MoveCount: Contagem automática de movimentos em exercícios físicos

Projeto Orientado em Computação II 2023/01 - Relatório Final

André Luís Ribeiro[1], **Antonio Alfredo Ferreira Loureiro**[1]

[1]Universidade Federal de Minas Gerais
Instituto de Ciências Exatas - Departamento de Ciência da Computação
Belo Horizonte - MG - Brasil

{andre.ribeiro,loureiro}@dcc.ufmg.br

Resumo. *Exercícios físicos são uma prática essencial para a prevenção de diversas doenças, inclusive câncer e diabetes. Apesar disso, parte considerável da população brasileira não os pratica. Como auxílio para a ampliação do acesso à prática de exercícios físicos, pode-se contar com um profissional para definir e mensurar os exercícios para um indivíduo. Monitorar a realização dos exercícios, entretanto, é uma tarefa dinâmica e repetitiva. Para isso, este trabalho apresenta uma metodologia para contagem de movimentos em exercícios físicos em tempo real através de visão computacional, contemplando três movimentos: levantamento de peso, agachamento e flexão. Ao final do trabalho é ainda apresentado um aplicativo para sistemas Android, desenvolvido para validar o método com indivíduos reais, processo no qual cerca de 80% dos exercícios foram computados adequadamente.*

1. Introdução

Segundo a Organização Mundial de Saúde (OMS), a prática regular de exercícios físicos está atrelada à prevenção e ao tratamento de doenças diversas, incluindo doenças cardiovasculares, câncer e diabetes [OMS et al. 2014]. Para atingir tais benefícios, ainda segundo a OMS, considera-se que o corpo humano necessite de ao menos 150 minutos de prática leve de atividades físicas por semana. Entretanto, boa parte da população falha em atingir essa quantidade.

No Brasil, considerando o tempo estimado para atividades físicas de lazer, trabalho e deslocamento para o trabalho, em 2019, cerca de 32,1% dos homens eram considerados insuficientemente ativos segundo o Instituto Brasileiro de Geografia e Estatística (IBGE) [IBGE 2020]. Para as mulheres, o número é ainda mais alarmante, 47,5%. Esse preocupante cenário é abordado no país através de campanhas de incentivo à prática de atividades físicas, que buscam ampliar o acesso a atividades físicas por diferentes formas.

Uma das formas de atingir melhores níveis de prática de atividades físicas é através do acompanhamento de um profissional. O profissional em questão atua guiando os exercícios de um indivíduo, indicando-lhe quais exercícios performar, em que quantidade e em qual ordem de maneira personalizada. Nesse processo, é fundamental garantir que a pessoa realizou os exercícios e também mensurar quantos exercícios foram realizados. Monitorar a realização das atividades, porém, é uma tarefa dinâmica e repetitiva. Abordagens computacionais apresentam-se como um ferramental em potencial para remediar esse problema e ampliar o acesso a atividades físicas.

Nesse contexto, este projeto orientado em computação (POC) possui, como objetivo geral, a exploração de modelos e métodos computacionais para auxílio e incentivo à realização de atividades físicas em casa, em particular para o reconhecimento e contagem de movimentos físicos. Este trabalho foi desenvolvido ao longo de dois semestres. No primeiro semestre, referente ao POC I, os objetivos específicos foram:

- Identificar uma base de dados para criação de modelos de reconhecimento e contagem de exercícios físicos;
- Desenvolver uma metodologia de contagem de movimentos em atividades física a partir de visão computacional.

Já no segundo semestre, referente ao POC II, os objetivos específicos foram:

- Adaptar e verificar o funcionamento da metodologia proposta em celulares através do desenvolvimento de um aplicativo;
- Validar o funcionamento adequado da metologia proposta em cenários não sintéticos, com participação de alguns usuários convidados para testar o aplicativo desenvolvido.

O restante deste trabalho encontra-se disposto como segue. A seção 2 discorre sobre os principais conceitos e trabalhos relacionados ao tema. Em seguida, a seção 3 descreve a metodologia desenvolvida e adotada para identificação e contagem de movimentos em tempo real. Já a seção 4 apresenta os resultados obtidos por essa metodologia, além do protótipo desenvolvido para testes em tempo real. Os resultados obtidos e as possíveis razões para eles são então discutidos na seção 5. Por fim, a seção 6 relata as conclusões e possíveis trabalhos futuros.

2. Referencial teórico

Esta seção detalha alguns dos principais conceitos de áreas relacionadas ao problema em questão. Em seguida, são apresentados os principais trabalhos correlatos.

Tanto a identificação de movimentos, quanto a contagem de repetições, são tarefas que se enquadram na área de Reconhecimento de Atividade Humana (RAH) [Soro et al. 2019]. Nessa área, comumente utilizam-se de métodos de aprendizado de máquina para performar tarefas de classificação, em particular aprendizado supervisionado e, mais recentemente, com foco em métodos baseados em redes neurais.

Aprendizado de Máquina é um campo de Inteligência Artificial que sistematiza a relação entre dados e informações [Awad and Khanna 2015]. No caso de aprendizado de máquina supervisionado, temos como característica principal a utilização de dados anotados de treino. A ideia é então utilizar as anotações associadas aos exemplos de treino para "supervisionar" o aprendizado de um sistema [Cord and Cunningham 2008].

No campo de aprendizado de máquina, uma área proeminente é a de redes neurais, estruturas matemáticas inspiradas nos aspectos observados em uma rede de neurônios biológica, capazes de mapear relações complexas entre entradas e saídas [D'Addona 2014]. Redes neurais são amplamente utilizadas para reconhecimento de padrões. Na área de imagens e de RAH, uma possibilidade é seu uso para estimação de pose [Spehr 2015]. Ao estimar a pose de um indivíduo em uma imagem, é possível posteriormente classificá-la ou extrair informações a respeito de que tipo de movimentos

ou ações ele estava realizando. Existe ainda, um tipo de rede neural que permite considerar *features* simultaneamente temporais e espaciais, como é o caso de movimentos em vídeos, sendo estas as redes convolucionais tridimensionais (3D) [Tran et al. 2015].

Seja em conjunto com técnicas de aprendizado de máquina ou não, uma abordagem tradicional para identificação e reconhecimento de movimentos é a utilização de máquinas de estados para modelar um dado movimento. Em suma, uma máquina de estados é capaz de modelar um movimento mapeando-o para um conjunto de estados descrevendo as etapas do movimento e um conjunto de transições entre esses estados descrevendo a ordem em que cada etapa deve ser realizada para realizar um movimento completo [Hong et al. 2000].

2.1. Métricas para avaliação de modelos de aprendizado de máquina

A fim de avaliar o desempenho de modelos de aprendizado de máquina, algumas métricas comuns são a precisão (*precision*), a revocação (*recall*) e o *f1-score* [Powers 2020]. A precisão descreve, dentre o número de amostras previstas para um classe, quantas de fato pertenciam a ela. No contexto de contagem de movimentos, por exemplo, um modelo com precisão 1 para todas as classes, i.e. precisão máxima, jamais contaria um movimento a mais do que o esperado. Entretanto, observe que isso não significa que ele contaria todos os movimentos que deveria. Para avaliar esse segundo ponto, existe a revocação.

A revocação descreve, dentre o número de amostras pertencentes a uma classe, quantas foram previstas como sendo dela. Um modelo de contagem de movimentos com precisão perfeita para todas as classes, portanto, jamais deixaria de contar um movimento. Entretanto, note que isso não significa que não foram contados mais movimentos do que seria esperado.

A fim de ponderar entre alta precisão e alta revocação, existe uma terceira métrica, o *f1-score*, que representa uma média entre as duas primeiras. Para essa métrica, utiliza-se a média harmônica entre a precisão e a revocação, que por sua vez significa que para obter alto *f1-score* é necessário ter tanto alta precisão quanto alta revocação.

Em termos de avaliação do modelo proposto, este trabalho adota as três métricas descritas para mensurar a qualidade do modelo para uma dada classe (também chamada aqui de "estágio"). Além delas, para a contagem em si, adota-se uma representação gráfica que ilustra a contagem esperada e a contagem obtida pelo modelo ao longo dos *frames* de um vídeo.

2.2. Trabalhos correlatos

Nesta seção, foram selecionados trabalhos recentes que tratam sobre contagem de movimentos e temas correlatos. Os trabalhos foram escolhidos com base na temática, no número de citações, no meio em que foram publicados e na semelhança com o problema sendo tratado e estão dispostos partindo dos menos similares ao presente trabalho até os mais similares. Uma gama de técnicas foi considerada nesse processo a fim de desenvolver uma visão geral sobre o estado da arte da área. Como auxílio aprofundado à esta seção, recomenda-se também a leitura da bibliografia comentada, disponível no Anexo A.

Soluções para contagem e identificação de movimentos existentes na literatura se dividem em dois tipos de abordagem principais para extração de *features*

[Ferreira et al. 2021]: aquelas pautadas em na utilização de câmeras (através de visão computacional) e aquelas baseadas em sensores. Este trabalho foca no primeiro tipo de abordagem, que não requer sensores específicos instalados no corpo dos usuários para funcionar, somente imagens em tempo real ou uma captura de vídeo previamente realizada. Para a segunda abordagem, geralmente são utilizados sensores para extrair dados como aceleração e translação de um membro do usuário [Prabhu et al. 2020, Ishii et al. 2020]. Existe ainda a possibilidade de utilização de um sensor externo, como o *Kinect*, para extração das poses do usuário e posterior reconhecimento de movimentos [de Souza Vicente et al. 2016].

Na área de visão computacional a partir de imagens de câmera, diversos trabalhos exploram identificação de movimentos repetitivos, sejam de atividades físicas ou não, em vídeos previamente gravados [Zhang et al. 2021, Fieraru et al. 2021, Zhang et al. 2020]. Esse tipo de abordagem tem a possibilidade de uso de todo o contexto de um vídeo - ou de uma parte considerável - como artifício para permitir a identificação de repetições e consequente contagem. Assim, geralmente essas abordagens não podem ser diretamente aplicadas em tempo real. A *RepNet* [Dwibedi et al. 2020], rede neural voltada a reconhecimento de movimentos repetitivos, por exemplo, requer a construção de uma matriz de similaridade compreendendo todos os *frames* de um vídeo para então identificar os momentos em que ocorre o início de uma repetição. Alguns desses trabalhos fazem uso também de redes neurais convolucionais tridimensionais, seja para modelagem de gestos [Molchanov et al. 2015] ou como parte de um método para contagem de movimentos [Zhang et al. 2021].

Entretanto, para uso efetivo por ferramentas voltadas à contagem de movimentos em atividades físicas, como propõe-se neste trabalho, são necessários modelos que performem em tempo real. Trabalhos nessa linha também existem, dentre eles o apresentado por Levy et al. [Levy and Wolf 2015], que utiliza de redes neurais convolucionais para identificar a área de interesse em uma cena e posteriormente classificá-la. A partir desse modelo, os autores implementam um sistema de contagem que identifica o término de um movimento em tempo real.

Outros trabalhos envolvendo redes convolucionais também já foram adotados na área de contagem de movimentos [Zhao et al. 2018]. Ademais, métodos similares a uma rede convolucional 3D já foram utilizados em conjunto com uma modelagem baseada em uma máquina de estados para reconhecimento e contagem de movimentos, dividindo um movimento complexo em um conjunto de movimentos simples [Yang et al. 2022].

Também na área de identificação e contagem de movimentos em tempo real, uma possibilidade é a utilização de métodos de estimação de pose para obtenção de *features* que descrevem a posição do usuário em um dado *frame*. Para identificar os movimentos, a pose pode ser comparada via similaridade com um conjunto pré-definido de poses [Antón et al. 2015]. Além disso, métodos de estimação de pose já foram combinados com uma máquina de estados e com outras redes neurais para efetuar contagem e validação de movimentos [Ferreira et al. 2021], similarmente à metodologia apresentada neste trabalho, porém utilizando outro tipo de modelo e de método para representação de *features*. Por fim, a tabela 1 sumariza os trabalhos selecionados nesta seção e os principais métodos adotados por deles.

Trabalho	Obtenção de dados	Principais métodos
[Antón et al. 2015]	Câmera (tempo real)	Estimação de pose, similaridade de poses e trajetórias
[Levy and Wolf 2015]	Câmera (tempo real)	Rede convolucional
[Molchanov et al. 2015]	Vídeos	Rede convolucional 3D
[de Souza Vicente et al. 2016]	Sensor externo (Kinect)	Estimação de pose, modelo *LDCRF*
[Zhao et al. 2018]	Vídeos	Rede convolucional (*TrajectoryNet*)
[Dwibedi et al. 2020]	Vídeos	Rede convolucional + *temporal self-similarity matrix*
[Ishii et al. 2020]	Sensores vestíveis	*Dynamic time warping*
[Prabhu et al. 2020]	Sensores vestíveis	Rede convolucional
[Zhang et al. 2020]	Vídeos	Rede convolucional 3D (*3D-ResNext101*)
[Ferreira et al. 2021]	Câmera (tempo real)	Estimação de pose, rede neural, máquina de estados
[Fieraru et al. 2021]	Vídeos	Estimação de pose, auto-correlação de sinal
[Zhang et al. 2021]	Vídeos	Rede convolucional 3D
[Yang et al. 2022]	Vídeos	Rede com módulo TSM *(temporal shift model)*, máquina de estados
Este trabalho	Câmera (tempo real)	Estimação de pose, rede convolucional 3D, máquina de estados

Tabela 1. Trabalhos correlatos, formas de obtenção de dados e principais métodos adotados por eles.

3. Metodologia

Esta seção detalha o conjunto de dados utilizado para modelagem de movimentos, a versão final da rede de estimação de pose adotada e a metodologia desenvolvida para a identificação da realização de um movimento e, por fim, para a contagem.

3.1. Conjunto de dados

Para realização deste trabalho, utilizou-se o conjunto de dados *InfinityRep*[1], que contempla dez exercícios distintos. Para cada um dos exercícios, o conjunto contém cerca de 100 vídeos sintéticos, com indivíduos em diferentes cenários e vistos de diferentes ângulos e posições. Para cada vídeo, existe também um arquivo *JSON* com anotações a respeito de cada *frame*, incluindo a quantidade de movimentos realizada até aquele *frame* (representada de forma contínua) e os pontos-chave tridimensionais do esqueleto do indivíduo se movimentando no vídeo.

[1]InfinityRep: `https://github.com/toinfinityai/InfiniteRep`.

Dos movimentos existentes no *dataset*, foi selecionado um subconjunto contendo três: levantamento de peso (*overhead press*), agachamento (*squat*) e flexão (*pushup*). Ao selecionar os movimentos, prezou-se por movimentos consideravelmente distintos, que trabalham diferentes membros do corpo ou que apresentam poses notoriamente diferentes, como é o caso das poses efetuadas ao realizar uma flexão em comparação às de um levantamento de peso.

Além disso, previamente ao desenvolvimento de modelos, realizou-se uma breve caracterização dos vídeos e demais dados disponíveis no dataset, principalmente no que diz respeito a quantidade de movimentos realizados em cada vídeo. Nesse processo, foram obtidas as distribuições ilustradas na figura 1. Em média, observa-se que cada vídeo contempla cerca de sete a oito movimentos, de forma que existem cerca de 700 a 800 repetições para cada movimento analisado.

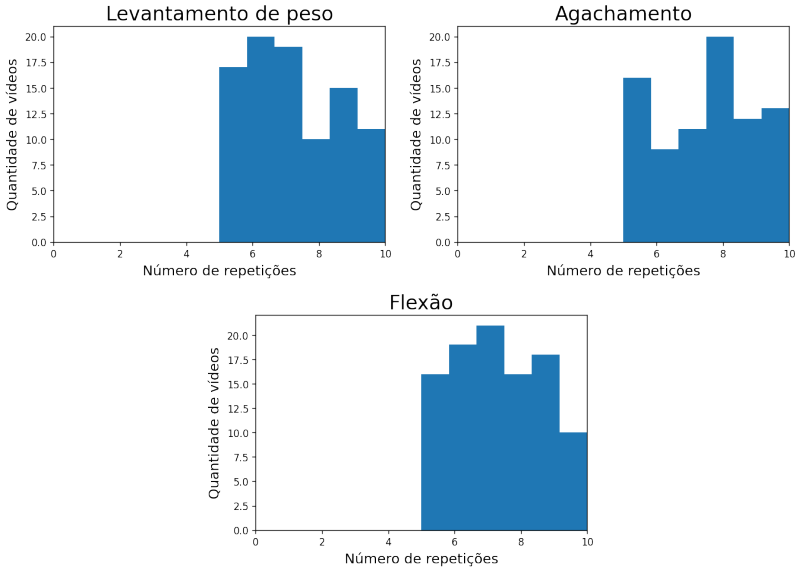

Figura 1. Distribuição do número de repetições disponível no *dataset* para cada movimento analisado.

Para efetiva utilização do conjunto de dados para contagem de movimentos, inicialmente realizou-se a discretização do número de exercícios realizados até um dado *frame*. Para isso, cada movimento foi dividido em cinco etapas. A etapa 0 representa o intervalo [0%, 20%) de um movimento completo realizado, a etapa 1 o intervalo [20%, 40%) e assim por diante até a etapa 4, com o intervalo [80%, 100%) de um movimento completo realizado. Assim, os contadores contínuos de movimentos foram mapeados para contadores inteiros seguidos do estado do movimento atualmente sendo realizado. Por exemplo, “2,768 movimentos” foi transformado em 2 movimentos e um, ainda não concluído, na etapa 3 (intervalo [60%, 80%)).

3.2. Modelagem de movimentos

A fim de modelar cada um dos movimentos considerados, foi utilizada uma abordagem baseada em máquina de estados, contendo 5 estados, um para cada etapa de movimento. Nessa abordagem, o estado 0 é o estado inicial e existe uma transição entre dois estados se e somente se o estágio representado por ele vem imediatamente após o estágio anterior.

Entre os estados 4 e 0, a transição sempre ocorre, dado que considera-se que um movimento completo foi realizado e que o próximo pode ser inicializado. A Figura 2 ilustra a máquina de estados criada.

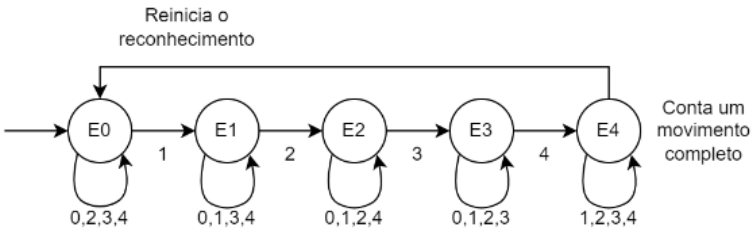

Figura 2. Máquina de estados desenvolvida para modelagem de movimentos.

3.3. Identificação de estágios

A fim de identificar o estágio de um movimento atualmente sendo realizado partindo de um vídeo ou câmera em tempo real, foi utilizado um processo dividido em duas etapas: (1) identificação do indivíduo na cena e estimação de pose, e (2) Predição do estágio a partir dos pontos chave da pose através de uma rede convolucional 3D. Nesse processo, foram considerados os últimos 8 *frames* obtidos pela câmera em questão[2]. A cada obtenção de um novo *frame* pela câmera, o processo se repete e o estágio atual do movimento é novamente previsto. A figura 3 sumariza as etapas descritas.

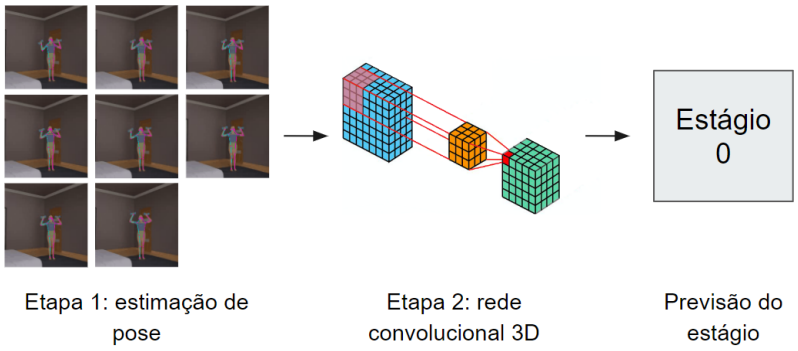

Figura 3. Processo de predição do estágio de um movimento.

3.3.1. Estimação de pose: Movenet

Como primeira etapa, realiza-se a estimação de pose do indivíduo presente na cena. Para isso, foi utilizada a *Movenet*[3], rede pré-treinada e disponibilizada via *TensorFlow Hub*. A rede consiste em um modelo rápido para estimação de pose, passível de utilização em tempo real até mesmo em alguns celulares, segundo os desenvolvedores. A *Movenet* foi treinada a partir do COCO [Lin et al. 2014], referência em *datasets* de estimação de pose,

[2]Cerca de 1 segundo de vídeo considerando a quantidade de *frames* por segundo (FPS) dos vídeos do conjunto de dados adotado.

[3]Movenet: `https://www.tensorflow.org/hub/tutorials/movenet`

acrescido de um *dataset* internamente desenvolvido pela Google. Duas principais versões do modelo são disponibilizadas: a *Movenet Thunder*, mais precisa, porém mais lenta e a *Movenet Lightning*, mais rápida, porém menos precisa. Ambos os modelos detectam 17 pontos chave referentes à pose do indivíduo na cena. Na primeira parte deste trabalho, em que se realizaram breves testes a partir de uma webcam em estrutura *desktop*, optou-se pela utilização da *Movenet Thunder*. No protótipo de aplicativo móvel desenvolvido, por outro lado, existe a possibilidade de o usuário selecionar o modelo de preferência, sendo mais recomendada a utilização da *Movenet Lightining*, devido às maiores limitações de *hardware*.

3.3.2. Predição do estágio: rede convolucional 3D

Como segunda etapa, foi utilizada uma arquitetura de rede convolucional 3D para, a partir dos pontos chave já estimados, prever em que estágio do movimento o indivíduo se encontra. A rede proposta possui uma camada de entrada de dimensão 8x17x2, sendo a entrada o X e o Y dos 17 pontos-chave previstos para cada um dos 8 *frames* considerados. Como saída, existe uma camada com 5 neurônios, um para cada fase, com ativação *sigmoid*. O neurônio de maior valor obtido representa a classe prevista.

A figura 4 ilustra as camadas da arquitetura utilizada. Para cada exercício físico foi treinada uma rede especialista a partir dessa arquitetura comum. Previamente ao treino, 30% dos dados foram separados como *dataset* de teste e, em cada época de treinamento da rede, 20% dos dados de treino foram utilizados para validação.

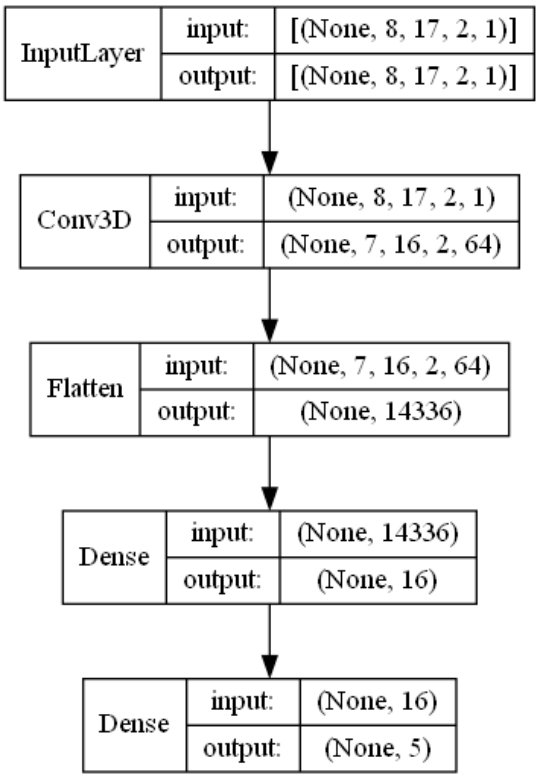

Figura 4. Arquitetura da rede neural desenvolvida.

Como camada oculta, a rede projetada possui somente uma camada convolucional 3D. Na versão inicial deste trabalho, existiam duas camadas convolucionais, separadas por *average pooling*. Nesta versão, por outro lado, existe apenas uma camada convolucional devido a limitações do *Tensorflow Lite* em relação à camada de *max pooling*. Ainda assim, adianta-se que não foram observadas diferenças de desempenho consideráveis após a redução. Além disso, uma arquitetura mais rasa é mais barata de treinar e de executar.

Como o conjunto de dados contém também um *JSON* já descrevendo cada *frame* de cada um dos vídeos utilizados, inclusive com os pontos chave da pose do indivíduo na cena, a rede convolucional foi treinada através dos dados disponibilizados por esse *JSON*, e não através da saída da *Movenet*. Sendo assim, as duas etapas só são combinadas no momento de realizar uma identificação de estágio e não no treinamento/desenvolvimento.

Por fim, para validar as redes desenvolvidas no conjunto de testes, simulou-se uma janela deslizante de 8 *frames* em cada um dos vídeos do conjunto. Na análise, foi considerada a concatenação dessas janelas.

3.4. Prototipagem

Na primeira versão deste trabalho, foi desenvolvido um pequeno protótipo que utilizava a *webcam* de um computador para executar a metodologia proposta em tempo real. Esse protótipo estava diretamente atrelado ao *Jupyter Notebook* de desenvolvimento e não oferecia opções essenciais para uma validação aprofundada, como a seleção de qual movimento está sendo realizado e o *reset* da contagem. No POC II, optou-se por desenvolver um protótipo mais completo, que permite validar a metodologia de forma mais adequada ao uso proposto. Para isso, foi desenvolvido um aplicativo móvel, mais adequado às pretensões iniciais do trabalho.

O aplicativo foi desenvolvido de forma nativa para Android, tendo em vista a necessidade de alto desempenho para execução dos modelos e o suporte oferecido pelos modelos projetados. Nesse processo, utilizou-se a linguagem *Kotlin*. Ademais, os modelos desenvolvidos em *Tensorflow* foram convertidos para *Tensorflow Lite*, o que permite que eles executem diretamente no celular dos usuários em tempo real. Como base para o desenvolvimento do aplicativo, foi utilizado o projeto[4] de exemplo do Tensorflow Lite, cujo código é aberto.

3.5. Validação com usuários

Como a metodologia foi totalmente desenvolvida a partir de dados sintéticos, uma etapa importante é submetê-la a uma validação, ainda que breve, com indivíduos reais. Essa validação objetiva garantir que os modelos não sofreram de *overfitting* devido a movimentos excessivamente precisos e mecânicos oriundos dos métodos sintéticos usados para modelar o *dataset* e também validar a metodologia como um todo.

A validação se deu com 5 indivíduos, cada qual tendo realizado 6 repetições de cada um dos movimentos suportados, totalizando 30 amostras de teste de cada exercício. No caso das flexões, 6 movimentos extras foram realizados após constatar-se que para um dos usuários em determinado ângulo de câmera, nenhum movimento havia sido contado corretamente. Previamente aos testes, os indivíduos foram instruídos em relação aos

[4]Disponível em `https://github.com/tensorflow/examples/tree/master/lite/examples/pose_estimation/android`.

movimentos e ao funcionamento do aplicativo e tiveram também cerca de 3 minutos para testá-lo livremente. Destaca-se que todos os testes foram gravados para fins de registro e de posterior análise dos resultados.

Cabe mencionar que algumas das análises realizadas no conjunto de testes sintético não puderam ser replicadas para a validação prática, dado que elas dependem do estágio do movimento em cada *frame* do vídeo. Essa extensa tarefa de rotulagem excede o escopo deste trabalho. Ademais, o interesse final é a contagem de movimentos, o que pode ser facilmente validado nos vídeos.

4. Resultados

Esta seção apresenta os resultados obtidos pela metodologia adotada. Em seguida, ilustra-se o protótipo desenvolvido para validar seu funcionamento em tempo real.

4.1. Identificação de etapas e contagem de movimentos - testes sintéticos

Inicialmente, cada uma das redes treinadas, uma para cada movimento, teve seu erro empírico estimado através do conjunto de teste. As redes foram avaliadas em relação às métricas de precisão, revocação e f1-score e, também, em relação à matriz de confusão em si.

A tabela 2 apresenta as métricas obtidas para o exercício de levantamento de peso, juntamente com a figura 5 que apresenta sua matriz de confusão. Observa-se que a maior parte das métricas fica acima de 0,8, com exceção das do estágio 0, cuja revocação foi consideravelmente baixa. Pela matriz de confusão, constata-se que geralmente essa classe foi confundida com a representante do estágio 4, anterior a ela. Entretanto, destaca-se que a revocação do estágio 0 é pouco relevante, dado que ela é a única transição que não é considerada na máquina de estados.

	precisão	revocação	f1-score	quantidade
estágio 0	0.75	0.21	0.33	675
estágio 1	0.98	0.55	0.71	1974
estágio 2	0.86	0.96	0.91	1966
estágio 3	0.90	0.98	0.94	1977
estágio 4	0.71	0.93	0.81	3110
média micro	0,82	0,82	0,82	9702
média macro	0,84	0,73	0,74	9702

Tabela 2. Métricas obtidas para levantamento de peso no conjunto de testes sintético.

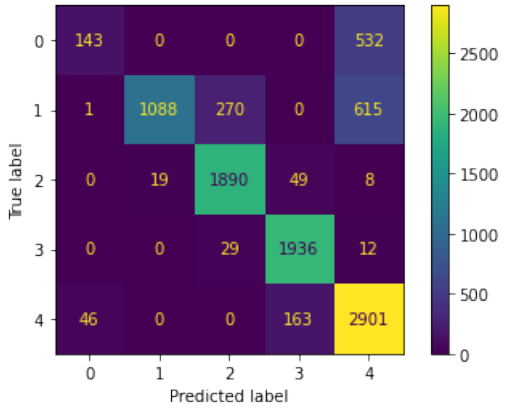

Figura 5. Matriz de confusão para levantamento de peso no conjunto de testes.

Em seguida, a tabela 3 e a figura 6 exibem as métricas e a matriz de confusão para o exercício de agachamento. Nesse caso, novamente observa-se uma baixa revocação para o estágio zero, constantemente confundido com os estágios 1 e 4, imediatamente anterior ou posterior. Para os demais estágios, as métricas tendem a permanecer acima de 0,9.

	precisão	revocação	f1-score	quantidade
estágio 0	0.65	0.52	0.58	398
estágio 1	0.91	0.93	0.92	1746
estágio 2	0.89	0.91	0.90	1774
estágio 3	0.92	0.87	0.90	1768
estágio 4	0.91	0.95	0.93	2917
média micro	0.90	0.90	0.90	8603
média macro	0.86	0.84	0.85	8603

Tabela 3. Métricas obtidas para agachamento no conjunto de testes.

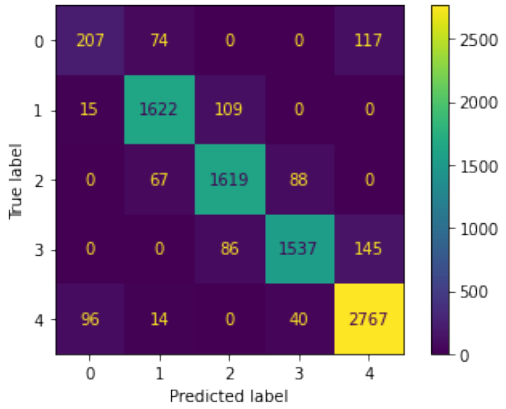

Figura 6. Matriz de confusão para agachamento no conjunto de testes.

Por fim, as métricas obtidas para a predição de estágios de flexão são ilustradas pela tabela 4 e pela figura 7. Não obstante às demais, parte dos erros se concentraram no estágio 0, também constantemente confundido com o estágio 4. Nesse caso, foi observada

também uma baixa revocação no estágio 2, o que pode ser mais preocupante que os erros no estágio 0, dado que esta transição é considerada na máquina de estados.

	precisão	revocação	f1-score	quantidade
estágio 0	0.68	0.75	0.71	508
estágio 1	0.83	0.90	0.86	2005
estágio 2	0.99	0.66	0.79	2022
estágio 3	0.83	0.92	0.87	2001
estágio 4	0.90	0.96	0.93	3268
média micro	0.87	0.87	0.87	9804
média macro	0.84	0.84	0.837	9804

Tabela 4. Métricas obtidas para flexão no conjunto de testes.

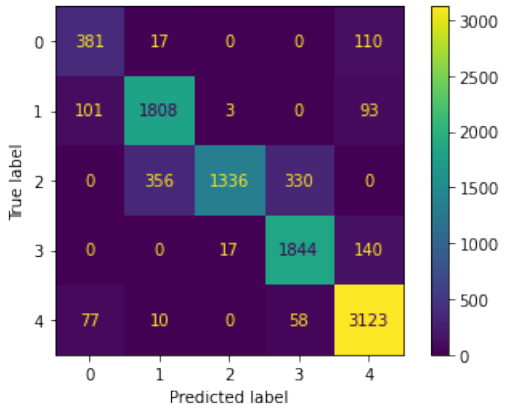

Figura 7. Matriz de confusão para flexão no conjunto de testes.

Por fim, a figura 8 apresenta um gráfico com a contagem de movimentos para cada um dos exercícios analisados. A cada término de vídeo do conjunto de testes, a contagem esperada e a prevista foi reiniciada. Observa-se que nenhum exercício foi previsto além do que deveria e que, dentre todos, somente alguns levantamentos de peso não foram corretamente identificados, principalmente de um vídeo em particular (próximo ao *frame* 8000). Destaca-se que essa verificação partiu das poses anotadas nos *JSONs*, não de poses obtidas via *Movenet* a partir dos vídeos diretamente.

4.2. Protótipo - Contagem em tempo real em *desktops*

A fim de efetivamente aplicar o processo proposto na metodologia deste trabalho, unindo as etapas de extração de pose e de identificação e contagem de movimentos, no POC I foi desenvolvido um pequeno protótipo que permite validar o funcionamento em tempo real para os três exercícios analisados. O protótipo pode ser executado a partir do próprio *Jupyter Notebook*, com *kernel Python 3*.

A figura 9 apresenta a tela do protótipo desenvolvido, executando para contagem de flexões. É possível observar em tempo real o número de movimentos previsto e o estado atual da máquina de estados através da barra superior da janela. Além disso, o esqueleto da pose estimada via *Movenet Thunder* é desenhado na tela para fins de validação da etapa 1 proposta.

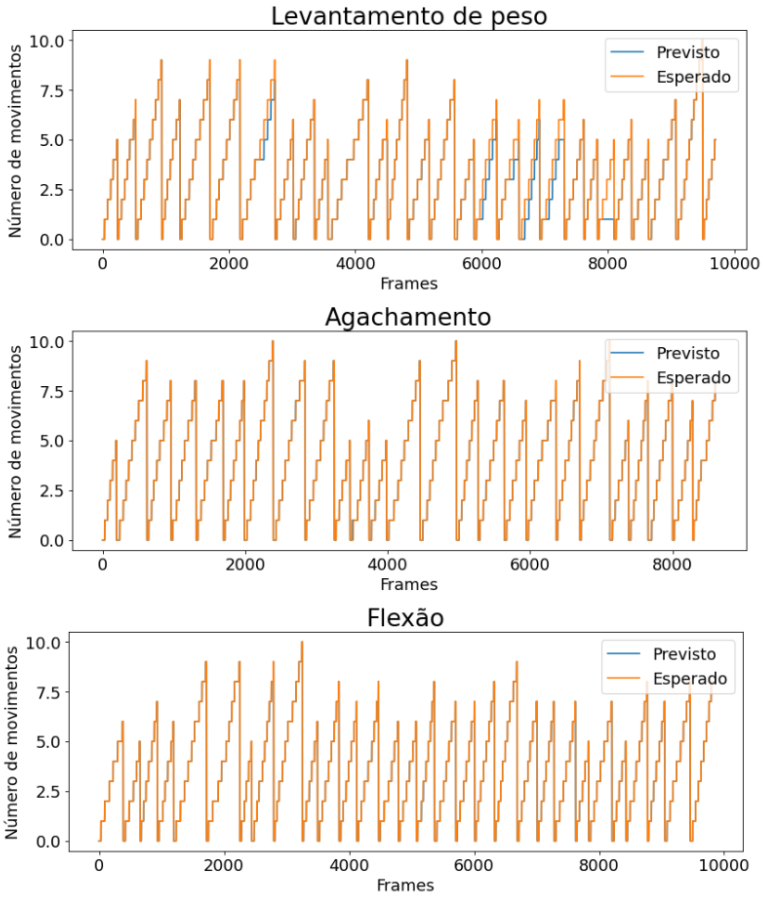

Figura 8. Contagem ao longo dos vídeos do conjunto de testes sintético.

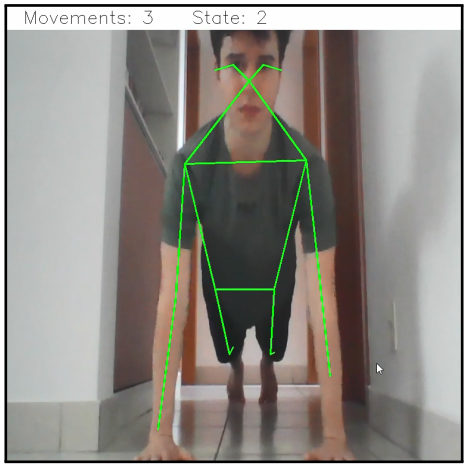

Figura 9. Protótipo de programa de contagem automática de movimentos em exercícios físicos para *desktop* - Flexão.

O protótipo desenvolvido no POC I apresentava diversas limitações, inclusive a dependência de um ambiente *Python* e do próprio *Jupyter Notebook*. Ademais, uma vez iniciado, não era possível alterar movimentos, reiniciar a contagem, nem acessar opções mais avançadas. Por fim, o ambiente *desktop* não se adequa à proposta dos modelos e da

metodologia, pensados para serem utilizados de forma mais conveniente ao usuário em dispositivos móveis. Assim, para o POC II, o foco se deu em desenvolver e validar um aplicativo mais completo para dispositivos móveis e o protótipo *desktop* foi descontinuado.

4.3. Protótipo - contagem em tempo real em dispositivos móveis

No caso do aplicativo móvel, desenvolvido no POC II, é possível selecionar o movimento a ser contado e também o modelo a ser utilizado: *Movenet Lightning* ou *Movenet Thunder*. É possível ainda alternar entre o uso de CPU e GPU, se disponível. A interface do aplicativo é mostrada na Figura 10, e inclui o desenho do esqueleto detectado do usuário e uma bandeja de opções mais avançadas na parte inferior, acessível a qualquer momento durante o uso.

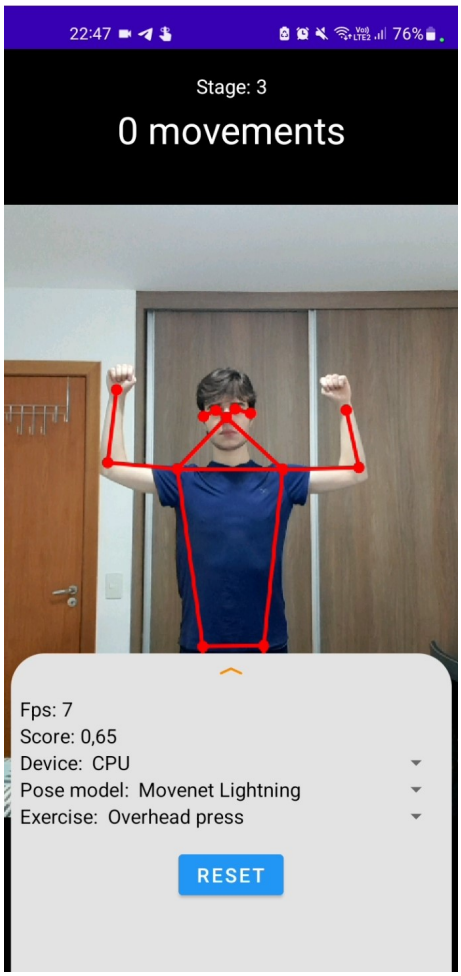

Figura 10. Aplicativo desenvolvido no decorrer do POC II.

Além disso, o aplicativo permite monitorar a taxa de quadros por segundo. A partir dela, foi desenvolvido o gráfico da Figura 11, que indica as variações de desempenho obtidas de acordo com a configuração dos testes realizados em um dispositivo básico do ano de 2020, o Samsung Galaxy A21S. De fato, constata-se o desempenho bem superior da *Movenet Lightning* em dispositivos móveis básicos. Ademais, constata-se uma melhora sutil de desempenho quando opta-se pela utilização de GPU. Destaca-se que o aplicativo

desenvolvido tem potencial para atingir performance muito superior em celulares mais modernos e de maior valor de mercado.

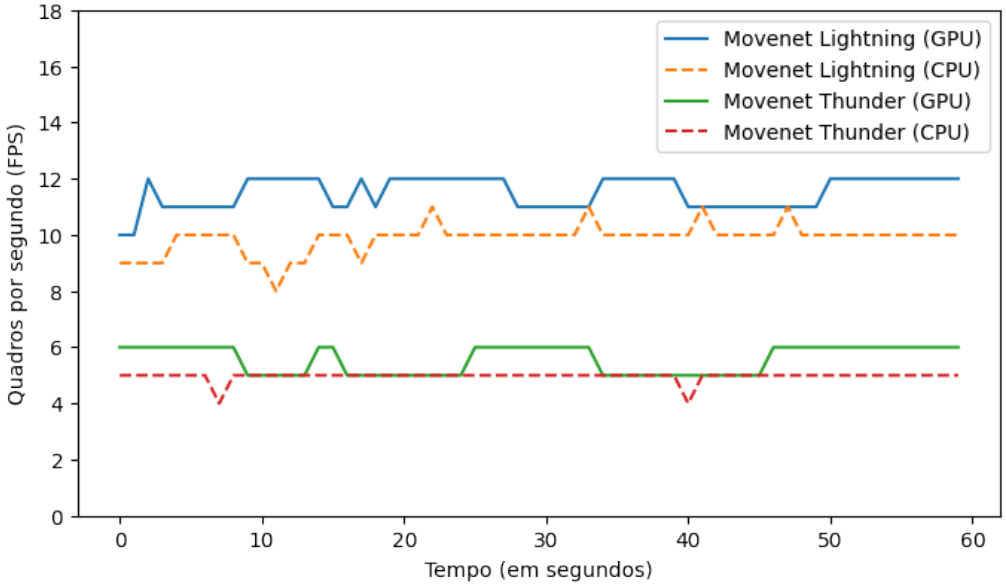

Figura 11. Taxa de quadros por segundo do aplicativo desenvolvido. Testado em um Samsung Galaxy A21S ao longo de 1 minuto para cada configuração.

4.4. Validação com usuários

O protótipo desenvolvido para dispositivos móveis foi utilizado para validar o funcionamento da metodologia com usuários reais, em casos não sintéticos. A tabela 5 indica os resultados obtidos no conjunto de testes com cinco usuários. Os resultados variam consideravelmente por indivíduo, dependendo de fatores como ambiente, ângulo da câmera, taxa de quadros por segundo no momento do teste, dentre outros. Entretanto, no geral, eles se mostram satisfatórios.

	Levantamento de peso	Agachamento	Flexão
Usuário 1	100% (6/6)	100% (6/6)	100% (6/6)
Usuário 2	100% (6/6)	66,6% (4/6)	100% (6/6)
Usuário 3	83,3% (5/6)	83,3% (5/6)	100% (6/6)
Usuário 4	100% (6/6)	66,6% (4/6)	83,3% (5/6)
Usuário 5	66,6% (4/6)	83,3% (5/6)	41,6% (5/12)
Total	90% (27/30)	80% (24/30)	77,7% (28/36)

Tabela 5. Resultados dos testes com usuários

Em particular, para o usuário em que foi solicitada a repetição da série de flexões, ao modificar o ângulo da câmera constatou-se que a contagem saltou de 0/6 movimentos para 5/6 movimentos computados adequadamente. No primeiro teste, o usuário se encontrava em posição frontal, posição esta em que a profundidade pode ser particularmente relevante. Entretanto, sendo a Movenet um modelo de estimação de pose 2D, a profundidade não pôde ser diretamente considerada. Ainda assim, esse fator não explica totalmente a falha do modelo, dado que outro usuário obteve resultados de 100% também em ângulo frontal.

5. Discussão

Nos resultados obtidos, cabe destacar dois aspectos em particular. Em primeiro lugar, a diferença entre a predição dos estágios e a contagem. Em segundo lugar, a diferença entre os resultados obtidos em testes sintéticos e os resultados obtidos nos testes com usuários.

5.1. Diferença entre predição de estágios e contagem efetiva

A diferença entre a performance de predição dos estágios em si e a contagem se dá por duas razões. Em primeiro lugar, os erros se concentram no estágio zero, que não precisa ser previsto para que ocorra uma transição. Em segundo lugar, prever um estágio errado não necessariamente significa que a transição não ocorrerá, dado que possivelmente na próxima iteração da predição o movimento ainda estará no mesmo estágio e uma predição correta ocorrerá. Cabe ainda destacar que transições só ocorrem se o estágio previsto vier imediatamente após o atual, de forma que para contar um movimento a mais que o realizado, uma série de estágios teriam que ser erroneamente previstos.

Uma das possibilidades para o número de erros relacionados ao estágio zero cometidos diz respeito ao desbalanceamento natural do conjunto de dados, que possui consideravelmente menos amostras para o estágio zero em todos os movimentos analisados. O número inferior de amostras nesse estágio, por sua vez, pode dizer respeito a características naturais dos movimentos realizados, por exemplo, um leve impulso ao início do movimento. Ademais, constata-se que geralmente o modelo troca um estágio pelo imediatamente próximo ou anterior, o que pode estar relacionado a previsões no limiar de dois estágios, quando possivelmente torna-se mais difícil distinguir entre um e outro.

Por fim, destaca-se que ainda que a maioria dos erros de predição se concentrem no estágio zero, considera-se importante que a possibilidade de prevê-lo exista. Isso porque, na prática, prever o estágio zero evita que outros estágios sejam arbitrariamente previstos quando se está inicializando o movimento.

5.2. Diferença entre testes sintéticos e testes com usuários

Já em relação à diferença entre os resultados em testes no *dataset* sintético e em testes com usuários, observou-se que os resultados com usuários costumam ser inferiores. No caso dos teste sintéticos, para dois dos três movimentos foram observados resultados de 100% de contagem. Por outro lado, com usuários, para os mesmos movimentos observou-se resultados próximos à 80%. Essa diferença possivelmente se deve a diversos fatores inerentes a cenários reais, como variações de movimento, fenótipo de usuários, taxa de quadros por segundo do aplicativo e ângulo da câmera, ao passo que cenários sintéticos tendem a ser bem mais controlados.

Em particular, cabe mencionar que o movimento com pior performance nos testes sintéticos, o levantamento de peso, foi o com melhor performance prática, o que pode ser um indício de *overfitting* dos demais modelos. Ainda assim, considera-se que os testes práticos foram satisfatórios e respaldaram a viabilidade de utilização da metodologia em dispositivos móveis de potência razoável, como celulares intermediários.

6. Conclusão

Este trabalho apresentou o desenvolvimento e validação de uma metodologia para contagem de movimentos através de estimação de pose, de redes neurais e de uma máquina

de estados. Foram explorados três movimentos distintos: levantamento de peso, agachamento e flexão. Foi ainda desenvolvido um protótipo que permite validar o funcionamento da abordagem proposta em tempo real, que foi efetivamente testado com usuários.

No geral, considera-se que a metodologia desenvolvida tenha se mostrado promissora na tarefa de contagem de movimentos em exercícios físicos. Individualmente, o reconhecimento de estágios demonstra métricas próximas ou superiores a 90% em quase todos os estágios dos exercícios nos testes sintéticos, com exceção do estágio 0 (que não é necessário para realizar transições na máquina de estados). Nos testes com usuários, os resultados foram inferiores, porém ainda validaram a possibilidade de utilização da metodologia em dispositivos móveis e atestaram seu potencial.

Alguns desafios mostraram-se presentes ao longo da realização do trabalho, dentre eles: localizar conjuntos de dados abertos para a tarefa, ponderar o custo computacional dos modelos em relação à precisão, localizar trabalhos correlatos que abordem o tema de contagem especificamente em tempo real e, por fim, unir as etapas propostas a fim de construir um protótipo e validá-lo com usuários. Uma vez superadas, considera-se que os resultados atendam aos requisitos de um Projeto Orientado em Computação, tendo cumprido o cronograma de entregas e concluído todas as etapas propostas inicialmente.

Como principais trabalhos futuros, sugere-se a exploração de diferentes modelos para identificação de estágios, utilizando arquiteturas um pouco mais profundas ou que não necessariamente a rede convolucional 3D proposta, e a validação da metodologia com outros movimentos além dos três selecionados. Ademais, seria interessante ampliar os testes com usuários e construir um *dataset* de validação a partir das gravações, que possibilitaria melhorar continuamente a metodologia desenvolvida.

Referências

[Antón et al. 2015] Antón, D., Goni, A., and Illarramendi, A. (2015). Exercise recognition for kinect-based telerehabilitation. *Methods of information in medicine*, 54(02):145–155.

[Awad and Khanna 2015] Awad, M. and Khanna, R. (2015). *Machine Learning*. Apress, Berkeley, CA.

[Cord and Cunningham 2008] Cord, M. and Cunningham, P. (2008). *Machine learning techniques for multimedia: case studies on organization and retrieval*. Springer Science & Business Media.

[D'Addona 2014] D'Addona, D. M. (2014). *Neural Network*. Springer Berlin Heidelberg, Berlin, Heidelberg.

[de Souza Vicente et al. 2016] de Souza Vicente, C. M., Nascimento, E. R., Emery, L. E. C., Flor, C. A. G., Vieira, T., and Oliveira, L. B. (2016). High performance moves recognition and sequence segmentation based on key poses filtering. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8. IEEE.

[Dwibedi et al. 2020] Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., and Zisserman, A. (2020). Counting out time: Class agnostic video repetition counting in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10387–10396.

[Ferreira et al. 2021] Ferreira, B., Ferreira, P. M., Pinheiro, G., Figueiredo, N., Carvalho, F., Menezes, P., and Batista, J. (2021). Deep learning approaches for workout repetition counting and validation. *Pattern Recognition Letters*, 151:259–266.

[Fieraru et al. 2021] Fieraru, M., Zanfir, M., Pirlea, S. C., Olaru, V., and Sminchisescu, C. (2021). Aifit: Automatic 3d human-interpretable feedback models for fitness training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9919–9928.

[Hong et al. 2000] Hong, P., Turk, M., and Huang, T. S. (2000). Gesture modeling and recognition using finite state machines. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, pages 410–415. IEEE.

[IBGE 2020] IBGE, M. d. E. (2020). Pesquisa nacional de saúde 2019: Percepção do estado de saúde, estilos de vida e doenças crônicas e saúde bucal-brasil e grandes regiões.

[Ishii et al. 2020] Ishii, S., Yokokubo, A., Luimula, M., and Lopez, G. (2020). Exersense: physical exercise recognition and counting algorithm from wearables robust to positioning. *Sensors*, 21(1):91.

[Levy and Wolf 2015] Levy, O. and Wolf, L. (2015). Live repetition counting. In *Proceedings of the IEEE international conference on computer vision*, pages 3020–3028.

[Lin et al. 2014] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

[Molchanov et al. 2015] Molchanov, P., Gupta, S., Kim, K., and Kautz, J. (2015). Hand gesture recognition with 3d convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 1–7.

[OMS et al. 2014] OMS, W. H. O. et al. (2014). *Global status report on noncommunicable diseases 2014*. Number WHO/NMH/NVI/15.1. World Health Organization.

[Powers 2020] Powers, D. M. (2020). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.

[Prabhu et al. 2020] Prabhu, G., O'Connor, N. E., and Moran, K. (2020). Recognition and repetition counting for local muscular endurance exercises in exercise-based rehabilitation: A comparative study using artificial intelligence models. *Sensors*, 20(17).

[Soro et al. 2019] Soro, A., Brunner, G., Tanner, S., and Wattenhofer, R. (2019). Recognition and repetition counting for complex physical exercises with deep learning. *Sensors*, 19(3):714.

[Spehr 2015] Spehr, J. (2015). *Human Pose Estimation*. Springer International Publishing, Cham.

[Tran et al. 2015] Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497.

[Yang et al. 2022] Yang, F., Wang, G., Li, D., Liu, N., and Min, F. (2022). Research on repetition counting method based on complex action label string. *Machines*, 10(6):419.

[Zhang et al. 2020] Zhang, H., Xu, X., Han, G., and He, S. (2020). Context-aware and scale-insensitive temporal repetition counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 670–678.

[Zhang et al. 2021] Zhang, Y., Shao, L., and Snoek, C. G. (2021). Repetitive activity counting by sight and sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14070–14079.

[Zhao et al. 2018] Zhao, Y., Xiong, Y., Lin, D., and b (2018). Trajectory convolution for action recognition. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 2208–2219, Red Hook, NY, USA. Curran Associates Inc.

Anexo A - Bibliografia comentada

Foram selecionados trabalhos recentes que discorrem sobre contagem de movimentos e temas correlatos. Os trabalhos foram escolhidos com base na temática, no número de citações, na revista em que foram publicados e, claro, na semelhança com o problema sendo tratado. Para cada trabalho, foi feito um breve resumo metodológico seguido de algumas observações pessoais a respeito da aplicabilidade da metodologia desenvolvida no problema alvo deste POC. Uma gama de técnicas foi considerada nesse processo (não apenas baseadas em visão computacional mas também em *wearables*, por exemplo) a fim de desenvolver uma visão geral sobre o estado da arte da área. Destaca-se que não necessariamente todos os trabalhos citados no referencial teórico encontram-se nesta bibliografia e que ela tem caráter meramente complementar.

Deep learning approaches for workout repetition counting and validation (2021) [Elsevier] [Real time] [pose estimation features]: Propõe um pipeline para contagem e validação de movimentos através de *features* extraídas frame a frame, incluindo keypoints (transformados para pairwise distances e keypoints flows) e heatmap flows obtidos via estimação de pose (modelo OpenPose). Classifica os *frames* em momentos (divididos em fases e *key poses*) e utiliza uma máquina de estados interna para classificação.

Observações: A metodologia parece bem estruturada e próxima da ideia inicial, porém o código e os modelos não são abertos. Apesar de não ficar 100% claro, aparentemente treina-se um modelo para cada movimento (não tem módulo de identificação de movimento integrado).

Research on Repetition Counting Method Based on Complex Action Label String (2022) [MDPI Machines] [Real time] [Template matching]: Propõe um método para contagem de movimentos complexos em tempo real. Para isso, divide um movimento complexo em um conjunto de movimentos simples, reconhecidos por um classificador TSM (similar a uma rede convolucional 3D) e gera uma assinatura (*template*) para cada movimento. Para o reconhecimento, mantém uma fila de previsões efetuadas, cuja frente é frequentemente comparada com o *template* sendo buscado.

Observações: A abordagem de decompor ações complexas em ações simples é interessante. De forma geral, o método acaba sendo bem similar a ideia de decomposição dos exercícios em estágios e de utilizar uma máquina de estados para modelar as ações.

Gesture Modeling and Recognition Using Finite State Machines (2000) [IEEE Conference on face and gesture recognition]: Apresenta modelagem de movimentos através de máquinas de estado finitos. Mapeia cada "fase" de um dado tipo de movimento para um estado específico.

Observações: Um dos clássicos da área de modelagem de reconhecimento em vídeos, a ideia de uma máquina de estados é similar a atualmente utilizada e é provavelmente

similar à mencionada em “Deep learning approaches for workout repetition counting and validation” (2021).

Live Repetition Counting (2015) [CVPR] [Real time] [agnostic]: Introduz um pipeline para contagem de repetições em tempo real estimando o tamanho da janela de *frames* necessária para repetir o movimento (tamanho do ciclo). Como *input*, o método recebe um conjunto de 20 *frames* selecionados aleatoriamente de uma sequência de 20N *frames* (em que N determina o tamanho da janela analisada e deve ser selecionado de forma que ocorram alguns movimentos completos na janela), que serão cortados de acordo com o ROI (*region of interest*) do movimento. Uma rede neural convolucional é então treinada para reconhecer o tamanho do ciclo do movimento. Na próxima interação, o segundo frame da janela anterior torna-se o primeiro frame e a previsão é refeita. O contador de movimentos é incrementado uma vez que a janela se desloque X vezes (em que X é o tamanho do ciclo previsto para a janela atual).

Observações: Relativamente bem citado, aborda especificamente o problema em tempo real. O código atualizado e o *dataset* já não estão mais disponíveis e aparenta ser relativamente complexo de reproduzir (existem diversos comentários desde 2020 solicitando o código e o modelo aos autores no Github, todos ignorados: https://github.com/ofirlevy/repcount/issues/3). A seleção do N é um dificultador. A quantidade de *frames* analisada pode ser grande e o processo como um todo aparenta ser caro. Por outro lado, é treinado de forma genérica e não é dependente do tipo de exercício.

Hand Gesture Recognition with 3D Convolutional Neural Networks (2015) [CVPR] [3DConv]: Utiliza de duas redes neurais convolucionais 3D (de três camadas separadas por max-pooling) para prever a classe de vídeos de tamanho normalizado (32 *frames*) em diferentes gestos, uma rede de alta resolução e uma de baixa.

Observações: Não é em tempo real e assume entradas de tamanho muito normalizado. Ainda assim, a adoção de uma rede convolucional 3D é interessante e similar a ideia original deste trabalho.

Learning Spatiotemporal Features with 3D Convolutional Networks (2015) [ICCV] [3DConv]: Introduz o aprendizado de movimentos em vídeo através de convoluções 3D. Sugere que convoluções 3D são aptas a serem utilizadas em situações que envolvem movimento e aparência simultaneamente (vídeos) e que um conjunto de camadas com kernel 3x3x3 é um bom ponto de partida para treinar essas redes.

Observações: Um dos trabalhos seminais de utilização de redes neurais convolucionais 3D para modelagem de movimentos (mais de 7 mil citações).

Context-aware and Scale-insensitive Temporal Repetition Counting (2020) [CVPR] [vídeos]: Modela o problema de contagem de movimentos de forma levemente diferente dos demais, mais invariante ao contexto e à escala da repetição. Dado um frame, tenta

prever o início da última repetição e o fim da próxima repetição (bidirecionalmente), de forma a estimar pontos de repetição.

Observações: Novamente, requer o contexto geral do vídeo para ser utilizado. Para o problema em questão aqui, essa preocupação com invariância introduz dificuldade para a previsão em tempo real, dado que não se presume praticamente nada sobre a constância do movimento, o que não é exatamente condizente com atividades físicas. Ainda assim, a metodologia diferente das demais é interessante.

Recognizing Exercises and Counting Repetitions in Real Time (2020) [preprint] [pose estimation features]: Extrai *features* via estimação de pose (modelo OpenPose) e conta movimentos através do monitoramento dos ângulos entre membros. Também classifica em um conjunto de exercícios pré-determinados. Abordagem mais simples, porém mais manual.

Observações: É um preprint. A metodologia parece um pouco simplória e manual. Não parece adequado para movimentos mais complexos. Ainda assim, para movimentos simples, levanta o questionamento de se só usar estados baseados em ângulos mapeados manualmente não seria suficiente.

High Performance Moves Recognition and Sequence Segmentation Based on Key Poses Filtering (Techwondo) (2015) [pose estimation features] [vídeos]: Extrai *features* via estimação de pose (kinect) e identifica diferentes movimentos de *taekwondo* através de um modelo LDCRF. Como *input*, o modelo recebe *key poses*, ou seja, somente uma seleção prévia das poses principais que compõem o movimento (no geral a inicial e a final).

Observações: A priori não é em tempo real. O foco é em reconhecimento e não em contagem.

Exercise Recognition for Kinect-based Telerehabilitation (2015) [Methods of information in medicine] [pose estimation features] [real time]: Extrai *features* de *frames* através de estimação de pose via kinect. Trata um movimento como um conjunto de passos, com uma etapa inicial e uma final. Compara as poses e a trajetória do movimento com as poses e posturas esperadas por uma métrica de similaridade.

Observações: O conceito em si é interessante e pode ser utilizado em tempo real. Entretanto, o trabalho é de 2015 e utiliza medidas de similaridade que provavelmente são menos invariantes a leves diferenças nos movimentos que um modelo estado da arte seria.

RepNet: Counting Repetitions in Videos (2020) [CVPR] [vídeos] [agnostic]: Extrai *features* via redes neurais convolucionais (arquitetura Resnet) frame a frame de um vídeo. Encontra *frames* semelhantes através das *features* extraídas comparando frame a frame. Gera então uma matriz de similaridade entre *frames* que é passada para um outro

modelo responsável por prever o período dos movimentos e se um frame compõe ou não as repetições para, finalmente, contá-los.

Observações: Bem citado e conceitualmente interessante. A abordagem de tentar prever o período lembra a do Live Repetition Counting (2015). O problema é a necessidade de ser executado em um vídeo previamente gravado para que a matriz comparando frame a frame possa ser efetivamente gerada e utilizada. Portanto, não se adequa a uma abordagem em tempo real. Tem um bom *post* no Google Blogs a respeito: https://ai.googleblog.com/2020/06/repnet-counting-repetitions-in-videos.html.

AIFit: Automatic 3D Human-Interpretable Feedback Models for Fitness Training (2021) [CVPR] [videos]: Similar a RepNet, conta repetições somente em vídeos maximizando uma auto-correlação entre *frames*. Utiliza ângulos de estimação de pose 3D como entrada para permitir também feedback sobre a corretude dos movimentos. Introduz também um *dataset* denominado *Fit3D*.

Observações: A contagem de movimentos se dá somente em vídeos e não em tempo real como é mencionado no *abstract*. Por outro lado, disponibiliza um *dataset* que aparenta ser interessante para movimentos além do que está sendo utilizado (disponível em https://fit3d.imar.ro/). Os termos do *dataset* proíbem usos com fins lucrativos.

Repetitive Activity Counting by Sight and Sound (2021) [CVPR] [vídeos] [3DConv]: Utiliza tanto a parte visual quanto o áudio (via espectrograma) de um vídeo para efetuar a contagem. Na parte de vídeo, utiliza uma rede convolucional 3D como *backbone*.

Observações: Não há aparente benefício imediato em incluir áudio na contagem de movimentos em atividades físicas. A parte de vídeo é similar a outros trabalhos com *backbone* com redes convolucionais 3D e não se adequa a tempo real. O processo como um todo aparenta ser muito caro computacionalmente.

Recognition and Repetition Counting for LME exercises in Exercise-based CVD Rehabilitation: A Comparative Study using Artificial Intelligence Models (2020) [wearables] [Convolution]: Testa abordagens de detecção de picos usando séries temporais com dados extraídos de sensores, como aceleração e translação. Para identificação dos picos, testa thresholding especificamente de acordo com o exercício e também redes convolucionais para distinguir entre pico e não pico. Como entrada das redes, cria conjuntos de imagens contendo os sinais de 6 *features* (acelerômetro 3D e giroscópio 3D) obtidas em uma janela de 4 segundos.

Observações: Utiliza sensores para obtenção das *features*, não visão computacional. A abordagem de identificação de picos via thresholding é conceitualmente simples, porém requer mais contexto e aparenta ser facilmente enviesada. A abordagem de transformar o sinal de *features* 6D em uma imagem com os gráficos, para então passar para uma rede convolucional que distingue entre picos e não picos parece forçar uma representação de um sinal em uma imagem que atenda aos requisitos da rede usada e, portanto, não convence muito. A obtenção dessas *features* sem sensores não é trivial. O tipo de exercício reconhecido deve ter picos bem determinados.

ExerSense: Physical Exercise Recognition and Counting Algorithm from Wearables Robust to Positioning (2020) [wearables]: Utiliza de sensores em dispositivos vestíveis para, através de picos de aceleração, contar movimentos. Não utiliza aprendizado de máquina na identificação / contagem, mas sim de Dynamic Time Warping para identificar de forma eficiente a que tipo de movimento os picos correspondem.

Observações: Utiliza sensores, não visão computacional. A utilização de correlação via Dynamic Time Warping é interessante, porém também só faz sentido em sequências mais longas de entrada.