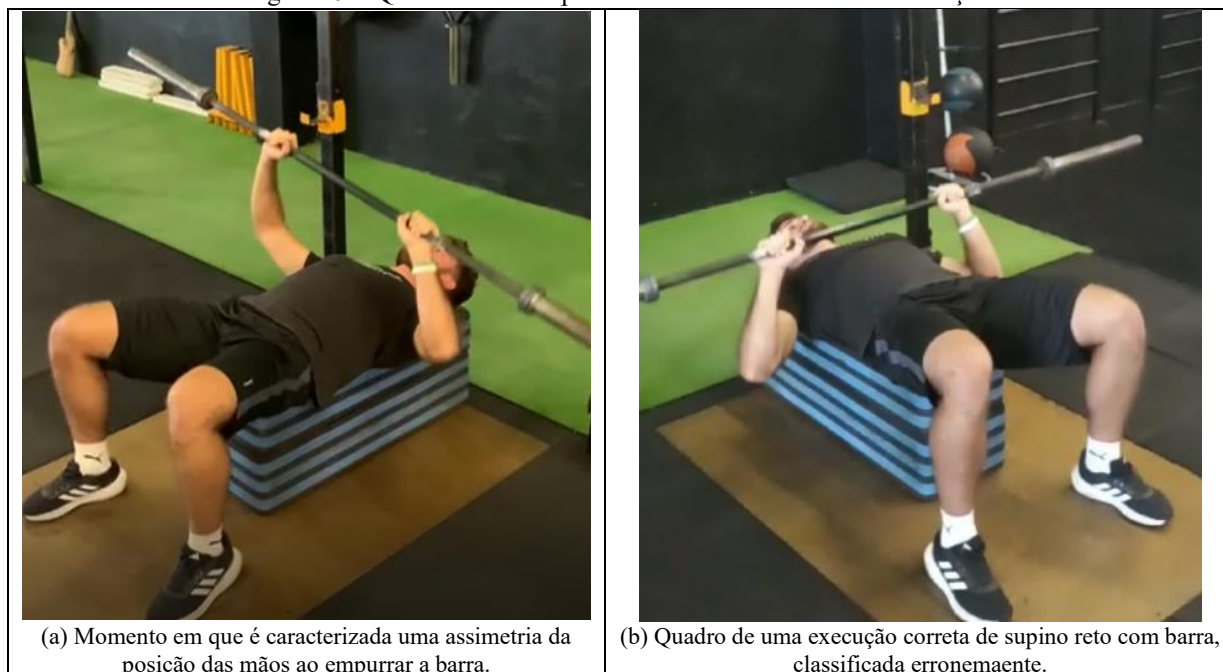


Fonte: elaborado pelo autor.

Ao analisar os vídeos de supino reto com barra que foram classificados incorretamente, observa-se que todos os quatro casos de vídeos com erro classificados como corretos apresentam o erro de assimetria no movimento, com um dos lados da barra sendo elevado antes do outro. Em dois desses vídeos, essa assimetria estava presente apenas em algumas

repetições, enquanto nos outros dois vídeos o desvio era sutil, dificultando sua detecção, especialmente considerando o ângulo em que as gravações foram realizadas. A Figura 19 item (a) exemplifica uma dessas repetições com erro, presente em um vídeo em que, apesar dessa falha, as demais repetições foram executadas corretamente. Já no único caso em que um vídeo de execução correta foi classificado como incorreto, ele não apresenta nenhum motivo claro que justifique a confusão por parte do modelo, porém, possivelmente seu ângulo de captura pode ter feito com que a diferença da posição dos ombros e mãos tenha se assimilado a uma execução com assimetria, conforme visto na Figura 19 item (b).

Figura 19 – Quadros de exemplos de vídeos com erro de classificação.



Fonte: elaborado pelo autor.

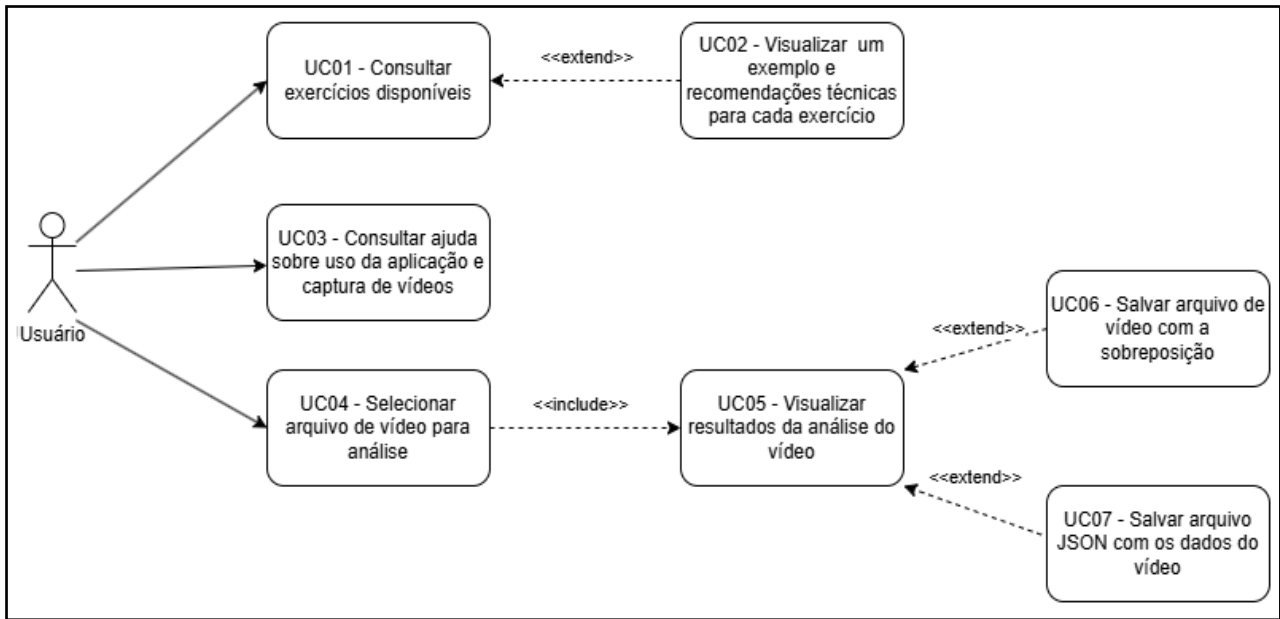
Por fim, vale destacar que, embora ambos os exemplos da Figura 19 tenham sido executados pelo mesmo voluntário, dois dos vídeos com erro classificados como corretos foram realizados por pessoas diferentes, bem como os demais vídeos realizados por esse voluntário foram classificados corretamente.

### 3.3 INTERFACE DE USUÁRIO

Nessa seção é apresentado o processo de desenvolvimento de uma aplicação Windows para permitir a interação dos usuários com o modelo através de uma interface visual intuitiva e amigável. A aplicação foi elaborada conforme as funcionalidades presentes no diagrama de casos de uso apresentando na Figura 20. O ator *Usuário* pode consultar a lista de exercícios propostos e para os quais a rede neural foi treinada (UC01) e, caso deseje, visualizar dados de exemplo de execução e recomendações técnicas para cada um desses exercícios (UC02). Para garantir os melhores resultados, o usuário pode consultar uma seção de ajuda sobre a utilização da aplicação, contendo instruções sobre os melhores ângulos e afins, conforme apresentado anteriormente nesse artigo (UC03).

Por fim, a aplicação permite ao usuário selecionar um arquivo de vídeo local no formato .mp4, que será processado pela rede neural (UC04). Após o processamento, a aplicação apresenta uma tela de resultados contendo: (i) o vídeo processado com sobreposição dos pontos corporais extraídos; (ii) a estimativa do tipo de exercício e sua respectiva classificação quanto à corretude; e (iii) as recomendações técnicas associadas ao tipo de movimento identificado (UC05). A interface também oferece a opção de exportar os arquivos resultantes, permitindo ao usuário salvar tanto o vídeo com as sobreposições (UC06) quanto o arquivo no formato JSON, contendo os dados temporais dos pontos-chave extraídos do vídeo original (UC07).

Figura 20 – Diagrama de casos de uso da aplicação.



Fonte: elaborado pelo autor.

Com base no diagrama apresentado na Figura 20, foram elaborados os requisitos funcionais e não funcionais da interface, conforme descrito nos Quadros 14 e 15. Os requisitos funcionais abrangem as principais funcionalidades esperadas pelo sistema, incluindo listagem de exercícios (RF01), visualização de orientações e dicas (RF02), seção de ajuda (RF03), seleção e processamento de vídeos (RF04), apresentação de resultados (RF05) e funcionalidades de exportação (RF06 e RF07).

Quadro 14 – Requisitos Funcionais.

RF01 – Listar os exercícios para os quais a rede neural foi treinada.
RF02 – Apresentar detalhes da execução e as recomendações técnicas associadas ao tipo de movimento identificado.
RF03 – Apresentar uma página com ajuda sobre a funcionalidade da aplicação e os métodos de captura de vídeos.
RF04 – Permitir ao usuário selecionar um arquivo de vídeo para ser processado pela rede neural.
RF05 – Apresentar uma tela com o resultado da classificação do vídeo, uma visualização do vídeo processado e as dicas e possíveis erros para esse determinado tipo de exercício identificado.
RF06 – Permitir ao usuário salvar o vídeo com a sobreposição dos pontos e conexões anatômicas gerado pela aplicação.
RF07 – Permitir ao usuário salvar o arquivo JSON gerado com os dados do vídeo e dos pontos em cada quadro.

Fonte: elaborado pelo autor

Quadro 15 – Requisitos Não Funcionais.

RNF01 – Ser desenvolvido para a plataforma Windows.
RNF02 – Utilizar a linguagem Python para o desenvolvimento.
RNF03 – Utilizar o ambiente de desenvolvimento Visual Studio Code.
RNF04 – Utilizar as classes para processamento e classificação de vídeo desenvolvidas anteriormente.
RNF05 – Utilizar as bibliotecas CustomTkinter e tkinter para implementar os elementos da interface.

Fonte: elaborado pelo autor.

Baseando-se nesses requisitos, a interface da aplicação “Treine Certo” foi desenvolvida utilizando linguagem Python, a biblioteca CustomTkinter e algumas classes base diretas do tkinter. Ao iniciar aplicação, é apresentado o menu principal, conforme visto na Figura 21, que lista as principais funcionalidades da aplicação. Nele, estão disponíveis três opções: “Analisar Vídeo”, que permite ao usuário iniciar o processo de envio e classificação de um vídeo de exercício, “Exercícios Disponíveis”, que direciona a uma seção com a lista dos exercícios reconhecidos pelo modelo e seus respectivos detalhes técnicos, e “Informações / Ajuda”, que fornece orientações sobre o uso da aplicação e boas práticas para gravação dos vídeos.

Figura 21 – Menu principal da aplicação.



Fonte: elaborado pelo autor.

Ao selecionar “Informações / Ajuda”, a partir do menu principal, é aberta a tela correspondente, representada na Figura 22, desenvolvida para orientar o usuário sobre a proposta geral da aplicação. É apresentada uma breve explicação da funcionalidade das estimativas geradas pelo OpenPose, o processamento da rede neural e os fatores que podem interferir na qualidade do resultado obtido, proporcionando maior transparência ao processo. Em seguida, são fornecidas algumas instruções para a captura de vídeo, abordando o posicionamento da câmera, ângulos de filmagem e outras boas práticas, similar ao apresentado anteriormente nesse artigo, porém, de maneira mais didática, contendo inclusive as mesmas imagens de exemplo para exemplificar os ângulos ideais para a captura de vídeos.

Figura 22 – Tela de informações e ajuda.



Fonte: elaborado pelo autor.

Voltando ao menu principal, o usuário pode acessar a tela "Exercícios Cadastrados" por meio do botão correspondente. Nessa tela, são apresentados os três exercícios mapeados pelo sistema: Agachamento com Barra, Levantamento Terra com Barra e Supino Reto com Barra. Cada um é representado por um botão com o nome do exercício, permitindo que o usuário selecione qual deseja explorar com mais detalhes, conforme visto na Figura 23.



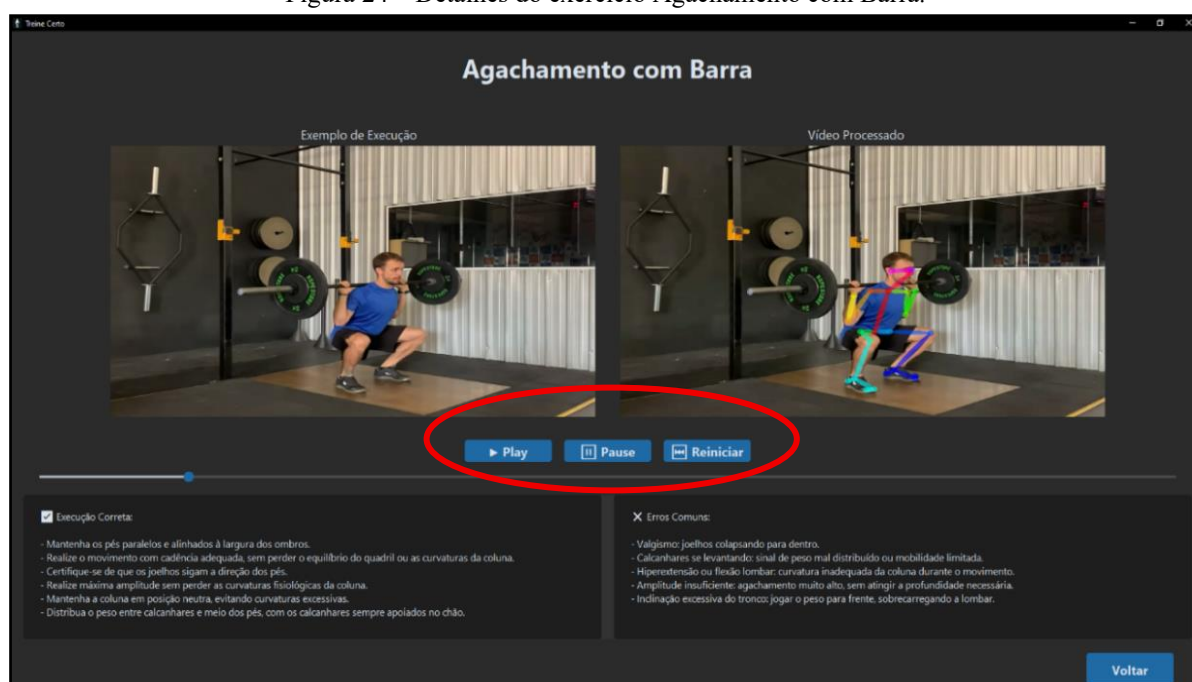
Figura 23 – Tela com a lista de exercícios disponíveis.



Fonte: elaborado pelo autor.

Ao selecionar um dos exercícios listados, o usuário é redirecionado para a tela de detalhes, onde são exibidas informações técnicas sobre a execução correta do movimento e alguns dos principais erros cometidos por praticantes. Essas informações são apresentadas em duas caixas de texto lado a lado, intituladas “Execução Correta” e “Erros Comuns”, que se baseiam em fundamentos da biomecânica e recomendações descritas pelo coorientador do projeto, um profissional da área de educação física e biomecânica aplicada. A interface também exibe dois vídeos simultâneos: um exemplo da execução correta do exercício, realizada por um profissional da área e o mesmo vídeo após ter sido processado pela aplicação, contendo a sobreposição dos pontos-chave e suas ligações anatômicas. Botões de controle (Play, Pause e Reiniciar) dos vídeos permitem controlá-los simultaneamente para analisar o resultado do processamento da aplicação em um momento específico. Essa tela, contendo os detalhes do exercício agachamento com barra, pode ser vista na Figura 24.

Figura 24 – Detalhes do exercício Agachamento com Barra.

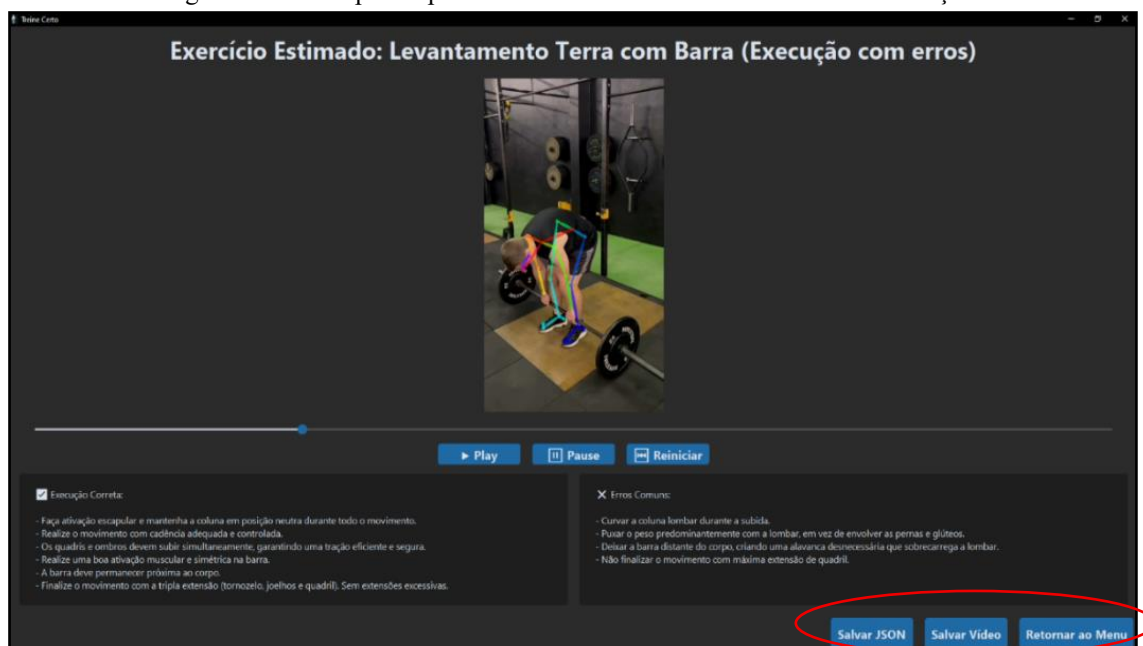


Fonte: elaborado pelo autor.

Já o botão “Analisar Vídeo”, disponível também na tela inicial - Figura 23, abre uma janela do sistema operacional para a seleção de um arquivo .mp4. Após a seleção, o usuário é automaticamente redirecionado para a tela “Processando Vídeo”, que exibe uma barra de progresso animada enquanto o sistema realiza a análise do vídeo em outra *thread*, através das classes desenvolvidas anteriormente para encapsulamento das rotinas da rede neural.

Uma vez completo o processamento do vídeo, a tela de detalhes é aberta, apresentando qual exercício foi estimado, sua corretude, o vídeo processado com controles de reprodução, as recomendações técnicas associadas ao tipo de movimento identificado e as opções de “Salvar JSON” e “Salvar Vídeo”, permitindo exportar os resultados gerados pela rede, conforme previsto nos requisitos previamente definidos. Um exemplo dessa tela pode ser visto na Figura 25, com o resultado do processamento de um vídeo contendo a execução de levantamento terra com barra com erros, nesse caso a falha em manter a coluna neutra, gerando o arqueamento das costas.

Figura 25 – Exemplo de processamento de um vídeo com erros de execução.



Fonte: elaborado pelo autor.

O processamento do vídeo resulta na geração de um arquivo JSON estruturado conforme o padrão previamente apresentado, contendo os dados gerais do vídeo e dos pontos em cada quadro. Essa padronização visa permitir que os dados possam ser reutilizados em outras aplicações externas. Ao optar por salvar o vídeo, a aplicação salva o vídeo apresentado em formato .mp4, contendo a sobreposição dos pontos-chave detectados, o que possibilita o compartilhamento dos resultados em ambientes externos, como em avaliações com profissionais da área ou em relatórios para acompanhamento do progresso do usuário.

### 3.4 AVALIAÇÃO DA APLICAÇÃO

Essa seção aborda o processo de avaliação da aplicação e seus resultados. Para validar a aplicabilidade e a usabilidade da aplicação desenvolvida, foi realizada uma avaliação empírica com a participação de 18 voluntários, selecionados entre pessoas com diferentes níveis de formação e familiaridade com a prática esportiva. O processo avaliativo foi conduzido através de um formulário estruturado em forma de roteiro de tarefas guiadas.

As etapas do formulário foram estruturadas para conduzir os participantes por cada uma das funcionalidades principais da aplicação, solicitando que realizassem operações como: consultar informações técnicas, visualizar exemplos de exercícios, submeter vídeos à análise e interpretar os resultados apresentados. Após cada atividade, os voluntários responderam a uma série de perguntas baseadas na Escala de Likert, variando de 1 (discordo totalmente) a 5 (concordo totalmente), com todas as questões relacionando a concordância com um aspecto positivo da aplicação. Além dessas questões objetivas, foram apresentados campos abertos para observações qualitativas relacionadas a cada etapa e sobre aspecto gerais da aplicação, ao final do questionário.

Os respondentes apresentaram perfis variados de experiência na área, com 9 participantes (50%) declarando-se praticantes regulares, 4 participantes (22,22%) indicando não terem nenhum tipo de formação ou experiência, 3 (16,7%) sendo pós-graduados ou especialista, 1 respondente sendo graduado e 1 sendo atualmente um estudante em formação na área. Essa variação de experiência ajuda a proporcionar uma visão mais geral de diferentes públicos sobre a ferramenta.

A primeira etapa da avaliação instrui o usuário a acessar a seção de informação e ajuda da aplicação, através do menu inicial. Avaliando a apresentação da proposta, 17 participantes (94,4%), afirmaram concordar totalmente que as informações sobre a aplicação auxiliam a melhor compreender sua proposta e funcionalidade, enquanto apenas 1 (5,6%) respondeu "Concordo". Sobre a clareza das recomendações, 14 participantes (77,8%) concordaram totalmente que as recomendações são claras e relevantes para obter bons resultados na análise de vídeo. Outros 2 (11,1%) indicaram "Concordo" e 2 (11,1%) mantiveram neutralidade ("Não concordo nem discordo"). Sobre as imagens de exemplo para ângulos de captura de vídeos, 15 participantes (83,3%) concordaram totalmente que as imagens de exemplo ajudaram na compreensão das instruções, e 3 (16,7%) responderam "Concordo". Utilizando os dados apresentados nessa seção de ajuda e informações, 14 participantes (77,8%) concordam totalmente que se sentiriam preparados para gravar um vídeo compatível com as recomendações, enquanto 3 (16,7%) concordaram e 1 (5,6%) manteve neutralidade. A elevada concordância nas respostas desta etapa indica que as informações e exemplos disponibilizados pela aplicação foram eficazes em orientar os usuários quanto às melhores práticas de gravação, contribuindo para a produção de vídeos mais

adequados e, consequentemente, para a obtenção de resultados mais consistentes na análise automatizada. As perguntas dessa etapa e seus resultados podem ser vistos na Tabela 2.

Tabela 2 – Resultados de perguntas sobre as informações

Pergunta \ Resposta	Discordo totalmente.	Discordo.	Não concordo nem discordo.	Concordo.	Concordo totalmente.
As informações sobre a aplicação auxiliam a compreender sua proposta e funcionalidade.	0	0	0	1	17
As recomendações apresentadas são claras e relevantes para obter bons resultados na análise de vídeo.	0	0	2	2	14
As imagens de exemplo ajudaram na compreensão das instruções.	0	0	0	3	15
Eu me sentiria preparado para gravar um vídeo após ler essa seção.	0	0	1	3	14

Fonte: elaborado pelo autor.

A etapa seguinte conduz o respondente para a tela que lista os exercícios implementados, juntamente com informações e exemplos da execução de cada um. Ao analisar as informações textuais apresentadas, 16 participantes (88,9%) concordaram totalmente que as informações sobre execução correta e erros apresentadas são claras e úteis, 1 (5,6%) responderam “Concordo” e novamente 1 (5,6%) foi neutro. Todos os participantes avaliaram positivamente os vídeos de exemplo, com 16 (88,9%) concordando totalmente que eles auxiliam a entender como reproduzir o exercício corretamente e 2 (11,1%) indicando “Concordo”. Novamente, a alta concordância nessa etapa indica a relevância e usabilidade providas pelas informações textuais acompanhadas de vídeos de exemplo apresentados para cada exercício disponível. As perguntas dessa etapa e seus resultados podem ser vistos na Tabela 3.

Tabela 3 – Resultados de perguntas sobre os exercícios disponíveis

Pergunta \ Resposta	Discordo totalmente.	Discordo.	Não concordo nem discordo.	Concordo.	Concordo totalmente.
As informações sobre execução correta e possíveis erros são claras e úteis.	0	0	1	1	16
Os vídeos de exemplo auxiliam a entender como reproduzir o exercício corretamente.	0	0	0	2	16

Fonte: elaborado pelo autor.

Em seguida, a terceira etapa leva ao processo de análise de um vídeo. Sobre a precisão das marcações de pontos e qualidade visual do vídeo resultante, 12 participantes (66,7%) concordaram totalmente que as marcações são precisas e bem-posicionadas, 4 (22,2%) concordaram e 2 (11,1%) mantiveram neutralidade. Avaliando a identificação do exercício executado e a classificação de sua corretude, 13 participantes (72,2%) concordaram totalmente que a classificação foi coerente, 3 (16,7%) concordaram e 2 (11,1%) indicaram neutralidade. Sobre a clareza das informações apresentadas após a análise, 15 participantes (83,3%) concordaram totalmente que as informações apresentadas após a análise são claras e compreensíveis, 2 (11,1%) concordaram e 1 (5,6%) foi neutro. Todos os usuários concordaram com a facilidade do processo de salvamento do vídeo e arquivo JSON gerado pela análise, 12 participantes (66,7%) concordaram totalmente que conseguiram realizar as ações sem dificuldades, enquanto 6 (33,3%) concordaram. Esses resultados apontam uma experiência positiva com os resultados obtidos na análise de vídeos, com uma distribuição levemente mais neutra no processo de salvamento, indicando talvez que essa parte não seja tão simples ou intuitiva, bem como uma maior quantidade de respostas neutras nas perguntas sobre avaliação de corretude podem indicar uma incerteza da parte do público em sua capacidade de validar esses resultados. As perguntas dessa etapa e seus resultados podem ser vistos na Tabela 4.

Tabela 4 – Resultados de perguntas sobre o processamento de um vídeo.

Pergunta \ Resposta	Discordo totalmente.	Discordo.	Não concordo nem discordo.	Concordo.	Concordo totalmente.
As marcações de pontos sobre os vídeos são precisas e bem posicionadas.	0	0	2	4	12
A classificação entre execução correta e incorreta foi coerente.	0	0	2	3	13
As informações apresentadas após a análise do vídeo são claras e compreensíveis.	0	0	1	2	15
Eu consegui realizar as ações de salvar sem dificuldades.	0	0	0	6	12

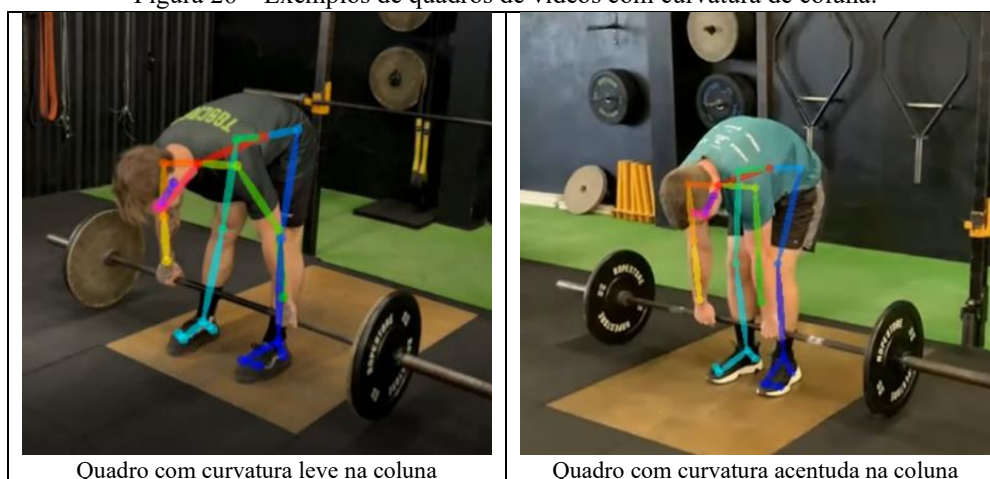
Fonte: elaborado pelo autor.

De forma geral, os resultados obtidos nas perguntas qualitativas indicam uma experiência positiva e uma concordância alta com as afirmações sobre a facilidade do uso da aplicação, as informações providas para a execução dos exercícios físicos abordados e a assertividade dos resultados obtidos pelos usuários. Além disso, foram feitas perguntas qualitativas abertas para coletar opiniões e sugestões.

Na pergunta de comentários gerais da aplicação, a maioria dos participantes destacou de maneira positiva a capacidade do programa estimar os pontos-chave do corpo humano, mesmo quando oclusos em determinados quadros do vídeo, com alguns profissionais da área destacando a utilidade da visualização desses pontos articulares e suas ligações para análises de particularidades dos exercícios, facilitando a visualização de proporções corporais e relacionamentos ergonômicos, podendo utilizar o vídeo gerado pela aplicação para fins fora do contexto inicialmente proposto. Dois comentários também ressaltaram a utilidade das instruções para execução correta e possíveis erros disponibilizados, afirmando sua utilidade especialmente para o público leigo ou iniciante no esporte.

Um respondente destacou a limitação de representar a coluna como uma linha reta que conecta o pescoço ao centro do quadril, o que não evidencia desvios posturais como a curvatura da coluna. Exemplos dessa limitação podem ser observados na Figura 26, que apresenta dois quadros de vídeos de praticantes com curvaturas visíveis na região da coluna, uma mais sutil no primeiro caso e outra mais acentuada no segundo. No entanto, a sobreposição gerada representa ambos com uma linha reta entre o centro do quadril e a base do pescoço, o que evidencia a dificuldade do sistema em capturar deformações da coluna por meio dos pontos-chave utilizados.

Figura 26 – Exemplos de quadros de vídeos com curvatura de coluna.



Fonte: elaborado pelo autor.

Além disso, foi exposto por alguns respondentes a dificuldade da obtenção de um ângulo para evitar oclusão de partes do corpo durante a execução do supino reto com barra. Esta particularidade já foi observada durante a captura de dados para o treinamento e validação da rede neural, necessitando de uma atenção mais deliberada para o posicionamento da câmera do que os demais exercícios, que apresentam a maioria de seus pontos anatomicamente relevantes nos ângulos comumente utilizadas para gravação. Esse exercício foi inclusive o único que apresentou erros de classificação pelo modelo durante o treinamento, vide a matriz de confusão apresentada anteriormente na Figura 18.

Quanto aos contextos nos quais essa aplicação pode ser útil, a maioria das respostas apontou a sua utilidade como ferramenta de autoavaliação, com destaque para praticantes iniciantes e intermediários, devido às instruções apresentadas e a capacidade do programa apontar a corretude da execução, evidenciando erros que possivelmente não seriam percebidos por esses praticantes e sendo uma alternativa mais acessível do que o acompanhamento de um profissional dedicado para essa análise individual. O segundo contexto mais mencionado foi o de consultoria online, tanto para os alunos compreenderem mais facilmente os exercícios propostos, como para ser utilizada como ferramenta de análise por parte do professor.

A última pergunta se trata de uma sessão para sugestões de melhorias na aplicação. Oito dos dezoito respondentes afirmaram que o desenvolvimento de uma interface para dispositivos móveis tornaria a aplicação mais acessível e prática para o uso e captura de vídeos durante sua prática, tanto em academias comerciais como em casa, além de modernizar sua usabilidade geral. O segundo tópico mais abordado, mencionado por seis pessoas, envolve a adição de mais exercícios e variações no futuro e a capacidade de apontar mais detalhadamente qual erro foi detectado na execução, possivelmente se relacionando diretamente com os erros comuns apontados na descrição dos exercícios.

## 4 CONCLUSÕES

Diante do crescente número de praticantes de exercícios físicos e dos riscos associados à execução inadequada, como lesões e baixa eficiência nos treinos, em razão da limitada acessibilidade ao acompanhamento profissional qualificado, este trabalho teve como proposta o desenvolvimento de uma aplicação voltada à análise da execução de exercícios físicos por meio de técnicas de visão computacional e aprendizado de máquina. A aplicação utiliza dados posicionais de 25 pontos-chave do corpo humano, extraídos por meio da API do OpenPose, e os processa com um modelo híbrido de redes neurais. Três exercícios clássicos do treinamento de força foram selecionados para o projeto: levantamento terra com barra, supino reto com barra e agachamento com barra.

O primeiro objetivo específico do trabalho consistia em desenvolver um modelo de aprendizado de máquina híbrido, combinando Redes Convolucionais de Grafos (GCNs) com Redes LSTM, capaz de avaliar a corretude da postura e do movimento durante a execução de exercícios físicos. Esse objetivo foi alcançado por meio da implementação de uma arquitetura que modela relações entre as poses de um vídeo com uma GCN e, em seguida, explora a dinâmica temporal dos movimentos com uma camada LSTM, complementada por um mecanismo de atenção, utilizando como dados de entrada as coordenadas estimadas pelo OpenPose.

O segundo objetivo era identificar os principais fatores que influenciam a precisão do modelo na avaliação da técnica de execução. Para isso, foram conduzidas buscas por materiais de referência sobre as capacidades da ferramenta, testes práticos e análises dos erros de classificação, com ênfase nos vídeos rotulados incorretamente. As observações revelaram que erros sutis dentro de séries longas e ângulos de filmagem desfavoráveis foram fatores determinantes para a qualidade do modelo. Esses detalhes auxiliaram a criar diretrizes para a captura de vídeos que sejam mais bem interpretados pela rede.

O terceiro objetivo tratava do desenvolvimento de uma interface *desktop* funcional e amigável, que integrasse o modelo de aprendizado de máquina e permitisse ao usuário visualizar a análise de sua execução. Esse objetivo também foi atendido por meio do desenvolvimento de uma aplicação que permite a consulta de exercícios, detalhes de sua execução ou possíveis erros e o envio de vídeos para classificação automática por meio da rede. A aplicação foi submetida a um teste empírico, conduzido através de um formulário baseado em tarefas práticas guiadas, permitindo aos participantes experimentarem as funcionalidades da aplicação e registrar sua percepção quanto à usabilidade e à utilidade do sistema. Ao ser respondida por dezoito voluntários, de diferentes perfis e níveis de experiência com a prática esportiva, os resultados demonstraram elevado grau de aceitação e satisfação quanto ao uso da aplicação. A maioria absoluta dos participantes avaliou positivamente aspectos como a clareza das informações, a utilidade das recomendações técnicas, a qualidade dos vídeos processados e a precisão das análises realizadas pelo modelo.

Embora a amostra de participantes seja pequena, os resultados obtidos oferecem indícios promissores sobre a usabilidade da aplicação e sua receptividade junto a diferentes perfis de usuários. Para validações futuras mais robustas, recomenda-se a realização de testes com uma amostra maior e mais diversificada, que permita uma análise estatística mais abrangente e representativa.

O desenvolvimento do sistema demandou um processo de coleta de dados, realizado através da gravação de vídeos de exercícios físicos executados por voluntários e supervisionados por profissionais da área. No total, foram capturados 430 vídeos, formando um conjunto de dados utilizados para o treinamento e validação do modelo de rede neural, garantindo que a aplicação fosse capaz de reconhecer diferentes padrões de execução, tanto corretos quanto incorretos, com base em exemplos reais e seguindo orientações de execução. Para o treinamento do modelo de aprendizado de máquina proposto utilizou-se os dados obtidos, técnicas de aumento de dados, mecanismos de memória e atenção e mostrou-se capaz de identificar exercícios físicos e a corretude de sua execução com uma acurácia de validação de 99,66% e perda de 0,0195 após percorrer 184 épocas.

Em relação aos trabalhos correlatos, Pham *et al.* (2022) propuseram uma aplicação *web* capaz de auxiliar praticantes inexperientes através do reconhecimento em tempo real de exercícios físicos pela análise de imagens RGB. Já Rangari *et al.* (2022) focaram no desenvolvimento de um modelo de rede neural para estimar a posição tridimensional de juntas do corpo humano através de imagens bidimensionais e avaliar a correteza postural durante prática de exercícios físicos, enquanto Singh *et al.* (2022) buscaram desenvolver um algoritmo focado em classificar a correteza do exercício de desenvolvimento militar com barra através do processamento de vídeos, com menor tempo de execução e requisitos de infraestrutura que outras proposta mais convencionais.

Diante dos resultados obtidos e apresentados, conclui-se que a aplicação desenvolvida atingiu seu objetivo de auxiliar na classificação da execução de exercícios físicos e sua correteza através da análise de vídeos RGB, apresentando potencial prático significativo para ser utilizada tanto como ferramenta de autoavaliação, por iniciantes ou até praticantes experientes, quanto como suporte a profissionais da área de Educação Física em contextos como aplicação e acompanhamento de consultorias remotas. Apesar de possuir certas particularidades como ângulos de capturas de vídeo para a obtenção de melhores resultados, devido à fatores como a oclusão de pontos do corpo, os usuários apontaram satisfação com a usabilidade e os resultados obtidos, mas destacaram o fato da aplicação ter sido desenvolvida exclusivamente para ambiente Windows como uma possível limitação para sua adoção, sugerindo o desenvolvimento de uma aplicação para dispositivos móveis.

Por fim, as possíveis extensões para esse trabalho são: 1) desenvolvimento de uma aplicação *mobile* e processamento remoto dos vídeos e classificação pela rede neural; 2) apresentar detalhadamente quais erros foram encontrados em uma determinada execução, ao invés de apenas classificá-la como errada; 3) disponibilizar mais exercícios e variações.

## REFERÊNCIAS

- AMADIO, Alberto Carlos; SERRÃO, Júlio Cerca. A Biomecânica em Educação Física e Esporte. **Rev. bras. Educ. Fis. Esporte**, São Paulo, v. 25, p. 15-24, dez. 2011.
- AMADIO, Alberto Carlos *et al.* Introdução à análise do movimento humano: descrição e aplicação dos métodos biomecânicos de medição. **Revista Brasileira de Fisioterapia**, v. 3, n. 2, p. 41-54, 1999.
- AMRUTHA, Kallingle; PRABU, P.; PAULOSE, Joy. Human Body Pose Estimation and Applications. In: 2021 INNOVATIONS IN POWER AND ADVANCED COMPUTING TECHNOLOGIES (I-PACT), I, 2021, Kuala Lumpur. **Proceedings...** Kuala Lumpur, IEEE, 2021. p. 1-6.
- BALDINGER, Melanie; REIMER, Lara Marie; SENNER, Veit. Influence of the Camera Viewing Angle on OpenPose Validity in Motion Analysis. **Sensors**, v. 25, n. 3, p. 799, 28 jan. 2025.
- BARTLETT, Roger. **Introduction to Sports Biomechanics: Analysing Human Movement Patterns**. London: Routledge, 2014.
- BOESCH, Gaudenz. **The Complete Guide to OpenPose in 2024**. 15 de dez. de 2023. Disponível em: <<https://viso.ai/deep-learning/openpose/>>. Acesso em: 14 set. 2024.
- BRITO, Geisson C. *et al.* Adesão A Prática De Musculação Nas Academias Do Brasil. **Revista CPAQV - Centro de Pesquisas Avançadas em Qualidade de Vida**, Belo Horizonte, v. 13, n. 1, p. 1-7, nov. 2021.
- CAMARGO, Edina Maria de; AÑEZ, Ciro Romelio Rodriguez. **Diretrizes da OMS para atividade física e comportamento sedentário: num piscar de olhos**. Edina De Camargo, 2020.
- CAO, Z. *et al.* **OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields**. 2018. Disponível em: <<http://arxiv.org/abs/1812.08008>>.
- CASPERSEN, C. *et al.* Physical activity, exercise, and physical fitness: definitions and distinctions for health-related research. **Public health reports**, Washington D.C, v. 100, n. 2, p. 126-131, 1985.
- CUESTA-VARGAS, Antonio; GALÁN-MERCANT, Alejandro; WILLIAMS, Jonathan M. The use of inertial sensors system for human motion analysis. **Physical Therapy Reviews**, v. 15, n. 6, p.462-473, dez. 2010.
- ESPITIA-MORA, Laura Alejandra; VÉLEZ-GUERRERO, Manuel Andrés; CUERVO, Mauro Callejas. “Development of a Low-Cost Markerless Optical Motion Capture System for Gait Analysis and Anthropometric Parameter Quantification”. **Sensors**, v. 24 n. 11, mai. 2024.
- HAUPENTHAL, Alessandro; PEREIRA, Natália Duarte; MICHAELSEN, Stella Maris. Um Movimento, Dois Olhares: a mudança de paradigma na análise do controle motor e seu efeito na abordagem da marcha humana. **Revista Brasileira de Ciência e Movimento**, v. 18, n. 2, p.107-115, 2010.



HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. **Neural Computation**, v. 9, n. 8, p. 1735–1780, nov. 1997.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA, Coordenação de Trabalho e Rendimento, organizador. **Pesquisa nacional de saúde**. IBGE, 2021.

KE, YiRan *et al.* Recognition technology of human body movement behavior in fitness exercise based on transfer learning. In: 2022 IEEE NINTH INTERNATIONAL CONFERENCE ON COMMUNICATIONS AND ELECTRONICS (ICCE), IX, 2022, Nha Trang. **Proceedings...** Nha Trang, IEEE, 2022. p 349–354.

KIPF, Thomas N.; WELING Max. **Semi-Supervised Classification with Graph Convolutional Networks**. 2016. Disponível em: <<http://arxiv.org/abs/1609.02907>>.

MALLMANN, Laura. **Conheça os benefícios do treinamento com personal trainer**. 23 de nov. de 2020. Disponível em: <<https://www.univates.br/noticia/28910-conheca-os-beneficios-do-treinamento-com-personal->>. Acesso em: 14 set. 2024.

MINISTÉRIO DA SAÚDE. **Vigitel Brasil 2006-2021: vigilância de fatores de risco e proteção para doenças crônicas por inquérito telefônico**. Ministério da Saúde, 2022.

NEON. **Quanto custa um personal trainer? Saiba se vale a pena ter um**. [S.l.], [2024]. Disponível em: <<https://neon.com.br/aprenda/financas-pessoais/quanto-custa-um-personal-trainer/>>. Acesso em: 14 set. 2024.

NOVACHECK, Tom F. The biomechanics of running. **Gait & Posture**, v.7, n.1, p. 77-95, jan. 1998.

PERES, A. et al. Accuracy of hidden Markov models in identifying alterations in movement patterns during biceps-curl weight-lifting exercise. **Applied sciences (Basel, Switzerland)**, v. 13, n. 1, p. 573, 2022.

PHAM, Quang-Tien *et al.* Automatic recognition and assessment of physical exercises from RGB images. **ASIAN CONFERENCE ON COMPUTER VISION**, XVI, 2022, Macao, **Proceedings...** Switzerland, Springer Nature, 2022. p 189–205.

RANGARI, Tushar. Video based exercise recognition and correct pose detection. **Multimedia Tools and Applications**, v. 81, n. 19, p. 30267–30282, abri. 2022.

SHIN, Soyong; LI, Zhixiong; HALILAJ, Enj. Markerless Motion Tracking with Noisy Video and IMU Data. **IEEE Transactions on Biomedical Engineering**, v.70, p. 3082–3092, nov. 2023.

SILVA, Paula Fernanda de Sous. Novo Contexto Cultural Do Exercício Físico Na Era Tecnológica. **PISTA: Periódico Interdisciplinar: Sociedade Tecnologia Ambiente**, Belo Horizonte, v. 3, n. 2, p.100–108, ago./nov. 2021.

SINGH, Ashish *et al.* Fast and robust video-based exercise classification via body pose tracking and scalable multivariate time series classifiers. **Data Mining and Knowledge Discovery**, v. 37, n. 2, p. 873–912, 21 dez. 2022.

SHORTEN, Connor; KHOSHGOFTAAAR, Taghi. M. A survey on image data augmentation for deep learning. **Journal of big data**, v. 6, n. 1, 2019.

TECNOFIT. **Qual a importância do personal trainer no treino de musculação?** [S.l.], [2024]. Disponível em: <<https://www.tecnofit.com.br/blog/a-importancia-do-personal-trainer-no-treino-de-musculacao/>>. Acesso em: 23 set. 2024.

VASWANI, Ashish *et al.* Attention is All you Need. **Advances in Neural Information Processing Systems**, v. 30, 2017.

VECCHIO, Luke de.; DAEWOUD, Hays. The health and performance benefits of the squat, deadlift, and bench press. **MOJ Yoga & Physical Therapy**, v. 3, n. 2, 6 abr. 2018.

WU, Zonghan *et al.* A Comprehensive Survey on Graph Neural Networks. **IEEE Transactions on Neural Networks and Learning Systems**, v. 32, n. 1, p. 4–24, jan. 2021.

YAN, Sijie; XIONG, Yuanjun; LIN, Dahua. Spatial temporal graph convolutional networks for skeleton-based action recognition. In: **AAAI CONFERENCE ON ARTIFICIAL INTELLIGENCE**, 32, Nova Orleans. **Proceedings...** AAAI, 2018. p. 7444–7452.

ZHANG, Mingming. *et al.* Multi-view emotional expressions dataset using 2D pose estimation. **Scientific Data**, v. 10, n. 1, p. 649, 22 set. 2023.

ZHAO, Ziyi. *et al.* **3D pose based feedback for physical exercises**. 2022. arXiv. Disponível em: <<http://arxiv.org/abs/2208.03257>>.