Revista de Informática Teórica e Aplicada - RITA - ISSN 2175-2745
Vol. 31, Num. 01 (2024) 11-31

RESEARCH ARTICLE

Detection and Pose Adjustment in Physical Exercises using Computer Vision Techniques: Approaches, Challenges and Opportunities

Deteção e Ajuste de Postura em Exercícios Físicos usando Técnicas de Visão Computacional: Abordagens, Desafios e Oportunidades

João Gonçalves[1], João Palhares[1], Vasco N. G. J. Soares[1,2], Paulo A. C. S. Neves[1]*

Abstract: In the fast-paced rhythm of modern life, the regular practice of physical exercise emerges as a source of vitality and well-being. However, it is not always feasible to practice on a gym and/or pay a personal trainer to assure a correct pose on calisthenic exercises. This article aims to investigate technological and scientific advancements in the field of computer vision to enhance exercise detection and execution, with the goal of creating a system that can be used autonomously by users. To achieve this, an introduction to the fundamental concepts inherent to the subject is provided, followed by an analysis of different technological approaches, identifying their strengths and weaknesses. Finally, the various challenges and limitations of these technologies are discussed, which remain open to be solved and explored in future research endeavors.
Keywords: Body Detection – Pose Estimation – Convolutional Neural Network – Algorithms – Survey

Resumo: No ritmo acelerado da vida moderna a prática regular de exercício físico emerge como fonte de vitalidade e bem-estar. No entanto nem sempre é possível a deslocação ao ginásio e/ou pagar a um treinador pessoal para garantir a postura correcta na realização de exercícios de ginástica. Este artigo tem como objetivo investigar os avanços tecnológicos e científicos na área da visão computacional para melhorar a deteção e aprimorar a execução de exercícios físicos, visando criar um sistema que possa ser utilizado de forma autónoma pelos utilizadores. Para isso, é feita uma introdução aos conceitos fundamentais, inerentes à temática, seguido de uma análise das diferentes abordagens tecnológicas, identificando os seus pontos fortes e fracos. Para finalizar são discutidos os vários desafios e limitações destas tecnologias, que permanecem em aberto para serem solucionados e explorados em investigações futuras.
Palavras-Chave: Deteção corporal – Estimação de Pose – Rede Neuronal Convolucional – Algoritmos – Estado da Arte

[1] *Instituto Politécnico de Castelo Branco, Av. Pedro Álvares Cabral nº 12, 6000-084 Castelo Branco, Portugal*
[2] *Instituto de Telecomunicações, Rua Marquês d'Ávila e Bolama, 6201-001 Covilhã, Portugal*
***Corresponding author**: pneves@ipcb.pt
DOI: http://dx.doi.org/10.22456/2175-2745.135436 • **Received:** 09/09/2023 • **Accepted:** 16/01/2024

1. Introdução

Nos últimos anos, assistimos a um notável crescimento na procura de sistemas que possam detetar e interpretar movimentos do corpo humano. Esta procura não é apenas um capricho, mas sim uma resposta às exigências de várias áreas, desde a saúde à robótica, passando pela segurança, o desporto e os videojogos. Estes sistemas, baseados em tecnologia de visão computacional, têm o potencial de transformar a forma como interagimos com o mundo e com a tecnologia, potenciando a criação para aplicações inovadoras e mais abrangentes [1].

Uma tendência que tem vindo a ganhar expressão é a procura crescente de alternativas aos ginásios tradicionais. Esta ânsia ganhou ainda mais destaque com o surgimento da pandemia de COVID-19, que forçou restrições rigorosas e quarentenas generalizadas, impossibilitando muitas pessoas de frequentar ginásios e espaços públicos de exercício. As empresas e as indústrias foram então desafiadas a adaptar-se rapidamente a novas realidades, operando remotamente e procurando soluções que permitissem manter um estilo de vida ativo e saudável, mesmo sem acesso aos recursos tradicionais [2, 3].

Neste contexto, surgiram diversas aplicações e programas que prometem oferecer a possibilidade de efetuar treinos sem

a necessidade de equipamento especializado ou saídas de casa. No entanto, uma grande parte destes sistemas é limitada em termos de eficácia, baseando-se maioritariamente na criação de planos de treino predefinidos. Esta abordagem apresenta desafios relacionados com a qualidade da execução dos exercícios, uma vez que não há supervisão profissional para corrigir a postura e a técnica [4]. É neste contexto que este artigo se insere.

O objetivo principal é explorar soluções tecnológicas e compreender as tecnologias subjacentes à criação de um sistema capaz de detetar e interpretar os movimentos do corpo humano durante exercícios físicos. Uma das principais metas é conseguir informar o utilizador quando um exercício está a ser executado de forma incorreta, potenciando assim a correção em tempo real, ajudando desta forma a evitar possíveis lesões.

Apesar da inovação promissora, é importante reconhecer que ainda existem aspetos que tais aplicações não conseguem replicar, tornando a opção de frequentar ginásios bastante válida e complementar. No entanto, várias metodologias estão em desenvolvimento que pretendem abordar estas limitações de forma eficiente.

O processo proposto para atingir estes objetivos envolve várias etapas, começando pela captura contínua de imagem em tempo real dos movimentos. Posteriormente estas imagens são processadas, efetuando a deteção da pessoa, identificação dos movimentos, e verificação da sua execução correta. Finalmente é efetuada a indicação das correções necessárias. Ainda que seja uma tarefa ambiciosa, a segmentação destas etapas permitirá uma abordagem mais eficaz aos desafios, tendo como resultado um sistema mais eficiente e útil para os utilizadores.

Os autores começaram por procurar sínteses de estado da arte na área, tendo concluído que existem para a área genérica de visão computacional, mas não conseguiram encontrar na vertente específica de deteção de pose e auxílio na realização de exercícios físicos. Assim sendo,a informação compilada ao longo deste artigo baseia-se em estudos obtidos de diferentes base de dados científicas, nomeadamente IEEE Explore, ScienceDirect, arXiv e ResearchGate. A primeira pesquisa retornou cerca de 70 artigos, que encaixavam nos critérios estabelecidos. Foi efetuada uma filtragem, com base na data de publicação, onde foram escolhidos artigos apenas a partir de 2019 inclusive, por ser um intervalo de tempo relativamente abrangente e com ênfase nos avanços mais recentes. Foram ainda escolhidos artigos que expusessem implementações práticas, ou que apresentassem um contributo relevante nesse sentido. O principal foco desta pesquisa concentra-se em duas questões: (a) Deteção e classificação de ações humanas em exercícios físicos e (b) Identificação e correção na execução dos exercícios. Nesse sentido o principal critério de seleção foi a adequação ao objetivo que o trabalho se propunha - deteção de pose e acompanhamento na execução de exercícios físicos.

Para alcançar os objetivos delineados, é essencial adquirir um amplo espectro de dados pertinentes. Estes dados estão fortemente conectados com parâmetros associados ao organismo humano e ao ambiente em que se encontra inserido. Dentro deste contexto, destaca-se principalmente a obtenção de informação corporal e posicional relativa à pessoa em questão [5]. Para obter estas informações existem duas abordagens principais, sendo imperativo saber qual a que se adequa melhor para a situação em que se pretende aplicar. Estas opções são a utilização de sensores estrategicamente colocados sobre o corpo ou utilização de técnicas de visão computacional através de obtenção de imagens com câmaras digitais[6, 7]. Devido à natureza da tarefa que é proposta neste documento, foi decidido incidir sobre sistemas baseados em visão computacional, uma vez que se torna mais simples e acessível para os utilizadores.

Este documento está estruturado em várias secções. A Secção 2 apresenta as diferentes técnicas de visão computacional relacionadas com a temática em questão. A Secção 3 apresenta os diversos estudos de investigação na área. A Secção 4 discute os desafios e oportunidades, enquanto a Secção 5 conclui e aponta oportunidades para trabalho futuro.

2. Técnicas de Visão Computacional

Computer vision (CV) ou visão computacional é uma disciplina interligada à *Artificial Intelligence* (AI) e suas várias subáreas. O foco principal é conferir às máquinas a capacidade de assimilar informações a partir de imagens e empregar técnicas visuais para aquisição e análise de dados. Com essa finalidade, estes sistemas empregam diferentes mecanismos para captar os dados, essencialmente câmaras, muitas vezes incorporadas com outras tecnologias, conferindo uma maior versatilidade face a diferentes tipos de situações. Com esta etapa realizada, são aplicados diferentes algoritmos e metodologias de *deep learning* [8] de forma a tratar e trabalhar os dados, culminando na geração de resultados estatísticos que permitem a deteção de caraterísticas (*feature extraction*).

Este tipo de tecnologia é formado por algoritmos que fornecem conhecimento à máquina, com base num grande conjunto de dados através de observação contínua, suprimindo a necessidade de intervenção direta de um ser humano para que seja realizado a aprendizagem [9]. Esses algoritmos funcionam com base em várias camadas de processamento, tornando possível classificar um determinado input fornecido ao sistema [10]. *Human action recognition* (HAR), é uma subárea dentro da *computer vision*, que foca todas estas técnicas e conceitos de forma a usá-los para a deteção do ser humano e todo o tipo de aplicações relacionadas com o mesmo. Os principais contribuintes no estudo e desenvolvimento dentro da HAR são as *Convolutional Neural Networks* (CNN), *ConvNet* ou redes neuronais convolucionais, que serão aprofundadas em mais detalhe na próxima secção.

2.1 Arquiteturas CNN

Uma CNN é um tipo de arquitetura aplicada em algoritmos de *deep learning* que procura identificar e detetar objetos em imagens, tendo a capacidade de adaptação e aprendizagem

com base em dados pré-fornecidos. É uma versão mais especifica de uma *Deep Neural Network* (DNN), uma vez que é projetada especialmente para análise de imagens e vídeos. O seu funcionamento baseia-se numa hierarquia bem definida, passando de níveis de complexidade mais baixos para níveis mais altos. Geralmente possuem 3 camadas, podendo variar de acordo com o tipo de arquitetura a ser utilizada, sendo as duas primeiras responsáveis por extrair dados e recursos (*Convolution e Pooling*) e a última (*Fully-Connected*) por conectar todos os dados provenientes das camadas e operações anteriores a uma determinada classificação, tal como é observado na Figura 1 [11, 12].

O processo de *convolution* é o primeiro passo dentro do funcionamento de uma CNN. Atua através da aplicação de um *kernel*, uma matriz com pesos (valores) associados, que realiza operações de multiplicação sobre todos os pontos de uma imagem, de forma a minimizar a complexidade da mesma. Este processo procura facilitar a análise dos dados por parte da máquina, de forma a aumentar a precisão e eficiência, extraindo os dados que são considerados relevantes. *Pooling* é a camada que sucede *convolution*, onde é efetuada uma redução da dimensão dos pontos provenientes da camada anterior. Esta etapa pode ser realizada de duas formas: *max pooling* ou *average pooling*. No é tido em conta o valor máximo dentro de um grupo, enquanto o *average pooling* efetua a média dos valores de um grupo, tornando-se imprescindível saber quando se deve aplicar cada uma destas abordagens. O processo de *convolution* é o primeiro passo dentro do funcionamento de uma CNN. Atua através da aplicação de um *kernel*, uma matriz com pesos (valores) associados, que realiza operações de multiplicação sobre todos os pontos de uma imagem, de forma a minimizar a complexidade da mesma. Este processo procura facilitar a análise dos dados por parte da máquina, de forma a aumentar a precisão e eficiência, extraindo os dados que são considerados relevantes. *Pooling* é a camada que sucede *convolution*, onde é efetuada uma redução da dimensão dos pontos provenientes da camada anterior. Esta etapa pode ser realizada de duas formas: *max pooling* ou *average pooling*. No *max pooling* é tido em conta o valor máximo dentro de um grupo, enquanto o *average pooling* efetua a média dos valores de um grupo, tornando-se imprescindível saber quando se deve aplicar cada uma destas abordagens.

Geralmente todos os *kernels* são associados a um filtro, trabalhando em simultâneo de forma a atingirem o mesmo propósito e objetivo de análise, através da realização das suas tarefas de forma coletiva. Antes da última camada, a camada *fully-connected*, existe ainda uma operação de *falttening*, que consiste em transformar a matriz de dados proveniente das camadas antecessoras a esta operação num vetor uni-dimensional, de forma a facilitar o uso desses dados [13]. A camada *fully-connected* é a responsável por receber todos as saídas das camadas anteriores, unificá-las num único neurónio da rede, executando operações de conversão de dados, tal como *sigmog* ou *softmax*, de forma a fornecer a classificação final, através de um conjunto de probabilidades [14, 15, 16].

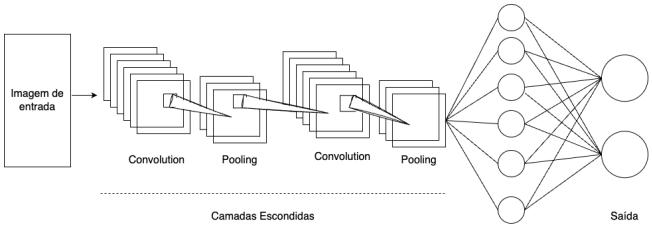

Figure 1. Arquitetura básica de uma CNN. Fonte: Adaptado de [12].

Devido à sua enorme capacidade de processamento e aplicabilidade no processo de classificação de objetos, foram desenvolvidas diferentes variações, podendo haver uma classificação entre arquiteturas *one-stage* e *two-stage*. Nos algoritmos *one-stage*, o processo de deteção de objetos é feito em uma única etapa, utilizando uma única rede convolucional para localizar e classificar o objeto. O objeto é identificado através de uma caixa de contorno (*bounding box*), sendo todo o processamento realizado na mesma etapa [17]. Esta característica torna estes algoritmos mais rápidos devido à sua rápida implementação, mas consequentemente menos precisos na classificação, tendo em conta que não é feita uma análise minuciosa dos dados [18, 19]. Alguns exemplos de algoritmos *one-stage* são *You Only Look Once* (YOLO), *Fully Convolutional Network* (FCN) e *CornerNet512* [20].

Os algoritmos *two-stage*, por outro lado, dividem o processo de localização e classificação em duas etapas distintas, efetuando diversas análises de uma mesma imagem, o que pode sacrificar o desempenho, podendo no entanto garantir uma precisão aumentada [21]. Na primeira etapa é efetuada a análise da imagem, onde são detetados e atribuídos diferentes pontos de interesse, indicando a existência de um objeto naquele determinado local. Realizada esta primeira etapa, é utilizado o seu resultado para fazer a classificação do objeto e identificá-lo usando uma caixa de contorno [17, 21, 22]. Alguns exemplos dos algoritmos *two-stage* são *Faster R-CNN*, *Feature Pyramid Network* (FPN) e *Region Based CNN* (R-CNN) [20].

2.1.1 *One-stage Algorithms*

A *Fully Convolutional Network* (FCN) ou Rede Totalmente Convolucional, é uma rede neuronal que não faz uso de camadas totalmente ligadas, no entanto apresenta-as como camadas convolucionais cujo campos recetores cobrem todo o mapa de características subjacentes [23]. Para tal, baseia-se em uma CNN convencional, devido à sua grande capacidade de análise, porém apresenta algumas alterações que fazem com que seja bastante diferente destas.

Todo o processo que é feito normalmente pelas *hidden layers* de uma CNN, é realizado através de camadas ligadas entre si localmente, tornando as camadas independentes e sem a necessidade de usar camadas densas, o que cria um processo de aprendizagem muito mais eficiente e rápido [24].

Este processo é representado na Figura 2, onde é possível verificar que cada etapa realizada pelo algoritmo trabalha em

conjunto com as etapas posteriores. Relativamente à saída, a FCN diferencia-se das CNN uma vez que a primeira permite realizar previsões ao nível de todos os pontos da imagem enquanto numa CNN é dado o resultado com base em probabilidades de correspondência [25, 26]. A FCN apresenta bons resultados na segmentação semântica ou segmentação de imagem, que consiste na tarefa de agrupar partes de uma imagem que pertencem à mesma classe de objetos [27].

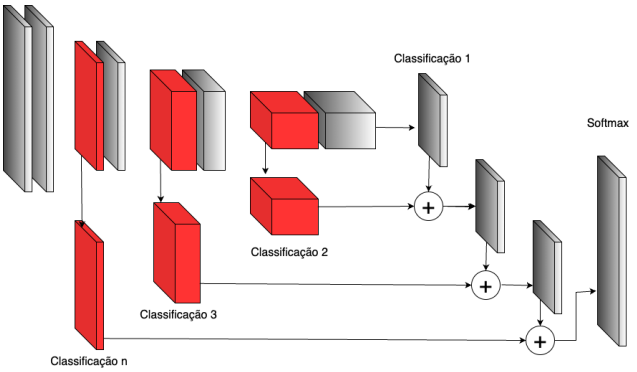

Figure 2. Exemplo de arquitetura de uma FCN. Fonte: Adaptado de [28].

O algoritmo YOLO (*You Only Look Once*) foi proposto por Joseph Redmon e Ali Farhadi [29]. É uma *deep convolutional neural network* sendo um dos principais algoritmos de *machine learning* para imagens usados na atualidade. É usado para realizar a deteção de objetos, conseguindo definir e isolar os objetos alvos em vídeos, transmissões em direto ou em imagens [30].

Inicialmente são realizadas diferentes operações de convolução para simplificar os dados recebidos. Após este primeiro passo é aplicada uma máscara em forma de grelha sobre estes dados, permitindo a divisão em diferentes secções. Posteriormente é realizada uma análise simultânea de todas as secções existentes, sendo verificado se existe algum objeto de interesse em cada uma destas secções. Caso exista, é aplicada uma *bounding box* em torno desse objeto. Para além disso, é ainda fornecido a probabilidade de classificação dos objetos de interesse, ou seja, um valor que indica o quão provável o resultado fornecido ser equivalente a um objeto de interesse [31].

A arquitetura do algoritmo é ilustrada na Figura 3. Todos estes passos são realizados em apenas uma única interação, onde o algoritmo percorre toda a imagem uma única vez, permitindo que tenha um maior desempenho e rapidez durante o processo de identificação e classificação, correndo o risco de ser menos preciso em determinados cenários. Para além disso, o YOLO ainda permite usar outros algoritmos ou modelos de visualização de objetos, o que aumenta ainda mais a sua versatilidade e capacidade de implementação para diferentes áreas.

DD-Net (*Double-Feature Double-Motion Network*) [33] é uma arquitetura CNN cujo objetivo é tornar o processo de recolha e processamento de dados de um esqueleto humano

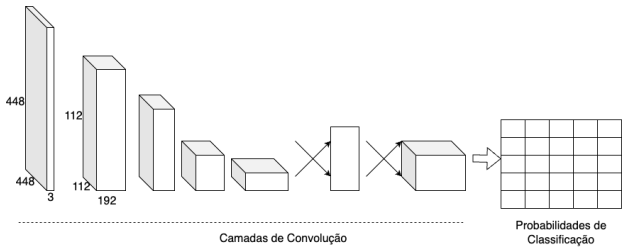

Figure 3. Arquitetura YOLO. Fonte: Adaptado de [32].

mais rápido, fluido e eficaz. Esta arquitetura tem em conta 2 tipos de dados de entrada: os dados geométricos e os dados de coordenadas Cartesianas.

Por funcionarem de maneira diferente, esta arquitetura tem a capacidade de analisar estes 2 tipos de forma separada, mas apenas numa única interação.

O algoritmo começa por aplicar um mecanismo de cálculo entre as distâncias das articulações do corpo humano, de forma a conseguir uma matriz de simetria. Com esta etapa realizada, é aplicado um sistema de calculo temporal, verificando as diferenças que existem entre os diferentes quadros de um vídeo, independentemente da velocidade do movimento, adequando a implementação a velocidade através de duas abordagens - *fast global motion* e *slow global motion*. O processo de associação dos diferentes elementos, os 2 métodos de escala de movimento e as articulações do corpo, é realizado através de um sistema de incorporação que utiliza 3 níveis de CNN, tal como se pode verificar na Figura 4.

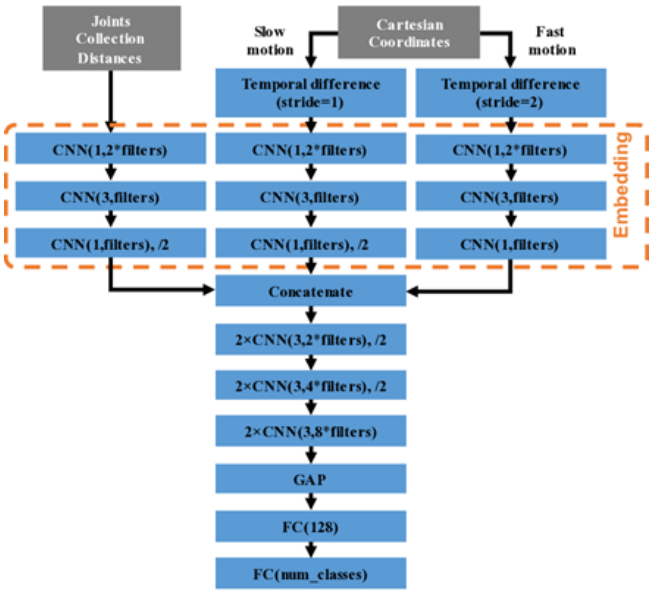

Figure 4. Arquitetura DD-Net. Fonte: [23].

2.1.2 *Two-stage Algorithms*

O *OpenPose* é um sistema de deteção de poses humanas de uma ou múltiplas pessoas em tempo real, desenvolvido por Zhe Cao et.al [34]. Foi o primeiro sistema a conseguir detetar em simultâneo todos os pontos críticos no corpo humano

(mãos, cara, tronco e pés), totalizando 135 pontos-chave. Este método ganhou o desafio *COCO 2016 Keypoints Challenge* e é popular pela sua qualidade e robustez em ambientes com mais do que uma pessoa.

Tal como demonstrado na Figura 5, este sistema começa por receber uma imagem RGB que é analisada por uma CNN de duas fases. A primeira fase prevê a localização de até 18 *confidense maps*, equivalente às 18 partes do corpo humano. *Confidense map* é uma representação do local mais provável de existir uma articulação em uma determinada localização da imagem, sendo representado através de *heatpoints*. O ramo seguinte prevê outro conjunto de 38 campos de afinidade de partes (PAFs) que denota o nível de associação entre partes. Esta segunda fase, recebe os dados provenientes da primeira, bem como a imagem original, tornando possível identificar a que corpo pertencem os diferentes membros detetados pela etapa anterior. Isso possibilita que haja uma diferenciação de todas as pessoas da imagem. O PAF gera um conjunto de vetores entre os diferentes membros, representando a distância, direção e a relação entre eles. Em seguida é feita uma limpeza dos dados gerados até esta fase de forma a tornar mais simples os processos seguintes. A associação dos membros é feita através de gráficos bipartidos, que representam todas as relações provenientes da segunda fase, onde as ligações mais fracas são consideradas inválidas e eliminadas destes gráficos, podendo aplicar o esqueleto de pose sobre a imagem [34, 35, 36, 37, 38].

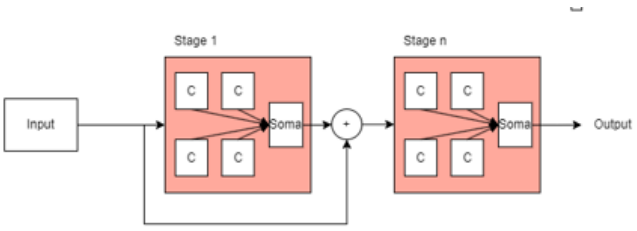

Figure 5. Arquitetura *OpenPose*. Fonte: Adaptado de [35].

O I3D também conhecido como *Two-Stream Inflated 3D ConvNet* é um tipo de arquitetura concebida para processar dados 3D, tais como vídeos ou dados volumétricos. Esta arquitetura baseia-se em redes neuronais convolucionais 2D, que são utilizadas para a realização de classificação de imagem.

O algoritmo começa por fazer uso de pesos CNN 2D pré-treinados (parâmetros utilizados para definir os filtros que são aplicados ao *input*, a fim de extrair características) que são, por sua vez, transformados em 3D. Esta transformação é a reprodução dos filtros 2D ao longo da terceira dimensão. Para além disso, este método ainda faz uso de um sistema de dois fluxos separados de entrada de dados, sendo cada um processado por uma rede I3D separada. Um fluxo consiste essencialmente emquadros cujos pontos têm informação de cor (*Red, Green, Blue*), enquanto o outro fluxo consiste em mapas de fluxo ópticos que codificam o movimento de pontos entre quadros. Esta expansão permite que sejam aprendidas as características espaço-temporais diretamente a partir de vídeos [39]. Na Figura 6 é representada a estrutura básica de uma I3D, onde é possível identificar dois fluxos de dados, imagem RGB e *Optical Flow*, que após serem tratados por duas redes diferenciadas, unificam-se para gerar a saída final.

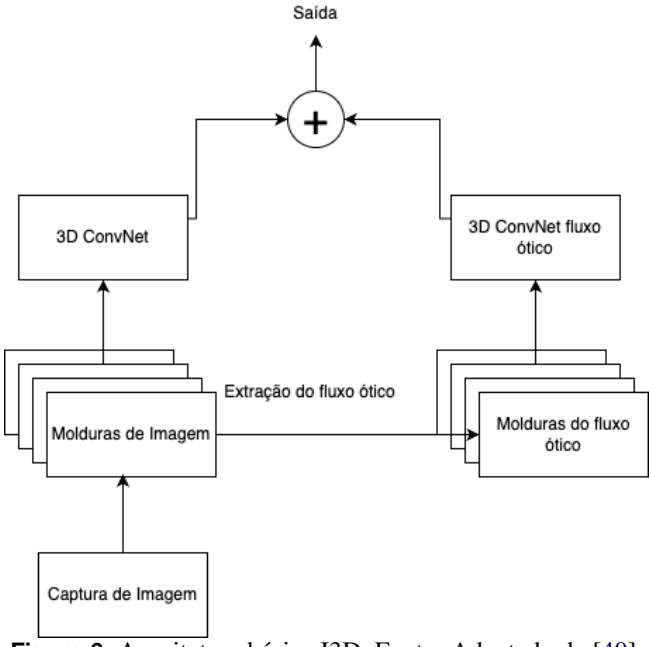

Figure 6. Arquitetura básica I3D. Fonte: Adaptado de [40].

2.2 Outras Arquiteturas

SVM (*Support Vector Machines*) é um modelo utilizado para classificação e regressão que se tornou uma estratégia comum para o reconhecimento de padrões visuais. O objetivo é encontrar um hiperplano (*hyperplane*) num espaço N-Dimensional onde o N representa o número de características que são distintamente classificadas nos pontos de dados. Um *hyperplane* é um limite de decisão que ajuda a classificar os pontos de dados. Para separar duas classes de pontos de dados existem muitos *hyperplanes* que podem ser escolhidos, o objetivo é encontrar e escolher um que tenha a distância máxima entre os pontos dos dados de ambas as classes.

Na Figura 7 são exibidos dois gráficos com um exemplo de uma grande variedade de possíveis hiperplanos e um exemplo de um hiperplano com uma distância máxima entre os pontos. O facto de existir uma maximização da distância entre os pontos, proporcionará uma maior confiança na classificação de futuros pontos de dados. Os hiperplanos surgem a partir dos vetores de suporte (pontos de dados) que estão mais próximos do hiperplano que tem um impacto quanto à posição e orientação do mesmo. Com a utilização destes vetores de suporte é possível aumentar a margem do classificador [41, 42].

A *framework MediaPipe* foi desenvolvida pela equipa de investigação da *Google* [44], sendo o seu principal objetivo permitir a criação de aplicações para análise de dados multimédia como processamento de vídeo, áudio e outros dados em tempo real, de forma fácil para os programadores. Baseia-se numa arquitetura de *pipelines*, que dá a capacidade de gerir

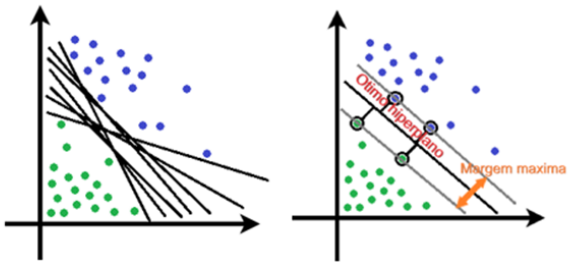

Figure 7. Exemplificação dos *hyperplanes*. Fonte: Adaptado de [43].

a ordem de execução e as principais características do sistema, preparando-o para o processo de receção de dados, através de modelos de *machine learning*.

Existem diferentes componentes nesta arquitetura, nomeadamente *packets*, *streams*, *calculators* e *graphs*. Um *packet* é a unidade básica de dados que passam por cada *pipeline*. Cada *pipeline* realiza operações de processamento sobre um *packet* de entrada e devolve um *packet* de saída, permitindo fluidez na transmissão das informações. Uma *stream* é um fluxo de dados formado por vários *packets*, com um determinado tipo específico, permitindo conectar os diferentes componentes dentro do sistema. Um *calculator* é um nó de cada grafo do sistema e representa um determinado componente, sendo neste componente realizada as diferentes operações do sistema.

Todas as ações de processamento funcionam com base em grafos, que é a representação gráfica e estrutural de todos os *pipelines* do processamento, mostrando todas as ligações e a organização geral do sistema [45, 46, 44]. Tal como é possível observar na Figura 8, o sistema recebe uma imagem de entrada que passa pelo transformador, onde é aplicado o modelo de *machine learning* escolhido, de forma a ser posteriormente usado em conjunto com as imagens originais em um processador, conseguindo desta forma gerar a saída final.

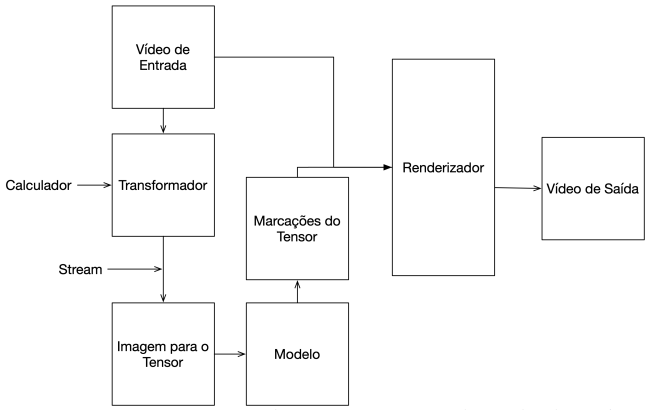

Figure 8. Estrutura *MediaPipe*. Fonte: Adaptado de [45].

MobileNet é um tipo de arquitetura que procura fornecer a maior eficácia em dispositivos com capacidade computacional mais limitada, como por exemplo telemóveis inteligentes. Criado pela *Google*, funciona através da implementação de duas camadas principais que diferenciam este algoritmo dos restantes.

De acordo com a Figura 9, a primeira camada é designada *depth separable convolution*, sendo dividida em *depth-wise convolution* e *point-wise convolution*. A *depth-wise convolution* é usada para aplicar um filtro único em cada entrada que a arquitetura recebe, permitindo um aumento da sua eficiência. Após isso é aplicado o *point-wise convolution* que é o responsável por combinar todos os filtros numa única unidade, de forma a esta poder ser usada para outras operações posteriores, através de operações de convolução.

O *MobileNet* necessita de 2 parâmetros principais para poder funcionar. O primeiro é um multiplicador de comprimento que é aplicado tanto nos filtros de entrada como de saída, que permite reduzir as dimensões e complexidade da rede. O segundo é um multiplicador de resolução que é aplicado ao mapa de entrada com o objetivo de reduzir o poder computacional necessário para executar o algoritmo [47, 48, 49].

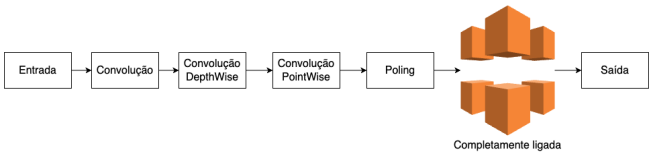

Figure 9. Arquitetura *MobileNet*. Fonte: Adaptado de [50].

PoseNet é um sistema de estimação da pose do corpo humano (*pose estimation*), desenvolvido pela *Google Creative Lab* [51]. Treinado com *MobileNet*, é utilizado em deteção corporal 2D e apresenta pesos já previamente estabelecidos.

O PoseNet recebe uma imagem que é sujeita a um processo de pré-processamento para ser compatível com a entrada da CNN. Após isso é feita a deteção das características do corpo humano, através da CNN, de forma a permitir extrair os pontos-chave/articulações principais do corpo. Ao todo são extraídos 17 pontos-chave, sendo aplicado um processo de limpeza e refinamento das características. Desta forma reduz-se a ocorrência de problemas como interferência ou ruído, resultando num diagrama de esqueleto *skeleton* do corpo humano, com a representação dos pontos-chave e sua respetiva localização.

Este processo todo ocorre, tal como é mostrada na Figura 10, através de um codificador, que realiza todo o processamento e o localizador para identificar os pontos de interesse [51, 52, 53, 54].

O algoritmo ICP (*Iterative Closest Point*) tem como principal objetivo registar dois conjuntos de pontos, reduzindo a distância entre os pontos correspondentes nos dois conjuntos [56] por P. Besl e N-D. McKay. Este algoritmo pode ser vantajoso na realização de tarefas de alinhamento de modelos 3D ou de nuvens de pontos obtidas através de digitalização 3D.

O algoritmo funciona iterativamente, encontrando a correspondência mais próxima entre os pontos dos dois conjuntos

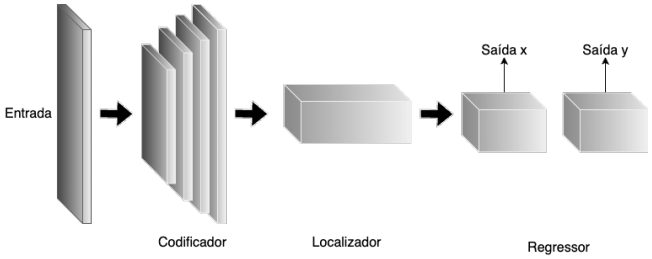

Figure 10. Estrutura básica do *PoseNet*. Fonte: Adaptado de [55].

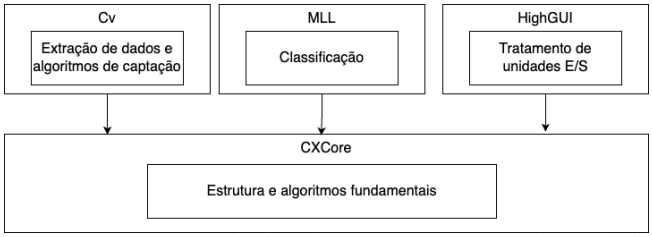

Figure 11. Estrutura básica do *OpenCv*. Fonte: Adaptado de [61].

de dados e em seguida é calculada a rotação e translação que minimizam a distância entre esses pontos. Após isso, o segundo conjunto é transformado usando essa rotação e translação, repetindo o processo de encontrar a correspondência mais próxima e calcular a rotação e translação até que a solução se estabilize [57].

Várias variações do algoritmo foram propostas ao longo dos anos, incluindo versões robustas, distribuídas e paralelas, para lidar com diferentes tipos de dados e casos específicos. É importante mencionar também que o algoritmo pode ser afetado por ruído e dados fora da gama esperada (*outliers*) s e pela qualidade da correspondência inicial.

OpenCV é uma biblioteca *open-source* da área de visão computacional, *machine learning* e processamento de imagem, desenvolvida pela Intel [58, 59, 60]. Esta tecnologia apresenta a capacidade de captação de imagens e vídeos, de forma a ser possível detetar objetos ou pessoas. A sua arquitetura assenta sobre uma estrutura modular, ou seja, é formado por pacote, cada um com uma biblioteca própria que pode ser partilhada ou privada.

A Figura 11 apresenta a arquitectura deste algoritmo, onde se destacam 3 componentes principais. O núcleo fornece estruturas de dados, como matrizes, para armazenar os diferentes tipos de dados. O módulo de processamento de imagem, também conhecido como *imgproc*, permite realizar as diversas operações de tratamento de imagens, como segmentação, transformação ou deteção de bordas. Fornece ainda um módulo específico para processamento de vídeo, com funcionalidades próprias para detetar, gravar e reproduzir vídeos.

Para além desses módulos principais, são fornecidos outros, destacando-se o módulo de *Features2D*, para extração de características e pontos críticos em imagens e um módulo de *machine learning*, que permite implementar diversos algoritmos, como CNNs com processos de classificação, regressão agrupamento e deteção de objetos. Para complementar todo o sistema, existe ainda um módulo específico, designado *HighGUI* que é responsável por tratar e coordenar todos os dispositivos externos do sistema, bem como todas as ações geradas pelos mesmos [58, 59, 60].

TensorFlow é uma *framework open-source*, desenvolvida pela *Google* [62]. Funciona com base em grafos que representam as conexões entre todos os componentes dentro do sistema. Existem ainda dois conceitos básicos neste modelo, o *tensor* e o *flow*.

O *tensor* é uma matriz multidimensional que armazena um determinado tipo de dados, não havendo mistura entre tipos em cada *tensor*. O *flow* é todo o processo que os dados percorrem, desde a sua entrada até a sua saída, envolvendo todos os tipos de operações e funcionalidades. No *flow* é realizado o tratamento de dados, é definida a função de perda e o modelo de aprendizagem, e é efetuada a escolha de um otimizador. É também realizado um ciclo de treino com propagação direta para gerar previsões, cálculo do valor de perda, propagação inversa para comparar os dados reais com os perdidos e uma atualização constante do modelo. Para finalizar é realizada a avaliação geral do modelo [63, 64].

3. Trabalhos Relacionados

Esta secção enumera e descreve os diferentes trabalhos resultantes da pesquisa efetuada. Os trabalhos inserem-se na área do tema proposto e fazem uso de pelo menos uma das tecnologias referenciadas na secção anterior. Em cada uma das descrições é apresentado de forma pormenorizada o processo de implementação de cada sistema.

3.1 Sistema de Reconhecimento em Tempo Real

Em [65] foi desenvolvido um sistema de reconhecimento de ação em tempo real que se destina principalmente à deteção de comportamentos perigosos em "*Empty Nesters*" e identificação de violência em recintos fechados. A estrutura base do sistema é formada por duas etapas principais: a deteção da pessoa e a classificação da ação como sendo perigosa ou não, tal como demonstra a Figura 12.

Inicialmente foram realizados treinos *offline* sobre o *dataset NTURGB+D*, devido à dificuldade de criar um grande conjunto de dados em vídeo. Foram utilizados 36880 vídeos para treino e 16000 vídeos para teste. Para isso, foi utilizada uma câmara Realsense D435 para captura de imagem. Através da câmara são recolhidos dois fluxos de dados, nomeadamente o de RGB e o de profundidade, no entanto só foi utilizado o fluxo RGB. Esta escolha deve-se há possibilidade de que a fusão dos dois fluxos possa reduzir a velocidade do reconhecimento.

A fase seguinte do sistema é a deteção humana, onde foi utilizado o *YOLONet3*. Devido à necessidade de um rápido desempenho, esta foi alterada para descartar as classes de objetos que não sejam pessoas. Após isso foi realizada uma filtragem das imagens existentes no *dataset MSCOCO*, de

forma a selecionar apenas as que continham pessoas. No processo de deteção em tempo real, se houver alguém na imagem, o detetor pode obter a posição da pessoa sob a forma de uma caixa. A forma mais fácil de trabalhar com estes dados é utilizar a informação dentro da caixa, podendo no entanto serem perdidos dados importantes. Para evitar isso, os autores estabeleceram um fator de ampliação para aumentar o tamanho da caixa numa certa percentagem e para isolar a área com base na região de interesse (RoI - *Region of Interest*) correspondente. Foram testados diferentes fatores de ampliação e os resultados mostraram que quanto maior era o fator de ampliação, pior era a precisão. Estas informações apontam para uma relação entre a capacidade de remoção dos elementos do fundo e o efeito de reconhecimento da rede, melhorando à medida que se consegue remover mais elementos do fundo.

Para realizar o reconhecimento da ação foi utilizado o algoritmo I3D (*Two-Stream Inflated 3D ConvNet*). A utilização desta rede é justificada pelo reconhecimento da ação neste estudo ser baseada em vídeo. Tendo isto em conta, existe a necessidade de aprender não só as características espaciais, como também as características temporais, fazendo o núcleo de convolução I3D ser expandido de 2D para 3D. Foi ainda definida uma "*flag*" que sinaliza se o sistema se encontra ou não em processo de reconhecimento. Se a *flag* for *true*, significa que o sistema está em processo de reconhecimento e o detetor irá processar cada imagem lida pela câmara. Caso contrário o detetor de pessoas irá processar a imagem em determinados intervalos, o que pode reduzir custos computacionais. O sistema toma decisões diferentes tendo em conta os resultados dos testes.

Perante a dificuldade de tratar vídeos com uma duração de ações variada, os autores utilizaram um modelo de amostragem aleatória para garantir a obtenção de informações suficientes sobre a ação e melhorar a capacidade de reconhecimento da rede, mesmo quando a duração exata da ação é desconhecida. Para comprovar a afirmação anterior os autores testaram um modelo de amostragem aleatória que obteve um resultado médio de 81.25% e outro uniforme que obteve 78.5%, estes testes foram realizados com o uso de uma GPU Nvidia 1080Ti.

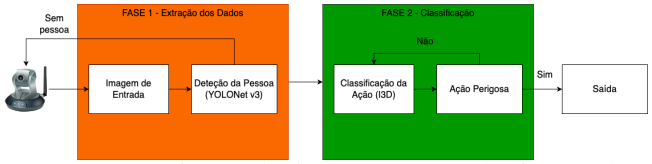

Figure 12. Arquitetura de sistema para reconhecimento de comportamento perigoso. Fonte: Adaptado de [65].

3.2 Captura do movimento humano

Em [66] é proposto um método para capturar o movimento humano de forma rápida e precisa a partir de uma única câmara. É utilizada uma combinação de técnicas de *machine learning* e modelos 3D para alcançar essa finalidade. O processo inicial de captação dos dados é realizado através de um sensor RGB-D, com capacidade de captura de imagens (RGB) e propriedades de profundidade. Neste estudo, a captura ocorre com base em duas etapas diferentes, sendo elas a captura de movimento híbrido semântico (SHMC) e a combinação de movimento semântico bidirecional (SBMB).

A captura de movimento híbrido semântico permite captar imagens e dados em tempo real, sendo para isso usado um modelo baseado em ICP e captura em 3D. A principal função do ICP é rastrear as posições ao longo do tempo, fazendo uma comparação entre os quadros de uma determinada sequência de ações e com isso detetar a ocorrência de erros, de forma a ser possível efetuar o tratamento. Na ocorrência de um erro o sistema sofre uma reinicialização, que permite aumentar a eficiência e o desempenho.

Para detetar e tratar erros são implementados um inspetor (*inspector*), um detetor (*detector*) e um rastreador (*tracker*). O inspetor é responsável por verificar e calcular a perda de dados semânticos, após o surgimento de algum problema durante a etapa de captação corporal. Para além disso, o inspetor tem ainda a função de criar uma máscara semântica de forma a detetar e lidar com estes mesmos erros. Os autores optaram por usar um modelo *skeleton* do tipo SMPL (*Skinned Multi-Person Linear Model*), o qual foi dividido em 4 partes principais (braços e pernas). Ao realizarem este tipo de divisão, é pretendido reduzir a ocorrência de erros associados a este tipo de membros, visto que são mais estreitos e realizam movimentos mais rápidos. Esta aplicação, juntamente com o uso do inspetor, permite que a correção de erros e falhas possa ocorrer apenas no membro do corpo que apresentou uma falha. Isto permite tratar esse membro de forma individual, sem haver o risco de comprometer o processamento dos restantes.

Na implementação do detetor, é recebido um *input* e fornecida uma reconstrução 3D do corpo humano. O detetor possui a capacidade de fazer uso das informações semânticas das perdas em cada uma das etapas do processo de deteção, de forma a permitir um aumento ainda maior da eficiência. O rastreador foi baseado num sistema de captura de movimento humano em tempo real que utiliza um rastreio de superfície não rígida e uma fusão geométrica (TSDF - *Truncated-Signed-Distance-Field-Based Fusion*). Este método permite resolver problemas de reconstrução de um modelo geométrico antes de ser realizada a captura de movimento, o que consome muito tempo.

O sistema é ainda capaz de criar uma superfície geométrica detalhada e completa após processar a informação de profundidade. Ao realizar a combinação destes 3 componentes foram obtidos resultados positivos, porém com bastante instabilidade e ocultação de membros. Para melhorar estes aspetos foi incorporado um tempo de deteção adicional na função de rastreio, o que aumenta a necessidade computacional. Este problema de desempenho em tempo real, foi solucionado com a utilização de um solucionador *Gauss-Newton* baseado na GPU.

Através da combinação de movimento semântico bidire-

cional é possível melhorar a precisão e a robustez do movimento, bem como resolver problemas de ocultação de membros. Para tal, esta etapa é dividida em duas fases: rastreamento reverso (*Inverse Tracking*) e combinação de movimento semântico bidirecional (*Semantic Bidirectional Motion Blending*). A primeira etapa permite ao sistema agir quando ocorre o reaparecimento de um membro, que se encontrava oculto até ao momento. É usado um rastreio total da sequência, de modo inverso, percorrendo todos os quadros da mesma. Isto previne a ocorrência de erros durante uma ocultação repentina, suavizando este fenómeno. Todo este processo é realizado usando o algoritmo ICP de forma a alinhar as posições ao longo da sequência. A segunda etapa cria uma combinação dos movimentos obtidos dos dados 2D e dos dados 3D que são gerados nas etapas anteriores, de forma a melhorar a geração final dos dados semânticos. Na fase de testes, verificou-se uma diminuição na ocorrência de erros ao realizar o *tracking* e uma melhor recuperação quando há ocorrência de erros, permitindo bons resultados mesmo perante ocultação de membros ou movimentos mais rápidos.

3.3 Modelo para Deteção e Reconhecimento de Ações Humanas

No trabalho [67] é proposto um novo modelo para deteção e reconhecimento de ações humanas, fazendo uso de YOLO e *OpenPose*. Os autores dividem o processo em 3 etapas diferentes, tal como ilustrado na Figura 13. Primeiro é realizado o reconhecimento da pessoa, em uma imagem, seguido por deteção e delimitação das articulações (*joints*).

A deteção é realizada através da implementação do YOLOv4, onde foi delimitada a busca de apenas seres humanos, excluindo todos os restantes, aumentando a sua precisão média (mAP). Com o alvo delimitado, os autores decidiram usar o *OpenPose* para extrair os pontos-chave do corpo humano. Foram realizados testes em 3 tipos de métodos (*OpenPose based methods*), tendo obtido os melhores resultados com *OpenPose-COCO*. Este método alcançou 1052 sucessos em 2000 imagens usadas durante os testes, usando o *skeleton* do *COCO dataset*, que fornece 18 pontos críticos.

Para resolver o problema de diferença de idades e tamanhos que podem surgir durante o processo de deteção, é usada uma forma normalizada. Esta metodologia fornece a capacidade de adaptação e aprendizagem ao modelo. Para o processo de classificação foi utilizada uma CNN. Esta rede é formada por uma *input layer*, uma *hidden layer* e uma *output layer*. Tem uma dimensão de 36, com 2 *outputs*, para cada uma das classificações que estavam a ser tidas em conta. Foi apenas usada uma *hidden layer*, uma vez que o *dataset* utilizado pelos autores era de dimensões reduzidas, evitando *overfitting*. O *dataset* utilizado para treinar o modelo foi o *CCTV dataset*, desenvolvido pela KISA (*Korea Institute Security Agency*). Os testes foram realizados num sistema com CPU Intel(R) Xeon(R) Silver 4215R, uma gráfica NVIDIA Quadro GTX5000, memória RAM DDR4 64GB e a versão 11.2 do CUDA. Foi alcançado um valor de precisão média de 91%.

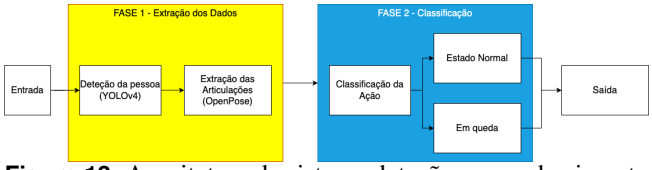

Figure 13. Arquitetura de sistema deteção e reconhecimento de ações humanas. Fonte: Adaptado de [67].

3.4 Auxilio à Realização de Exercícios Físicos

Os autores de [68] propõem a criação de uma aplicação para poder auxiliar a realização de exercícios físicos aos utilizadores, sem a necessidade de um especialista. Para isso, foi utilizado a ferramenta pública da *Google*, *MediaPipe* para extrair o esqueleto de uma pessoa em uma imagem RGB. Esta ferramenta permite fazer a segmentação do fundo e faz uso de 33 marcações no modelo do esqueleto, atribuindo em seguida uma classificação da ação.

O sistema recolhe os dados do esqueleto da pessoa, provenientes do *MediaPipe*, armazenando esta informação sob a forma de um *array*, quadro a quadro, até alcançar uma dimensão máxima de 60 quadros por *array*. Com os dados totalmente tratados, é aplicada uma DD-Net (*Double-Feature Double-Motion network*), de forma a permitir identificar que ação está a ser executada. Esta rede foi treinada com um *dataset* desenvolvido pelos próprios autores, que se foca em 3 exercícios, com imagens de pessoas que variam entre os 15 e 55 anos de idade. O resultado desse treino é carregado em um serviço da *Google Cloud*. O modelo repete este processo novamente até não existirem mais dados a serem processados, evitando o acumular de tarefas, o que otimiza o tempo de execução.

Finalizando o processo de deteção, são apresentados 2 resultados diferentes, baseado em quadros e baseado em sequência. O primeiro é tratado usando a informação fornecida pelo *MediaPipe*, de forma a calcular dados referentes aos ângulos entre as articulações dos membros do corpo. Isso permite a recolha de dados desse quadro, sem haver qualquer tipo de conexão com os restantes. Para ser possível realizar esta conexão e permitir identificar os exercícios e possíveis erros, é usado o método DTW. Com este método, fazendo uso da localização das articulações e da classificação do exercício, fornecidos pelas etapas anteriores é possível comparar as diferenças temporais entre os dados. Estas diferenças fornecem um valor de desvio entre as sequências, possibilitando avaliar a qualidade do exercício, alcançando uma precisão média de 98,33%. Para atingir estes valores, os testes foram implementados na linguagem *phyton* com uso da *framework TensorFlow*, em um servidor com um Xeon E5-2420 CPU, 16GB de RAM e uma GPU NVIDIA GTX 1080ti, sendo recolhidas amostras de 9 pessoas a realizarem os diferentes exercícios.

3.5 Reconhecimento de Ações de Reabilitação em Tempo Real

Em [69] é apresentada uma abordagem de reconhecimento de ações de reabilitação em tempo real utilizando *OpenPose* e FCN. Os autores começaram por utilizar um método de *pose estimation* 2D baseado em *OpenPose*.

É combinado com algoritmo baseado num filtro *Kalman* para seguir o alvo principal no vídeo e gerar as sequências de dados. O esqueleto extraído no fluxo de vídeo continua a ser um resultado de deteção independente. Devido a isso, o corpo alvo perde a relação temporal com os restantes quadros, perante a presença de vários utilizadores. Caso o corpo alvo não seja detetado durante 5 quadros consecutivos, a sequência de ação alvo é eliminada e o estado inicial do filtro é reposto. Na extração, são extraídas as características originais da sequência de ação através de uma janela deslizante.

Em cada quadro o corpo tem 18 pontos-chave, com um total de 36 características. De acordo com a duração da ação de reabilitação, o tamanho da janela deslizante é definido para 80 quadros. Para melhorar a robustez do algoritmo são removidos os olhos e as orelhas, que são considerados pontos-chave inúteis. As características são posteriormente fornecidas a uma 1D *Full Convolutional Neural Network* (FCN) para a classificação da ação. A FCN é treinada por sequências de vídeo, usando um *dataset* fornecido pelo departamento de reabilitação do hospital da universidade de *Zhengzhou*.

O *dataset* é formado por 6 tipos de ações, com um total de 2075 vídeos com resoluções de 720p e 1080p, com diferentes tipos de variáveis, tais como nível de luminosidade, plano de fundo, ângulo e distância de captação do vídeo. Foram obtidos valores de precisão média de 85,6%, usando uma gráfica NVIDIA GTX1060, atingindo um valor de 18.14 quadros por segundo (QpS).

3.6 Sistema de Deteção de Poses Humanas

O trabalho apresentado em [70] propõe sistema para deteção de poses humanas, com o objetivo de verificar possíveis erros cometidos pelo utilizador durante a execução de exercícios físicos, sendo portanto um trabalho de especial interesse para este estado da arte. Os autores decidiram aumentar a eficiência dos algoritmos de *human pose estimation* ao utilizarem o *MobileNetv2*. Esta decisão foi tomada uma vez que se trata de um sistema para implementação *mobile*, sendo necessário um algoritmo que funcione bem em ambientes com capacidade computacional restrita.

O algoritmo começa por executar uma extração dos pontos-chave do corpo humano, através de *heatmarkers*. Estes *heatmarkers* são construídos com base em 3 operações de desconvolução. Nesta etapa é ainda aplicado um método de *Knowledge Distilation* (KD) [71], com o objetivo de aumentar ao máximo a eficiência do processo. Esta fase do processo é fundamental, pois permite transferir o conhecimento adquirido por uma rede complexa, para uma rede mais simples e compacta. Para além disso, é usado um mecanismo de propagação de *bounding box*, que permite realizar um ajuste da localização exata das articulações com base na previsão da próxima posição.

O processo de avaliação do movimento é dividido em 3 etapas. Em primeiro lugar é efetuada uma avaliação de pontos críticos do movimento, com base num *dataset* com a definição dos principais movimentos característicos de um exercício. Esta decisão permite fazer uma rápida avaliação do movimento apenas com base nessa característica temporal. A segunda etapa consiste na comparação da *KeyPose* extraída pelo processo anterior com a *KeyPose* correspondente a esse exercício no *dataset*, no mesmo instante temporal. Para atingir os resultados pretendidos são usados dois vetores, o original e o previsto, sendo depois medido o valor do ângulo. Estes processos são repetidos para todos os quadros do vídeo e é calculado a quantidade de vezes em que se verifica diferença no movimento. Caso exceda um determinado valor limiar é indicado como sendo um movimento incorreto.

Foi obtida uma precisão de 97.39% em 5 movimentos com repetições de 5 sequências. Foi usado o *COCO dataset* para treinar o modelo e o *TensorFlow* para estimar a velocidade de execução em ambiente móvel com o sistema operativo Android.

3.7 Reconhecimento de Ações Humanas

Em [72] é proposto um método de reconhecimento de ações humanas com um modelo baseado em *PoseNet* usando um *Raspberry Pi* 4. A primeira etapa consiste na identificação das coordenadas de todas as articulações do corpo humano (pontos chave - *pontos-chave*), numa imagem. É usado o *PoseNet* para fazer esta extração, aplicando-o em todos os quadros, tanto de forma isolada como sequencialmente. Esta versatilidade permite aplicar o modelo em vídeos e reconhecimento em tempo-real. O modelo recebe uma imagem RGB, sendo analisada e codificada com uma CNN baseada na arquitetura *MobileNetV1*.

Os dados são descodificados usando um algoritmo de descodificação, fornecendo um *heatmap* do local mais provável de existir um ponto-chave. Finalizando este processo, é determinada a localização exata de cada ponto-chave específico através do valor do vetor de desvio presente no *heatmap*. A fase seguinte é responsável pelo reconhecimento da postura, sendo realizada através da comparação dos pontos-chave e os ângulos dos membros da pessoa. Estas informações indicam a localização e formato do esqueleto da pessoa, permitindo identificar que ação ou movimento está a ser executado.

Para testar o modelo, foram realizados testes com 10 tipos de posturas em 6 utilizadores diferentes. O ambiente de implementação utilizado foi um *Raspberry Pi* 4 com acelerador Coral TPU, uma ARM 1.5 GZ 64 bit quad-core CPU e 4 GB de RAM, sendo atingido um valor de 15 quadros por segundo. Foi alcançada uma média de precisão de 70%.

O artigo [5] propõe um algoritmo de reconhecimento de pose humana baseado em *OpenPose* usando uma gráfica NVIDIA GTX 1060 com 6GB memória gráfica e um CPU

Intel Core i7 8750H.

A primeira fase consiste na realização de um aumento dos dados experimentais, através da geração de amostras de treino, com algumas diferenças entre elas. Essas diferenças foram obtidas através de uma série de transformações da imagem. O objetivo desta etapa é aumentar a dimensão do *dataset* e reduzir a dependência do modelo em relação a alguns atributos, melhorando a capacidade de generalização do modelo. Algumas das técnicas de transformação utilizadas foram a translação, a inversão, o corte, a rotação e a adição de ruído.

Depois desta fase é realizada uma seleção dos dados de entrada, onde são utilizados apenas 18 articulações humanas para uma possível melhoria da taxa de precisão. Em seguida os autores treinaram e testaram o modelo fazendo uso de amostras de imagens de um *dataset* de vídeos descarregado da Internet. O tamanho da imagem de entrada é 368x368, todas as amostras são treinadas de uma só vez, o número de ciclos é definido como 5, a taxa de aprendizagem inicial é definida como 0,0001 e são introduzidas 10 imagens por lote.

Estes testes revelaram que a velocidade de deteção do fluxo de vídeo da câmara de teste atingiu cerca de 20 molduras por segundo. Os resultados experimentais mostram que a precisão do reconhecimento do algoritmo adotado pelos autores atinge 91,5% com uma média de 9,7 QpS.

3.8 Sistema de Treinador de Ginásio

O trabalho [73] propõe um novo método de *pose estimation* para um sistema de treinador de ginásio usando IA. É constituído por 4 fases principais: a captação, o processamento, o treino dos *pipelines* e a avaliação.

O primeiro processo é realizado através de uma câmara de telemóvel ou computador, onde são captados os diferentes movimentos. O processo tem em consideração diversos parâmetros da captura, como variação na luminosidade, profundidade ou ângulos de incidência da câmara.

O processamento dos dados é a segunda fase, com a aplicação de técnicas de transformação, de modo a ajustar tamanho, direção e formato de cada imagem. A terceira etapa é o treino de um *pipeline* para identificação da ação, tendo sido usado um *pipeline* baseado em *MediaPipe*. É formado por uma rede neuronal, treinada com auxílio do *OpenCV*. Este tipo de metodologia permite identificar as articulações do corpo humano e o movimento que está a ser executado em um ambiente 3D. Para otimizar este processo é usado um algoritmo de retro propagação, que atualiza o *pipeline* com base no gradiente da função de perda, desta etapa. Esta função procura comparar e ver as principais diferenças entre o movimento atual e o movimento que é previsto ser executado.

A última etapa consiste na avaliação do *pipeline*, em um ambiente separado, de forma a verificar a sua precisão e robustez. Esta avaliação permite medir os valores de deteção das articulações (mAP) e para pose 3D (MPJPE). Foi obtida uma precisão média de 90%, usando um *dataset* público que não é apresentado pelos autores. Para além disso realizaram testes de comparação de QpS com outros métodos, como *YOLOv7*, *OpenPose*, *PoseNet* e *MoveNet*, tendo atingido resultados superiores.

3.9 Método de Classificação de Exercícios Físicos

Um método de classificação de exercícios físicos é proposto em [74], dividido em várias etapas, com diferentes funções dentro do sistema. Começando pela etapa de *pose estimation*, foi usado um método de *Lightweigth OpenPose*, que proporciona as capacidades habituais do *OpenPose*, porém com uma maior eficiência e menor exigência de poder computacional. Este algoritmo é alterado de forma a ser ainda mais eficiente em ambientes *mobile* ou pequenos dispositivos.

A arquitetura principal da rede usada foi alterada, implementando *MobileNetv1* e uma *dilated convolution*. Isto permitiu reduzir as quedas na precisão e redução da quantidade de camadas de refinamento [75]. Esta técnica consiste na alteração dos dados de entrada, através da adição de espaços em branco no *kernel* que é usado nesta etapa. Com isto torna-se possível ignorar certos pontos de uma imagem, melhorando a eficiência. Através destas mudanças foi atingida uma redução do tempo de execução em 15%, com uma queda de apenas 1% na precisão.

Foi ainda aplicado um método de CPU *acceleration*, denominado *OpenVIVO*, para otimizar ainda mais o modelo, uma vez que foi usada uma configuração sem aceleração gráfica para a realização dos testes. Em seguida é realizada uma conversão do modelo gerado para uma *Intermediate Representation* (IR), de forma a permitir uma maior compatibilidade com outros tipos de *hardware* mais simples.

O modelo foi treinado usando o *COCO dataset*, onde existem 18 pontos-chave do corpo humano. Foram escolhidos dois exercícios diferentes: *standing* e *squat*. O ambiente de treino é formado por um pequeno computador com Intel Core i5-8250U CPU quad core com 1.60Hz e 4GB RAM. Para captação das imagens é usada uma câmara com resolução 1280 e profundidade 720 a 90 QpS e resolução RGB de 1920/1080 e 30 QpS. Foi obtida uma precisão média de 96,8%.

3.10 Sistema Baseado em OpenPose

Em [76] foi usado um sistema baseado em *OpenPose*. Para tal, em vez de recolher dados de treino através de uma câmara, os mesmos são inseridos manualmente de forma a permitir um maior controlo. Os dados são formados essencialmente por vídeos com duração entre os 0,2 segundos e os 2 minutos, com um tamanho de 640/480 e 10 QpS. Os restantes dados são adquiridos usando uma câmara.

É aplicado o *OpenPose*, de forma a captar o esqueleto humano, com as diferentes articulações, chegando a um total de 18 pontos-chave. Todo este processo é feito de forma automática por parte do *OpenPose*. Segue-se a fase de pré-processamento dos dados do esqueleto, constituída por 4 etapas. Em primeiro lugar é feito um reajustamento da escala das imagens, de forma a se adaptar a todos os tamanhos possíveis. Após isso são removidas as articulações da cabeça do esqueleto, uma vez que são desnecessárias para este cenário, seguido de uma filtragem dos quadros onde não foi captado

certas partes do corpo, como o pescoço. A última etapa deste processo é o preenchimento de lacunas, em casos de falha por parte do *OpenPose*. Para isso é usado um método de deteção com base em previsão da posição seguinte e estado do quadro anterior, ou seja, tem-se em conta o quadro atual e com base na direção do movimento que esta a ser feito é prevista a futura localização. Isso torna-se útil para casos de ocultação parcial ou total de membros do corpo.

O processo seguinte é a recolha de dados que possam ser pertinentes e que necessitem de ser armazenados, sendo usados diferentes métodos matemáticos. Dentre esses dados destacam-se a velocidade, altura do esqueleto e os ângulos entre as articulações. A classificação é realizada através de diferentes métodos, sendo os mais importantes as *Support Vector Machines* (SVM) e as *Deep Neural Networks* (DNNs). Foram realizados testes com o *dataset NWU/UCLA* e vídeos em tempo-real. Os testes de velocidade foram realizados em um ambiente com Intel Core i7 CPU e uma gráfica NVIDIA GTX 1070, tendo sido atingidos 7 QpS, tendo sido registados valores de precisão média de 99% tanto com DNN como SVM, apesar de haver uma ligeira superioridade do primeiro.

3.11 Combinação de OpenPose e YOLO

O método proposto em [77] combina a biblioteca *OpenPose* e a rede YOLO para lidar com pontos-chave difíceis de detetar, com o objetivo de aprimorar a deteção e reconhecimento de comportamentos humanos. Os autores verificaram que o uso exclusivo do algoritmo YOLO para deteção de curta distância e do OpenPose para deteção de longa distância apresentava limitações. Dentro dessas limitações as principais é a distinção entre ações como estar em pé e caminhar, além de dificuldades na identificação precisa de alvos difusos em longa distância.

Os autores propõem uma abordagem que combina os dois algoritmos, aproveitando as vantagens de cada um e fazendo algumas alterações necessárias, nomeadamente redução dos resíduos gerados pela determinação das bordas do corpo. Esta implementação usa o *OpenPose* para extrair o esqueleto humano, bem como dados indicativos do mesmo, nomeadamente as direções das articulações. O YOLO foi aplicado no processo de classificação das ações.

O *dataset* utilizado possuí 3000 amostras de várias posturas, obtido principalmente através de um *network crawler* e de auto gravação, onde as posturas gravadas foram nomeadamente ”de pé”, ”sentado”, ”a andar” e ”a cair”.

Os testes foram realizados utilizando um computador com 8 GB de RAM e processador Intel Core i7-3770. Foram anotados 15 pontos-chave do corpo humano para reconhecer diferentes movimentos. Os dados foram iterados 8000 vezes, e a perda do treinamento atingiu uma estabilização próxima a 0.32. A análise das mudanças dos pontos-chave permitiu determinar a categoria da ação a ser executada pela pessoa.

Os autores conduziram os testes utilizando 400 imagens diferentes, incluindo alvos únicos e múltiplos, com distâncias variadas. Os resultados mostraram melhorias significativas na deteção de comportamentos humanos difusos e de longa distância, demonstrando a eficácia da abordagem proposta. Os resultados dos testes mostraram uma precisão acima de 95% e uma melhoria significativa no desempenho da abordagem em tempo real.

A Tabela 1 apresenta um resumo dos estudos encontrados na literatura científica bem como as suas principais características, nomeadamente ano de publicação, objetivos do estudo, *datasets* utilizados e metodologia aplicada.

4. Análise, Desafios e Oportunidades

Esta secção visa tecer algumas considerações sobre a pesquisa efetuada, identificar os principais desafios e as oportunidades que se apresentam nesta área.

É possível constatar que o número de trabalhos centrados na temática de análise de postura e respetiva correção é ainda muito limitado. A maioria dos trabalhos refere a deteção de postura, nomeadamente o processamento de um esqueleto *skeleton* representativo da posição da pessoa, mas poucos abordam a temática da postura. Em relação à postura os trabalhos fazem uso de uma câmara com propriedades de profundidade, hardware que apesar de começar a ser fornecido com alguns dispositivos móveis de última geração, ainda é muito raro nas gamas de preço baixa e média.

Para conseguir detetar e posteriormente acompanhar a execução de exercícios as abordagens atuais trabalham com dois algoritmos - um para detetar o exercício que está a ser realizado e outro para acompanhar o processo. Das implementações iniciais já efetuadas pelos autores um algoritmo com bons resultados numa das vertentes apresenta resultados não ótimos na outra. Assim um dos desafios é mesmo desenvolver um algoritmo que não apenas detete a pose, mas que se adapte temporalmente à mesma, permitindo em tempo-real a mudança de exercício sem intervenção do praticante.

Centrando nos algoritmos, destaca-se a limitação dos algoritmos de deteção corporal em relação à demanda computacional necessária para operar com precisão. Isso ocorre porque alguns métodos envolvem uma grande quantidade de operações matemáticas complexas e iterativas, exigindo capacidades de processamento consideráveis.

Outro desafio surge na aplicação desses sistemas em cenários de processamento de vídeo em tempo real. A capacidade computacional muitas vezes é insuficiente para suportar as múltiplas etapas do processo, resultando em falhas nos resultados e na fiabilidade.

Além disso, a complexidade surge devido à natureza variável do *hardware* de captura de imagens. Diversos dispositivos de câmera com diferentes especificações, como resolução, captura de profundidade e ângulo de visão, introduzem complicações. Isso pode levar a incongruências na qualidade da resolução, afetando a precisão dos resultados.

Outro desafio concerne as ocultações parciais ou completas de partes do corpo durante os exercícios, especialmente em

sistemas sem componente de profundidade. Resolver isso requer abordagens adicionais que, por sua vez, aumentam os requisitos computacionais e a complexidade de implementação.

Precisar a localização das pessoas em cena e detetar vários indivíduos simultaneamente é outro desafio. Isso pode ser amenizado com algoritmos de deteção multi-pessoa e transformações geométricas de imagens, como demonstrado nas abordagens anteriores.

Para cumprir os objetivos deste trabalho, é vital usar tecnologias que possam enfrentar esses dilemas de forma eficaz. Isso exige algoritmos de deteção de pessoas, classificação de exercícios, arquivamento de dados e ajustes. A adoção de estratégias em nuvem e de algoritmos eficientes, como o *TensorFlow Lite*, pode ser fundamental.

É visível a preocupação com os algoritmos e a sua precisão, mas ainda é pouco visível a sua aplicação através de uma aplicação integrada que permita ao utilizador com um dispositivo móvel fazer os seus exercícios em qualquer lugar com as posturas adequadas, diminuindo assim a possibilidade de ocorrência de lesões. Assim existe a oportunidade de criar uma aplicação que permita a integração de valências aqui apresentadas com as de treinador físico pessoal, prescrição de exercícios e respetivo acompanhamento e evolução do praticante. É crença dos autores deste artigo que a tecnologia só é benéfica se efetivamente conseguir melhorar a vida das pessoas, sendo que claramente os trabalhos aqui apresentados são um passo muito importante nesse sentido.

5. Conclusões

Este trabalho surge para tentar facilitar a criação de um sistema de deteção e ajuste de postura em exercícios físicos, de forma a permitir aos utilizadores realizarem atividades, sem a necessidade de um treinador humano e sem ser preciso se deslocarem para algum estabelecimento próprio.

O foco de trabalho é o estudo de técnicas de visão computacional, mais especificamente deteção e classificação de ações e poses humanas. Para isso, foram apresentados vários modelos de visão computacional, bem como o seu funcionamento. Em seguida foi apresentada informação sobre os vários trabalhos e investigações relativas a deteção de seres humanos e ajuste dos seus movimentos, através do uso de diversos modelos. Por fim foi efetuado um levantamento de vários problemas e oportunidades que existem dentro do tema proposto.

Durante a redação deste artigo, foi realizada uma revisão literária sobre diversas tecnologias e pesquisas abordadas dentro da área, referindo alguns dos maiores avanços tecnológicos e científicos no intuito de deteção e classificação. Esta revisão naturalmente está limitada pelo tempo em que é realizada e pelas fontes consultadas. Em termos de trabalho futuro seria possível adotar uma metodologia PRISMA , bem como incluir literatura não revista pelos pares. Com base nas análises realizadas, planeia-se implementar as metodologias propostas por referências relevantes, nomeadamente [67, 69, 70, 73, 77].

Este artigo tem como finalidade servir como fonte de informação para o desenvolvimento cientifico e tecnológico, permitindo a criação de sistemas computacionais dentro desta área. Os autores consideram que o ponto central de investigação é mesmo a necessidade de utilizar um processamento em duas fases - um para detetar o exercício que está a ser realizado e outra para acompanhar a sua execução. A maior limitação prende-se com o facto de se o utilizador mudar de exercício é preciso reiniciar o processo de deteção. Nenhuma das soluções encontradas o permite realizar automaticamente, o que pode prejudicar a atratividade da solução para o utilizador final.

Table 1. Tabela de informações básicas de cada estudo.

Referência	Ano de Publicação	Objetivo do Estudo	Dataset	Metodologia
X Chen et al. [65]	2020	Desenvolver um sistema de deteção de comportamentos perigosos em Empty Nesters e identificação de violência em recintos fechados.	NTURGB+D [78, 79], MSCOCO [80], UCF101 [81] e HMDB51 [82].	Utilização de YOLONet3 para deteção de pessoas e deteção corporal, junto com I3D para reconhecimento de ação.
T Yu et al. [66]	2019	Apresentar uma nova abordagem para capturar o movimento humano de forma rápida e precisa a partir de um único sensor RGB-D.	Dataset não definido.	Utilização de ICP-Based Skeleton Tracking para acompanhar o movimento do esqueleto. Utilização de 3D Pose Detection para detetar a pose, com inverse tracking para detetar o corpo em caso de ocultação e Semantic Bidirectional Motion Blending para melhorar a precisão do rastreamento de movimento.
B Choi et al. [67]	2022	Propor uma classificação de comportamento humano, baseado em rede neural e com baixa necessidade computacional.	CCTV fornecido pelo Korea Institute Security Agency(KISA).	Utilização de YOLOv4 para deteção da pessoa. Utilização de OpenPose para identificação das joints.
Q Pham et al. [68]	2022	Desenvolvimento de uma aplicação de exercício para captar os movimentos humanos e fornecer avaliação do mesmo.	Dataset criado pelos autores.	MediaPipe para extração do esqueleto humano. DD-Net para identificar a ação a ser realizada.
H Yan et al. [69]	2020	Propor um método eficaz de deteção em tempo-real do corpo humano usando OpenPose e FCN.	Dataset fornecido pelo departamento de reabilitação do hospital da universidade de Zhengzhou.	Usado OpenPose com filtros kalman para captação do esqueleto. Usada uma FCN para classificação da ação.
H Jeon et al. [70]	2021	Novo modelo para criar aplicações de exercício físico em dispositivos móveis.	Dataset criado pelos autores.	Uso de MobileNet para treinar o algoritmo e identificar o movimento. TensorFlow para verificar a eficiência.
K Yamao et al. [72]	2021	Sistema de reconhecimento de pose humana com um Raspberry Pi.	Dataset não definido.	Extração de dados de imagem com OpenPose. Uso de MobileNetv1 para classificação de movimentos.
Z Shu et al. [5]	2020	Melhoria de uso do OpenPose para deteção em tempo-real do corpo humano.	Dataset criado pelos autores com uso de imagens descarregadas da internet.	Realizadas operações de transformação de imagens. Uso de OpenPose para deteção do corpo humano.
V Bhamidi-pati et al. [73]	2023	Propor uma metodologia para atingir precisão elevada para determinação de pose humana, usando MediaPipe e OpenCV.	Dataset público não especificado.	Uso de técnicas de transformação de imagens para compatibilidade com o pipeline. Arquitetura de pipelines baseada em MediaPipe. Uso de OpenCV para auxiliar o treinamento do algoritmo.
Y Jiang et al. [74]	2020	Proposto sistema de extração de pontos chave do corpo humano, usando um dispositivo de baixo poder computacional.	COCO [80].	Deteção da posição com um método Lightweigth OpenPose. Uso de MobileNetv1 para treino do modelo.

Table 1. Tabela de informações básicas de cada estudo.

Referência	Ano de Publicação	Objetivo do Estudo	Dataset	Metodologia
A Rao et al. [76]	2020	Identificar diferentes tipos de ações que estão a ser feitas por um utilizador através de OpenPose e CNN.	NWU/UCLA [83].	Uso de uma câmara para captar as imagens. OpenPose para detetar e extrair dados do corpo. SVM e DDN para classificação.
W Lin et al. [77]	2020	Detetar posições humanas com micro pontos chaves através da combinação de OpenPose com YOLO.	Dataset criado pelos autores.	Utilização de OpenPose para extração do esqueleto. Uso do YOLO para classificação da ação.

Contribuições dos Autores

João Gonçalves, João Palhares: investigação, metodologia, análise formal, validação, preparação de rascunho de redação inicial.
Vasco N. G. J. Soares, Paulo A. C. S. Neves: análise formal, validação, revisão da escrita e edição, supervisão.

Referências

[1] 4 Reasons Personal Trainer Software is important to your growth - Clubworx. Acedido em 22 Agosto 2023. Disponível em: ⟨https://www.clubworx.com/blog/4-reasons-personal-trainer-software-is-important-to-your-growth⟩.

[2] PIOTROWSKI, D.; PIOTROWSKA, A. I. Operation of gyms and fitness clubs during the covid-19 pandemic-financial, legal, and organisational conditions. *Journal of Physical Education and Sport ®(JPES)*, v. 21, p. 1021–1028, 2021.

[3] HOOSHYAR, H. et al. Impact in software engineering activities after one year of covid-19 restrictions for startups and established companies. *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., v. 11, p. 55178–55203, 2023. ISSN 21693536.

[4] WHY use an App for Personal Trainers? — FitSW. Acedido em 23 Agosto 2023. Disponível em: ⟨https://www.fitsw.com/whyFitnessSoftware/⟩.

[5] SHU, Z.; WANG, P.; ZHAN, W. The research and implementation of human posture recognition algorithm via openpose. *Proceedings - 2020 2nd International Conference on Artificial Intelligence and Advanced Manufacture, AIAM 2020*, Institute of Electrical and Electronics Engineers Inc., p. 90–94, 10 2020.

[6] MENOLOTTO, M. et al. Motion capture technology in industrial applications: A systematic review. *Sensors 2020, Vol. 20, Page 5687*, Multidisciplinary Digital Publishing Institute, v. 20, p. 5687, 10 2020. ISSN 1424-8220.

[7] TYPES of Motion Trackers And How To Use Them. Acedido em 14 Junho 2023. Disponível em: ⟨https://www.rokoko.com/insights/types-of-motion-trackers⟩.

[8] WHAT is Deep Learning? — IBM. Acedido em 14 Junho 2023. Disponível em: ⟨https://www.ibm.com/topics/deep-learning⟩.

[9] DEEP Learning what it is and why it is key to artificial intelligence - Iberdrola. Acedido em 14 Julho 2023. Disponível em: ⟨https://www.iberdrola.com/innovation/deep-learning⟩.

[10] WHAT Is Deep Learning? — How It Works, Techniques Applications - MATLAB Simulink. Acedido em 14 Julho 2023. Disponível em: ⟨https://www.mathworks.com/discovery/deep-learning.html⟩.

[11] YAMASHITA, R. et al. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, Springer Verlag, v. 9, p. 611–629, 8 2018. ISSN 18694101.

[12] A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way — by Sumit Saha — Towards Data Science. Acedido em 14 Julho 2023. Disponível em: ⟨https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53⟩.

[13] CONVOLUTIONAL Neural Networks (CNN): Step 3 - Flattening - Blogs - SuperDataScience — Machine Learning — AI — Data Science Career — Analytics — Success. Acedido em 3 Agosto 2023. Disponível em: ⟨https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-3-flattening⟩.

[14] INTRODUCTION to Convolution Neural Network - GeeksforGeeks. Acedido em 30 Julho 2023. Disponível em: ⟨https://www.geeksforgeeks.org/introduction-convolution-neural-network/⟩.

[15] CONVOLUTIONAL Neural Networks (CNN) — Architecture Explained — by Dharmaraj — Medium. Acedido em 26 Julho 2023. Disponível em: ⟨https://medium.com/@draj0718/convolutional-neural-networks-cnn-architectures-explained-716fb197b243⟩.

[16] BASIC CNN Architecture: Explaining 5 Layers of Convolutional Neural Network — upGrad blog. Acedido em 26 Julho 2023. Disponível em: ⟨https://www.upgrad.com/blog/basic-cnn-architecture/⟩.

[17] SOVIANY, P.; IONESCU, R. T. Optimizing the trade-off between single-stage and two-stage deep object detectors using image difficulty prediction. *Proceedings - 2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2018*, Institute of Electrical and Electronics Engineers Inc., p. 209–214, 9 2018.

[18] AN Overview of One-Stage Object Detection Models — Papers With Code. Acedido em 23 Julho 2023. Disponível em: ⟨https://paperswithcode.com/methods/category/one-stage-object-detection-models⟩.

[19] AN overview of object detection: one-stage methods. Acedido em 23 Julho 2023. Disponível em: ⟨https://www.jeremyjordan.me/object-detection-one-stage/⟩.

[20] BIBLIOMETRIC Analysis of One-stage and Two-stage Object Detection. Acedido em 26 Julho 2023. Disponível em: ⟨https://www.researchgate.net/publication/349297260_Bibliometric_Analysis_of_One-stage_and_Two-stage_Object_Detection⟩.

[21] WHAT is Two-stage detector. Acedido em 23 Julho 2023. Disponível em: ⟨https://www.tasq.ai/glossary/two-stage-detector/⟩.

[22] HSU, W. W. et al. Two-stage cascaded cnn model for 3d mitochondria em segmentation. *IST 2022 - IEEE International Conference on Imaging Systems and Techniques,*

Proceedings, Institute of Electrical and Electronics Engineers Inc., 6 2022.

[23] (PDF) What Do We Understand About Convolutional Networks? Acedido em 26 Julho 2023. Disponível em: ⟨https://www.researchgate.net/publication/324005705_What_Do_We_Understand_About_Convolutional_Networks⟩.

[24] SHELHAMER, E.; LONG, J.; DARRELL, T. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, v. 39, p. 640–651, 11 2014. ISSN 01628828. Disponível em: ⟨https://arxiv.org/abs/1411.4038v2⟩.

[25] AMATO, A. et al. Background subtraction technique based on chromaticity and intensity patterns. *Proceedings - International Conference on Pattern Recognition*, Institute of Electrical and Electronics Engineers Inc., 2008. ISSN 10514651.

[26] THE difference between the CNN and FCN (the transforming of fully... — Download Scientific Diagram. Acedido em 23 Dezembro 2022. Disponível em: ⟨https://www.researchgate.net/figure/The-difference-between-the-CNN-and-FCN-the-transforming-of-fully-connected-layers-into_fig15_341403564⟩.

[27] REVIEW: FCN — Fully Convolutional Network (Semantic Segmentation) — by Sik-Ho Tsang — Towards Data Science. Acedido em 22 Dezembro 2022. Disponível em: ⟨https://towardsdatascience.com/review-fcn-semantic-segmentation-eb8c9b50d2d1⟩.

[28] PIRAMANAYAGAM, S. et al. Supervised classification of multisensor remotely sensed images using a deep learning framework. *Remote Sensing*, MDPI AG, v. 10, 9 2018. ISSN 20724292.

[29] REDMON, J.; FARHADI, A. Yolov3: An incremental improvement. Association for Computing Machinery, Inc, 4 2018.

[30] YOLOV3: Real-Time Object Detection Algorithm (Guide) - viso.ai. Acedido em 2 Janeiro 2023. Disponível em: ⟨https://viso.ai/deep-learning/yolov3-overview/⟩.

[31] YOLO: Real-Time Object Detection. Acedido em 2 Janeiro 2023. Disponível em: ⟨https://pjreddie.com/darknet/yolo/⟩.

[32] MACHINE Learning with ML.NET - Object detection with YOLO. Acedido em 5 Janeiro 2023. Disponível em: ⟨https://rubikscode.net/2021/04/05/machine-learning-with-ml-net-object-detection-with-yolo/⟩.

[33] YANG, F. et al. Make skeleton-based action recognition model smaller, faster and better. 7 2019.

[34] CAO, Z. et al. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, v. 43, p. 172–186, 1 2021. ISSN 19393539.

[35] THE Complete Guide to OpenPose in 2023 - viso.ai. Acedido em 23 Julho 2023. Disponível em: ⟨https://viso.ai/deep-learning/openpose/⟩.

[36] OPENPOSE : Human Pose Estimation Method - GeeksforGeeks. Acedido em 23 Julho 2023. Disponível em: ⟨https://www.geeksforgeeks.org/openpose-human-pose-estimation-method/⟩.

[37] OPENPOSE Research Paper Summary: Multi-Person 2D Pose Estimation with Deep Learning — by Chonyy — Towards Data Science. Acedido em 23 Julho 2023. Disponível em: ⟨https://towardsdatascience.com/openpose-research-paper-summary-realtime-multi-person-2d-pose-estimation-3563a4d7e66⟩.

[38] MULTI Person Pose Estimation in OpenCV using OpenPose. Acedido em 23 Julho 2023. Disponível em: ⟨https://learnopencv.com/multi-person-pose-estimation-in-opencv-using-openpose/⟩.

[39] CARREIRA, J. et al. Quo vadis, action recognition? a new model and the kinetics dataset. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Institute of Electrical and Electronics Engineers Inc., v. 2017-January, p. 4724–4733, 5 2017.

[40] GOWADA, R.; PAWAR, D.; BARMAN, B. Unethical human action recognition using deep learning based hybrid model for video forensics. *Multimedia Tools and Applications*, Springer, 2023. ISSN 15737721.

[41] SUPPORT Vector Machine — Introduction to Machine Learning Algorithms — by Rohith Gandhi — Towards Data Science. Acedido em 23 Julho 2023. Disponível em: ⟨https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47⟩.

[42] YUAN, X.; YANG, X. A robust human action recognition system using single camera. *Proceedings - 2009 International Conference on Computational Intelligence and Software Engineering, CiSE 2009*, 2009.

[43] SVM: Feature Selection and Kernels — by Pier Paolo Ippolito — Towards Data Science. Acedido em 23 Julho 2023. Disponível em: ⟨https://towardsdatascience.com/svm-feature-selection-and-kernels-840781cc1a6c⟩.

[44] LUGARESI, C. et al. *MediaPipe: A Framework for Perceiving and Processing Reality*. 2019.

[45] MEDIAPIPE: Google's Open Source Framework for ML solutions (2023 Guide) - viso.ai. Acedido em 23 Julho 2023. Disponível em: ⟨https://viso.ai/computer-vision/mediapipe/⟩.

[46] INTRODUCTION to MediaPipe — LearnOpenCV. Acedido em 23 Julho 2023. Disponível em: ⟨https://learnopencv.com/introduction-to-mediapipe/⟩.

[47] UNDERSTANDING Depthwise Separable Convolutions and the efficiency of MobileNets — by Arjun Sarkar — Towards Data Science. Acedido em 23 Julho 2023. Disponível em: ⟨https://towardsdatascience.com/understanding-depthwise-separable-convolutions-and-the-efficiency-of-mobilenets-6de3d6b62503⟩.

[48] AN Overview on MobileNet: An Efficient Mobile Vision CNN — by Srudeep PA — Medium. Acedido em 23 Julho 2023. Disponível em: ⟨https://medium.com/@godeep48/an-overview-on-mobilenet-an-efficient-mobile-vision-cnn-f301141db94d⟩.

[49] REVIEW: MobileNetV1 — Depthwise Separable Convolution (Light Weight Model) — by Sik-Ho Tsang — Towards Data Science. Acedido em 23 Julho 2023. Disponível em: ⟨https://towardsdatascience.com/review-mobilenetv1-depthwise-separable-convolution-light-weight-model-a382df364b69⟩.

[50] WANG, W. et al. A novel image classification approach via dense-mobilenet models. *Mobile Information Systems*, Hindawi Limited, v. 2020, 2020. ISSN 1875905X.

[51] (PDF) Convolutional networks for real-time 6-DOF camera relocalization. Acedido em 23 Julho 2023. Disponível em: ⟨https://www.researchgate.net/publication/277334078_Convolutional_networks_for_real-time_6-DOF_camera_relocalization⟩.

[52] POSE Estimation: The What, Why, When, How and more. Acedido em 27 Julho 2023. Disponível em: ⟨https://topflightapps.com/ideas/pose-estimation/⟩.

[53] POSENET Pose Estimation - GeeksforGeeks. Acedido em 24 Julho 2023. Disponível em: ⟨https://www.geeksforgeeks.org/posenet-pose-estimation/⟩.

[54] POSTURE Detection using PoseNet with Real-time Deep Learning project. Acedido em 24 Julho 2023. Disponível em: ⟨https://www.analyticsvidhya.com/blog/2021/09/posture-detection-using-posenet-with-real-time-deep-learning-project/⟩.

[55] PENG, Y. El net: Ensemble learning in end-to-end learning. *Journal of Physics: Conference Series*, IOP Publishing, v. 1634, p. 12029, 2020.

[56] BESL, P. J.; MCKAY, N. D. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 14, p. 239–256, 1992. ISSN 01628828.

[57] 12.2: The Iterative Closest Point (ICP) Algorithm - Engineering LibreTexts. Acedido em 11 Junho 2023. Disponível em: ⟨https://eng.libretexts.org/Bookshelves/Mechanical_Engineering/Introduction_to_Autonomous_Robots_(Correll)/12\%3A__RGB-D_SLAM/12.02\%3A_The_Iterative_Closest_Point_(ICP)_Algorithm⟩.

[58] OPENCV: Introduction. Acedido em 23 Julho 2023. Disponível em: ⟨https://docs.opencv.org/4.x/d1/dfb/intro.html⟩.

[59] (PDF) OpenCV for Computer Vision Applications. Acedido em 23 Julho 2023. Disponível em: ⟨https://www.researchgate.net/publication/301590571_OpenCV_for_Computer_Vision_Applications⟩.

[60] COMPUTER Vision Fundamentals and OpenCV Overview — by Kerem Kargın — MLearning.ai — Medium. Acedido em 23 Julho 2023. Disponível em: ⟨https://medium.com/mlearning-ai/computer-vision-fundamentals-and-opencv-overview-9a30fe94f0ce⟩.

[61] KOUGIANOS, E. et al. Design of a high-performance system for secure image communication in the internet of things. *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., v. 4, p. 1222–1242, 2016. ISSN 21693536.

[62] ABADI, M. et al. Tensorflow: A system for large-scale machine learning. *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016*, USENIX Association, p. 265–283, 5 2016.

[63] WHAT is TensorFlow, and how does it work? – Towards AI. Acedido em 25 Julho 2023. Disponível em: ⟨https://towardsai.net/p/l/what-is-tensorflow-and-how-does-it-work⟩.

[64] WHAT is TensorFlow? The machine learning library explained — InfoWorld. Acedido em 25 Julho 2023. Disponível em: ⟨https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html⟩.

[65] CHEN, X. et al. Real-time human action recognition based on person detection. *2019 IEEE International Conference on Real-Time Computing and Robotics, RCAR 2019*, Institute of Electrical and Electronics Engineers Inc., v. 2019-August, p. 225–230, 8 2019.

[66] YU, T. et al. Towards robust and accurate single-view fast human motion capture. *IEEE Access*, Institute of Electrical and Electronics Engineers Inc., v. 7, p. 85548–85559, 2019. ISSN 21693536.

[67] CHOI, B.; AN, W.; KANG, H. Human action recognition method using yolo and openpose. *International Conference on ICT Convergence*, IEEE Computer Society, v. 2022-October, p. 1786–1788, 2022. ISSN 21621241.

[68] PHAM, Q. T. et al. Automatic recognition and assessment of physical exercises from rgb images. *ICCE 2022 - 2022 IEEE 9th International Conference on Communications and Electronics*, Institute of Electrical and Electronics Engineers Inc., p. 349–354, 2022.

[69] YAN, H. et al. Real-time continuous human rehabilitation action recognition using openpose and fcn. *Proceedings - 2020 3rd International Conference on Advanced Electronic Materials, Computers and Software Engineering, AEMCSE 2020*, Institute of Electrical and Electronics Engineers Inc., p. 239–242, 4 2020.

[70] JEON, H.; KIM, D.; KIM, J. Human motion assessment on mobile devices. *International Conference on ICT Convergence*, IEEE Computer Society, v. 2021-October, p. 1655–1658, 2021. ISSN 21621241.

[71] BORZA, D. L. et al. Teacher or supervisor? effective online knowledge distillation via guided collaborative learning. *Computer Vision and Image Understanding*, Academic Press, v. 228, p. 103632, 2 2023. ISSN 1077-3142.

[72] YAMAO, K.; KUBOTA, R. Development of human pose recognition system by using raspberry pi and posenet model. *Proceedings of ISCIT 2021: 2021 20th International Symposium on Communications and Information Technologies: Quest for Quality of Life and Smart City*, Institute of Electrical and Electronics Engineers Inc., p. 41–44, 10 2021.

[73] BHAMIDIPATI, V. S. P. et al. Robust intelligent posture estimation for an ai gym trainer using mediapipe and opencv. *Proceedings of the 1st IEEE International Conference on Networking and Communications 2023, ICNWC 2023*, Institute of Electrical and Electronics Engineers Inc., 2023.

[74] JIANG, Y. et al. Rgbd-based real-time 3d human pose estimation for fitness assessment. *Proceedings - 2020 3rd World Conference on Mechanical Engineering and Intelligent Manufacturing, WCMEIM 2020*, Institute of Electrical and Electronics Engineers Inc., p. 103–108, 12 2020.

[75] (PDF) Multi-Scale Context Aggregation by Dilated Convolutions. Acedido em 12 Junho 2023. Disponível em: ⟨https://www.researchgate.net/publication/302305068_Multi-Scale_Context_Aggregation_by_Dilated_Convolutions⟩.

[76] RAO, A. Efficient min-cost real time action recognition using pose estimates. *2020 IEEE International Conference for Innovation in Technology, INOCON 2020*, Institute of Electrical and Electronics Engineers Inc., 11 2020.

[77] LIN, W.; DING, J. Behavior detection method of openpose combined with yolo network. *Proceedings - 2020 International Conference on Communications, Information System and Computer Engineering, CISCE 2020*, Institute of Electrical and Electronics Engineers Inc., p. 326–330, 7 2020.

[78] SHAHROUDY, A. et al. Ntu rgb+d: A large scale dataset for 3d human activity analysis.

[79] LIU, J. et al. Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding.

[80] COCO - Common Objects in Context. Acedido em 15 Junho 2023. Disponível em: ⟨https://cocodataset.org/#home⟩.

[81] PENG, Y.; ZHAO, Y.; ZHANG, J. Two-stream collaborative learning with spatial-temporal attention for video classification. *IEEE Transactions on Circuits and Systems for Video Technology*, Institute of Electrical and Electronics Engineers Inc., v. 29, p. 773–786, 11 2017. ISSN 10518215.

[82] WANG, L.; QIAO, Y.; TANG, X. Action recognition with trajectory-pooled deep-convolutional descriptors. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, v. 07-12-June-2015, p. 4305–4314, 5 2015.

[83] DATASETS. Acedido em 15 Junho 2023. Disponível em: ⟨http://wangjiangb.github.io/my_data.html⟩.