UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

LUCAS LAUCK DOS PASSOS

POSEXAU: Ferramenta para auxiliar na postura durante a prática de exercício físico

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Engenharia da Computação

Orientador: Prof. Dr. Luigi Carro

Porto Alegre
2022

“If you want to go fast, go alone.
If you want to go far, go together.”
— AFRICAN PROVERB

AGRADECIMENTOS

Agradeço primeiramente a minha família que fez o possível e impossível para que eu tivesse o privilégio de estudar em uma Universidade Federal. Agradeço aos meus pais e namorada por todo o suporte nos momentos difíceis da graduação, quando muitas vezes a saúde ficou de lado em prol das provas e trabalhos.

RESUMO

É indiscutível que o exercício físico melhora a qualidade de vida e longevidade de quem o pratica. Com a chegada da pandemia muitas pessoas aderiram ao treinamento feito em casa a fim de evitar o risco de frequentar um ambiente compartilhado, e acabaram por preferir esse tipo de treinamento mesmo depois da volta das academias físicas. Por outro lado, um exercício executado de forma incorreta pode trazer lesões e ter resultado controverso, podendo prejudicar a saúde do praticante sem o acompanhamento constante. Nesse trabalho é proposta uma metodologia para auxiliar na avaliação da postura durante a execução de exercícios nomeada de POSEXAU. Para isso é usado o estado da arte da detecção de pose para extrair as informações do vídeo do usuário e, em seguida, através de modelos matemáticos analíticos propostos pelo autor, gerar uma realimentação para o praticante. Serão configurados dois exercícios, que seguem lógicas parecidas e podem ser avaliados com o POSEXAU baseados em guias de treinamento de profissionais da área. A Inteligência Artificial será responsável por identificar o movimento no vídeo enviado pelo praticante que ao final determinará se está correto ou não através de modelos matemáticos analíticos. Foi analisada a capacidade da Ferramenta OpenPose para ser usada como alimentação da POSEXAU, mas seu uso pode ser feito com qualquer rede de detecção de pose que possua resultado idêntico. O resultado final demonstra que apesar de algumas limitações ocasionadas por erros na detecção dos membros do corpo e questões físicas da perspectiva de captura do vídeo, a metodologia utilizada é robusta e pode escalar conforme a confiabilidade da rede de detecção e quantidade de análises da posição feitas na ferramenta. O sistema está previsto para funcionar seja na plataforma Windows seja na Linux, como só é feito inferência e não treinamento, é suficiente um computador com um poder de processamento de um processador i5 de quarta geração e uma câmera de 7 Megapixels no mínimo para gravar os vídeos.

Palavras-chave: Engenharia de Computação, OpenPose, OpenCV, UFRGS, Python, Exercício, Detecção de Pose, Jupyter Notebook, Rede Convolucional, POSEXAU, Numpy, Saúde, Exercício feito em casa, Correção de Pose, Inspeção, Processamento de Imagem, Visão Computacional.

POSEXAU: Tool to Auxiliate Exercises Execution Pose

ABSTRACT

It is unquestionable that physical exercise improves the quality of life and longevity of people who practice it. When pandemic started, many people joined home training in order to avoid the risk of attending a shared environment, and preferring this type of training even after the gyms back to work normally. On the other hand, an exercise performed incorrectly can cause injuries and have controversial results, which can harm the health of the practitioner without constant monitoring. In this work, a methodology is proposed to assist in posture during the execution of exercises named POSEXAU, for which the state of the art of pose detection is used to extract information from the user's video and then through analytical mathematical models proposed by the author, generate feedback for the practitioner. Two exercises will be configured, which follow similar logics and can be evaluated with POSEXAU based on training guides for professionals in the area. Artificial Intelligence will be responsible for identifying the movement in the video sent by the practitioner, in the end will determine whether it is correct or not, through analytical mathematical models. The ability of the OpenPose Tool to be used as a POSEXAU feed was analyzed, but others pose detection network that has an identical result can be use. The final result demonstrates that despite some limitations caused by errors in the detection of body members and physical issues from the perspective of capturing the video, the methodology used is robust and can scale according to the confiability of the detection network and the amount of position analysis performed. The system is designed to work on both platforms, Windows and Linux. Since it's only inference and not training, a computer with a processing power of a fourth generation i5 processor and a camera of at least 7 Megapixels to record the videos.

Keywords: Computer Engineering, UFRGS, OpenPose, OpenCV, Python, Exercise, Pose Detection, Jupyter Notebook, Convolutional Network, POSEXAU, Numpy, Health, Home Workout, Pose Correction, Inspection, Image Processing, Vision Computacional.

LISTA DE ABREVIATURAS E SIGLAS

GPU	Graphics Processing Unit
CNN	Convolutional Neural Networks
API	Application Programming Interface

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE CÓDIGOS PYTHON

SUMÁRIO

1 INTRODUÇÃO

Interpretar e melhorar imagens é um desafio antigo no processamento de imagens digitais. Já na década de 60 no Jet Propulsion Laboratory (Pesadena, California), incentivados pela a exploração aeroespacial, imagens da Lua transmitidas pelo Ranger 7 foram processadas por um computador para corrigir vários tipos de distorção das imagens inerentes à câmera de televisão a bordo (GONZALEZ, 2002). Porém, nessa época recém eram criados os primeiros computadores de 3ª geração, como IBM System/360 Model 91 e CDC 6600, o custo computacional era extremamente alto, tornando o Processamento de Imagens Digitais pouco acessível. Ao longo dos anos seguintes a tecnologia de processamento de imagens digitais fez evoluir muito a visão computacional, que começou a ter aplicações nas mais diversas áreas como medicina, indústria, física, geografia, inteligência artificial e outras.

Mais tarde, na década de 90, Yann LeCun já falava sobre Redes Neurais Convolucionais para o reconhecimento de dígitos (LECUN, 1995), Convolução é uma técnica de Redes Neurais que leva o nome de uma operação linear matemática entre matrizes, justamente por usá-la como base. Essa técnica usada por Yann LeCun só veio a ter a sua devida importância após a publicação do artigo e código-fonte que implementava a Rede Neural Convolucional ImageNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). A publicação impressionou pela qualidade de performance estatística de tais métodos na classificação de imagens, e dela partiram muitas aplicações de Aprendizado Profundo principalmente nas áreas de Computação Gráfica, Processamento de Imagens e Visão Computacional. Um fator predominante para o sucesso das Redes Neurais Convolucionais mundialmente conhecidas pelo nome em inglês *Convolutional Neural Networks* (CNNs), foi a evolução das *Graphics Processing Units* (GPUs), ou unidades de processamento gráfico, e sua superioridade sobre os processadores de propósito geral devido a sua alta densidade dos núcleos e paralelismo (SHI et al., 2016).

Após o sucesso da ImageNet, muitas outras CNNs surgiram também com intuito de classificar objetos, algumas delas tinham por objetivo identificar as partes de um corpo humano como OpenPose (CAO et al., 2021), PoseNet (KENDALL; GRIMES; CIPOLLA, 2016) e AlphaPose (FANG et al., 2017). Utilizando técnicas de Processamento de Imagens essas redes são capazes de definir onde estão localizados os membros inferiores, superiores, tronco e cabeça, podendo definir assim qual a atual posição daquele corpo detectado.

Visando a saúde e bem-estar das pessoas, nesse trabalho buscamos através do auxílio de ferramentas de manipulação de vídeos OpenCV e a ferramenta de detecção de pose corporal OpenPose, a criação de uma ferramenta chamada de POSEXAU - *Pose Exercise Auxiliary* capaz de avaliar a corretude de exercícios físicos feitos em casa. Estima-se que o mercado de aplicativos de exercícios cresceu 46% no mundo todo durante a Pandemia (ANG, 2021). Neste trabalho exploramos os conceitos de Rede Neural Convolucional usando *frames* de um vídeo para, através de processamentos de imagem desses *frames*, identificar pontos do corpo que possam definir a posição daquela pessoa. Também se deseja, através de características baseadas nos pontos encontrados no processamento, definir se aquele movimento de exercício está correto ou não. O estudo não visa desempenho, mas sim definições confiáveis de cálculos que só dependam de um apontamento preciso de pose corporal para não cometer erros, logo, nada é feito em tempo real. No trabalho foi escolhido a Ferramenta OpenPose que será melhor explicada na Seção 4.4, foi levado em conta a quantidade de documentação da ferramenta e sua confiabilidade versus tempo de Processamento.

Um ponto positivo da POSEXAU é não depender do uso exclusivo de um sistema de detecção de pose corporal. Portanto, qualquer Rede Neural que tenha como saída os pontos-chave utilizados para verificação do exercício podem ser usadas, se assim surgir uma rede mais precisa ou por necessidade da aplicação uma rede com menor poder de processamento é possível de mesma forma utilizar essa ferramenta. No trabalho foi escolhido a Ferramenta OpenPose que será melhor explicada na Seção 4.4, foi levado em conta a quantidade de documentação da Ferramenta, e sua confiabilidade versus tempo de Processamento.

2 OBJETIVOS

2.1 Objetivo Geral

Melhorar a qualidade do exercício físico das pessoas que praticam em casa, incentivar pessoas a praticar exercício com um processo mais iterativo e futurista que pode ser um fator de estímulo, facilitar o trabalho de educadores físicos que trabalham na área de acompanhamento remoto com seus clientes podendo corrigir um erro antes de causar uma lesão. Importante já ressaltar que a ferramenta não tem como objetivo substituir profissionais de Educação Física, mas sim auxiliá-los.

Desacoplar a POSEXAU da dependência de particularidades de uma CNN específica, facilitando sua integração e levando a sua evolução juntamente com as ferramentas de detecção de pose corporal.

Tornar mais simples o uso da POSEXAU podendo a pessoa se gravar com qualquer simples apoio que possibilite o celular ficar fixo e captar todo o movimento. Possibilitar que o usuário possa se enxergar na câmera frontal fazendo o movimento, a fim de facilitar o enquadramento dos membros alvos do exercício. Pensando em tornar mais acessível a um maior número de pessoas, o trabalho foi voltado para *smartphones* que, em sua maioria, possuem câmeras capazes de gerar boas imagens se o usuário tiver alguns cuidados que serão comentados mais adiante na Seção 4.4.1. Cada exercício possui suas peculiaridades e portanto exigem uma programação específica, visando extrair com maior qualidade as informações de movimento. Além das correções o POSEXAU leva em consideração os cálculos de confiabilidade indicados pela Rede Neural de detecção de Pose, capaz de determinar a qualidade da gravação e se ele pode ser levado em consideração ou se deve ser descartada a fim de não gerar resultados controversos.

2.2 Objetivos Técnicos

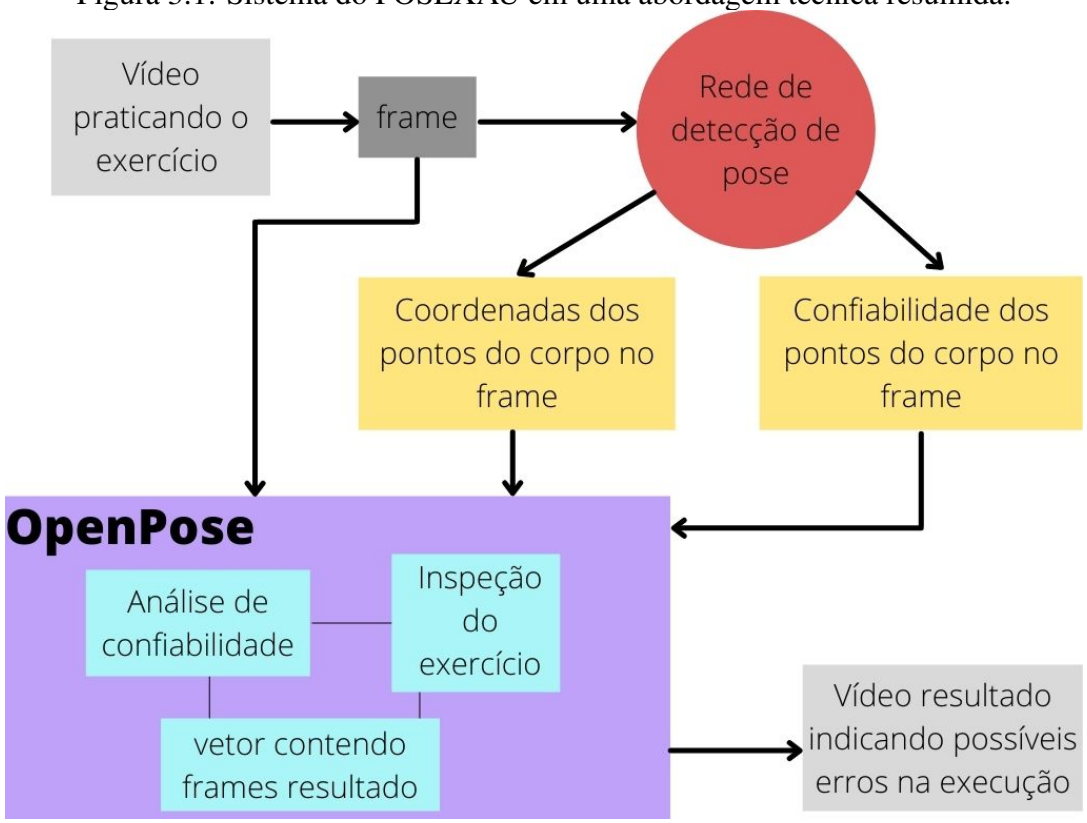
1. Investigar técnicas analíticas de processamento de imagem baseados em *frames* de um vídeo gravado praticando exercício, para através do auxílio de um educador físico produzir resultados de análise da execução correta dos exercícios;
2. Investigar uma biblioteca de estimativa de pose humana que forneça os pontos-chave do corpo e o grau de confiança entre estes de forma robusta;

3. Gravar e testar vídeos praticando diversos exercícios na Rede escolhida para determinar os exercícios iniciais;
4. Gravar vídeos em diferentes ângulos, distâncias e alturas, para reportar a influência no resultado;
5. Implementar uma inspeção analítica que utilize a saída da Rede de detecção para apontar erros na execução de um exercício;
6. Estabelecer modelos matemáticos capazes de interpretar a imagem como um gráfico 2D podendo analisar os pontos identificados pela rede através de proporção e ângulo dos membros do corpo identificados pela Rede, a fim de utilizá-las como dados de posição do corpo.

3 VISÃO GERAL

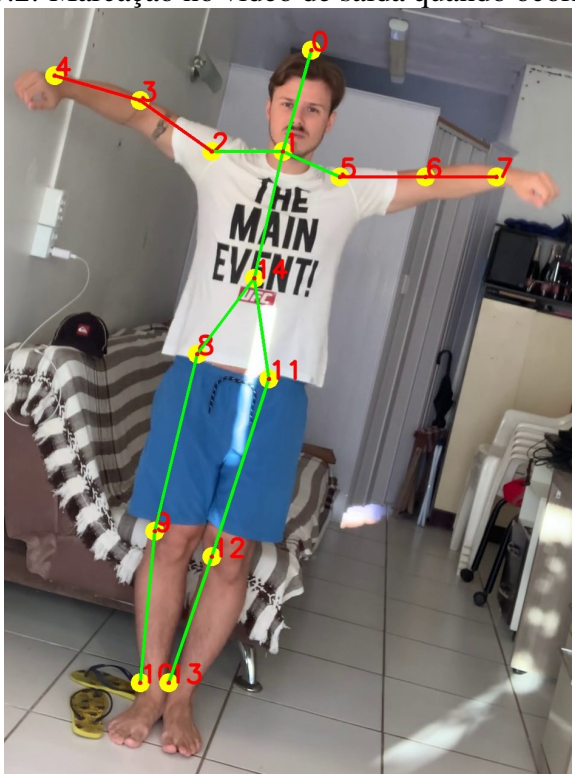
Nesse Capítulo será feita uma breve descrição das etapas que ocorrem na ferramenta POSEXAU como é visto na Figura 3.1. Primeiramente, deve ser gravado um vídeo praticando o exercício que se deseja analisar o movimento, até o momento pode ser escolhido o exercício de elevação lateral ou rosca direta. O vídeo será transformado em *frames* que serão usados como entrada na Rede de detecção de pose humana. A rede de detecção de pose humana terá como saída a detecção dos membros do corpo e a confiabilidade equivalente, esses valores serão usados como entrada da POSEXAU. Por fim, a ferramenta POSEXAU terá como saída um vetor contendo todos os *frames* do vídeo de entrada, e com eles será possível construir um vídeo de saída indicando os possíveis erros em vermelho como é visto na Figura 3.2.

Figura 3.1: Sistema do POSEXAU em uma abordagem técnica resumida.



Fonte: O Autor

Figura 3.2: Marcação no vídeo de saída quando ocorre erros.



Fonte: O Autor

4 REVISÃO DA LITERATURA

4.1 Transformação dos vídeos em imagens com OpenCV

OpenCV (*Open Source Computer Vision Library*) é uma biblioteca de código aberto e gratuita lançada em 1999 pela equipe da Intel com o objetivo de disseminar conhecimento de visão computacional e processamento de imagem. Desde então vários novos módulos surgiram e vários desenvolvedores contribuíram para o sucesso e evolução dessa biblioteca (ULJAK et al., 2012), que hoje possui mais de 2500 algoritmos otimizados para auxiliar nas áreas mais diversas de visão computacional. Sua mudança mais recente foi em 2009, com a OpenCV 2 que trouxe mudanças na interface do C++. Essa versão pode ser encontrada no site oficial do (OPENCV. . . , 2022).

Neste trabalho foi utilizado o seu pacote para linguagem de programação Python para transformar os vídeos em *frames* a uma taxa de 30 *frames* por segundo. As Imagens Coloridas vão ser responsáveis por alimentar a Rede Neural, e foram salvas em formato Jpg pensando no equilíbrio entre qualidade de informação de cada pixel e quantidade de armazenamento necessário para um vídeo de 30 *frames* por segundo. O fragmento de Código Python 4.1 mostra as partes isoladas da definição do vídeo fonte e leitura do vídeo *frame* a *frame*. Os *frames* por segundo são determinados pelo vídeo, sendo assim não há perda de informação. Quando o retorno da função de leitura for falso irá indicar que a leitura do vídeo acabou.

```
1 import cv2 as cv
2 vid = cv.VideoCapture('./video.mp4')
3 index = 0
4 while(True):
5     ret, frame = vid.read()
6     if not ret:
7         break
```

Código Python 4.1 – Transformação de vídeo para *frames*

Ao final da execução, após todos os *frames* do resultado estarem salvos em um vetor é preciso recriar o vídeo com os mesmos *frames* por segundo e dimensões do inicial. O fragmento de Código Python 4.2 mostra as partes isoladas dessa reconstrução usando OpenCV.

```
1 out = cv.VideoWriter(pathOut,cv.VideoWriter_fourcc(*'MP4V'), fps, size)
2 for i in range(len(frame_array)):
```

```
    out.write(frame_array[i])
out.release()
```

Código Python 4.2 – Transformação de *frames* para vídeo

4.2 Numpy

Numpy é uma biblioteca para a linguagem Python que inclui uma grande quantidade de funções matemáticas de alto nível para o uso do programador vistos nos fragmentos de código Python 5.1 e 5.3. Ela foi utilizada para fazer as operações matemáticas na extração de informação dos *frames*, como arco tangente e conversão de radianos para graus.

4.3 Algoritmos de Estimativa de Pose Humana

A estimativa de pose humana é uma tarefa de visão computacional que representa a orientação de uma pessoa em um formato gráfico. Esta técnica é amplamente aplicada para prever as partes do corpo de uma pessoa ou a posição das articulações. É uma das áreas mais interessantes de pesquisa em visão computacional que ganhou muita força devido à abundância de aplicativos que podem se beneficiar dessa tecnologia (WALIA, 2022).

4.3.1 Tipo 2D de Estimativa de Pose Humana

A estimativa de pose humana 2D foi a primeira a ser implementada e que é mais difundida devido ao seu sucesso. Nesse tipo, você simplesmente estima onde está localizado pontos em espaço 2D relativo ao dado de entrada que pode ser uma imagem ou *frame* de um vídeo como no caso da implementação desse trabalho. Esses pontos que podem ser localizados começaram a ser categorizados de forma simples, 15 pontos, e foram evoluindo a ponto de hoje no OpenPose, por exemplo, ser possível gerar mais de 70 pontos somente de face, com a finalidade de reconhecimento facial. Esses pontos são representado em coordenadas de X e Y em relação aos pixels da imagem. Neste trabalho é dado ênfase as partes do corpo que podem representar um corpo de forma geral e seus movimentos usando 15 pontos-chave com seus respetivos índices que são mostrados

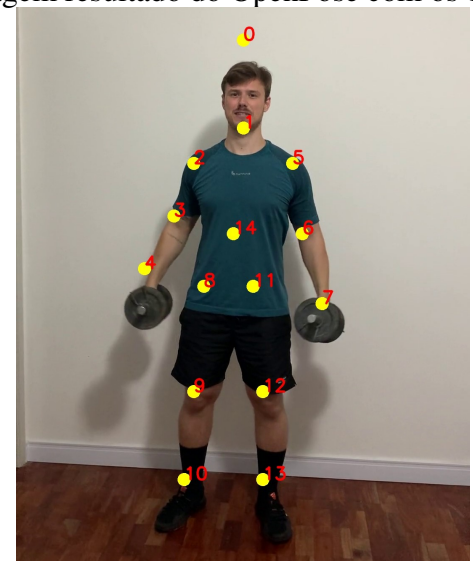
na imagem resultado. Na Tabela 4.1 é possível ver as partes do corpo e seus respetivos índices representados em um *frame* resultado da Figura 4.1:

Tabela 4.1: Pontos-chave utilizados

Parte do Corpo	*Índice*
Cabeça	0
Pescoço	1
Ombro Direito	2
Cotovelo Direito	3
Pulso Direito	4
Ombro Esquerdo	5
Cotovelo Esquerdo	6
Pulso Esquerdo	7
Anca Direita	8
Joelho Direito	9
Pé Direito	10
Anca Esquerda	11
Joelho Esquerdo	12
Pé Esquerdo	13
Peito	14

Fonte: O Autor

Figura 4.1: Imagem resultado do OpenPose com os 15 pontos-chave.



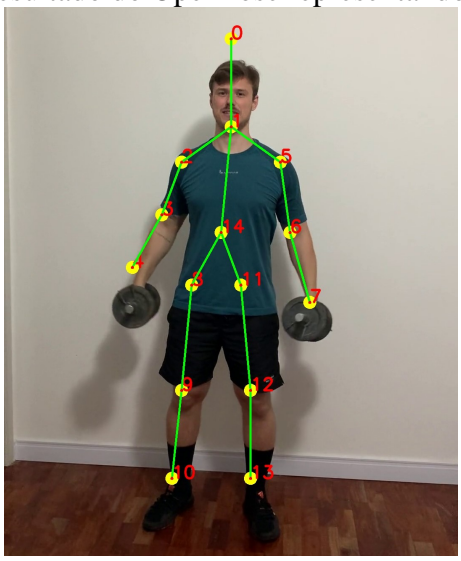
Fonte: O Autor

4.3.2 Modelo de Representação Cinemática

A cinemática é o ramo da Física que se ocupa da descrição do movimento, sem se preocupar com suas causas. Ela pode ser toda obtida da mecânica Newtoniana através

das leis de Newton, mas geralmente é exposta como um conjunto de tópicos separados e inicialmente desconexos da referida mecânica (NETO, 2016). Pensando em descrever o movimento, o Modelo Cinemático de Representação de Pose Humana define pontos, que são as articulações, que por sua vez são ligados por retas representando os membros e partes do corpo. Esta é uma forma flexível e indutiva de representar um esqueleto humano e assim interpretar de forma mais verossímil possível o seu movimento.

Figura 4.2: Imagem resultado do OpenPose representando o esqueleto humano.

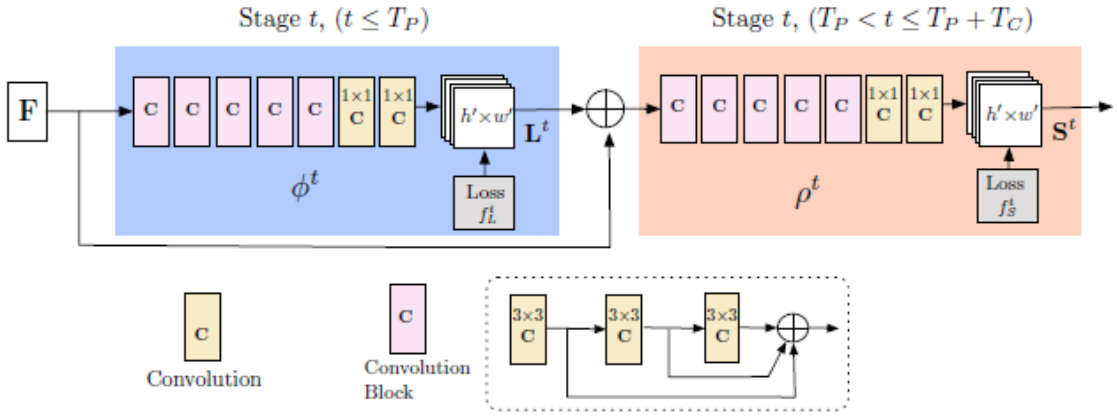


Fonte: O Autor

4.4 OpenPose

OpenPose foi a biblioteca para estimativa de pose humana escolhida para ser usada como alimentação desse trabalho (CAO et al., 2021) desenvolvida por Gines Hidalgo, Zhe Chao, e outros colaboradores do CMU-Perceptual-Computing-Lab. Trata-se de uma biblioteca multiplataforma, que pode fazer uso de GPU para aceleração e obtenção de respostas em tempo real. O OpenCV integrou o OpenPose em seu novo módulo Deep Neural Network (DNN) a partir da versão 3.4.1 e portanto ambos fazem parte da mesma API importada. A biblioteca foi utilizada com 15 pontos para o corpo, da Base MPII - Multi Person Dataset como na Figura 4.2. A OpenPose utiliza modelos treinados em uma rede VGG19(Learn OpenCV, 2018), cuja a arquitetura é mostrada na Figura 4.3, tendo como entrada uma imagem colorida, e como saída os locais 2D de pontos chave para cada pessoa na imagem.

Figura 4.3: Arquitetura OpenPose.

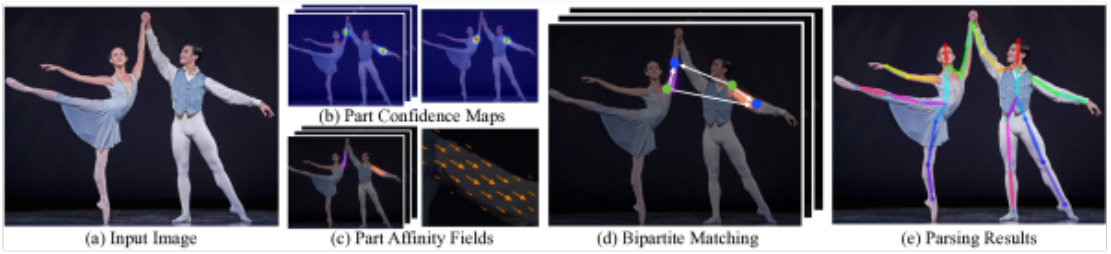


Fonte: Artigo original (CAO et al., 2021)

- *Estágio 0* – As 10 primeiras camadas criam os mapas de recursos;
- *Estágio 1* – duas ramificações preveem um conjunto 2D de mapas de confiança (S) das localizações das partes do corpo e um conjunto de vetores 2D para o mapa de afinidade das partes (L);
- *Estágio 2* - Os mapas de confiança e afinidade são analisados por inferência para produzir os pontos-chave para todas as pessoas na imagem.

Abaixo, na Figura 4.4, temos a representação do mapa de confiança para a região do ombro esquerdo (b), também vemos a representação da afinidade de partes entre o ombro esquerdo e o cotovelo esquerdo (c), em seguida a bipartição dos pontos para indicarem o corpo correto a que pertencem (d) e por fim a definição do esqueleto (e).

Figura 4.4: Pipeline geral.



Fonte: Artigo oficial (CAO et al., 2021)

A saída da rede é composta por:

- *A* – ID da imagem;

- *B* – índice do pontos-chave, sendo que para o modelo COCO temos 57 partes(18 mapas de confiança de pontos-chave + 1 plano de fundo + 19*2 mapas de afinidade de partes) e para o modelo MPII temos apenas 44;
- *C* - mapa de confiabilidade;
- *D* - altura do mapa de saída; e
- *E* - largura do mapa de saída.

Os arquivos treinados do modelo MPII, podem ser baixados no repositório do Github (LAUCK, 2022), sendo este dividido em dois arquivos:

- .prototxt – especifica a arquitetura da rede, organização das camadas e parâmetros;
- .caffemodel – armazena os pesos do modelo treinado.

No fragmento de Código Python 4.3, é possível ver a referência aos arquivos usados para definir a Rede OpenPose, a alimentação com o *frame* do vídeo e a definição dos pontos e suas afinidades.

```
net = cv.dnn.readNetFromCaffe(prototxt, caffemodel)
inpBlob = cv.dnn.blobFromImage(frame, 1.0 / 255, (inWidth, inHeight),
    (0, 0, 0),swapRB=False, crop=False)
net.setInput(inpBlob)
output = net.forward()
H = output.shape[2]
W = output.shape[3]
points = []
for i in range(15):
    probMap = output[0, i, :, :]
    minVal, prob, minLoc, point = cv.minMaxLoc(probMap)
    Height, Width = frame.shape[:2]
    x = (Width * point[0]) / W
    y = (Height * point[1]) / H
    if prob > threshold:
        cv.circle(frame, (int(x), int(y)), 15, (0, 255, 255),
        thickness=-1, lineType=cv.FILLED)
        cv.putText(frame,"{}".format(i),(int(x),int(y)),
        cv.FONT_HERSHEY_SIMPLEX,1.4, (0, 0, 255), 3,
        lineType=cv.LINE_AA)
        points.append((int(x), int(y)))
    else:
```

```
22          points.append(None)
```

Código Python 4.3 – OpenPose

4.4.1 Qualidade dos vídeos

Foi dado ênfase em câmeras de *smartphone*, devido ao objetivo final da ferramenta de atingir o público comum. Sendo elas câmeras frontais, pois facilitam uma visualização da gravação do exercício. Câmeras frontais tendem a ser piores que as câmeras traseiras, assim como possuem campo de visão menor, o que pode dificultar a captura de alguns exercícios. Veja que isso não é um fator determinante, então se por opção de quem está executando o exercício ou necessidade for gravado na câmera traseira o sistema também executará de forma normal.

Existe uma influência direta da qualidade do vídeo de entrada com o resultado final, essa influência se dá pela capacidade de inferência dos pontos da rede OpenPose depender dos padrões treinados com boa definição dos membros. Logo, apesar de não ser o objetivo do trabalho tratar questões referentes ao modo de gravar vídeos com qualidade, vale ressaltar alguns pontos observados durante a execução desse trabalho:

- Qualidade da câmera superior a 7 Megapixel;
- Quantidade de luz incidindo no ambiente para melhorar a nitidez;
- Captura de todos os membros do corpo alvos do exercício;
- Evitar que outras pessoas apareçam na gravação;
- Manter cadência no movimento para evitar borramento da imagem;

5 METODOLOGIA

5.1 Extração de informação dos pontos-chave da Imagem

A maneira escolhida e testada para avaliar a execução correta do exercício foi através de interação dos pontos entre si em um plano 2D usando trigonometria. Essa interação é medida com ângulos gerados em relação aos membros do corpo e ao eixo vertical da imagem.

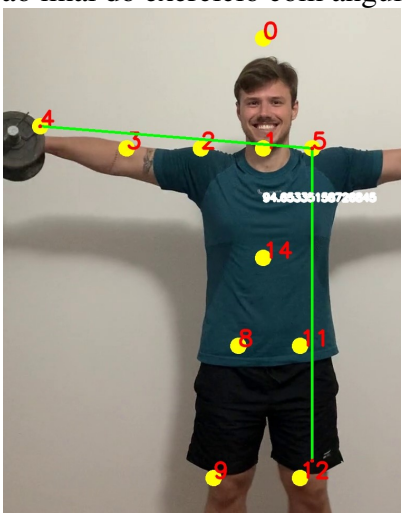
5.1.1 Extração de informação do exercício de Elevação Lateral

No exercício de ombro, elevação lateral, dois ângulos foram pensados conforme a biomecânica do exercício de ombro proposto, e são medidos através dos braços e a posição ereta mostradas nas Figuras 5.1 e 5.2.

A terceira surgiu com o objetivo de tornar essa análise mais robusta, visto que a rede necessita de um referencial vertical para medir o ângulo do movimento dos braços. Este permite que o vídeo seja gravado em ângulos que não sejam perpendiculares ao chão, sendo assim, mais inclinados em relação a posição do praticante do exercício como é visto nas Figuras 5.3 e 5.4. O ângulo medido é usado em um cálculo de ponderação na hora de verificar os ângulos dos braços, prevendo uma possível imprecisão do usuário. Dessa forma, apenas com 3 medições é possível determinar se o exercício está correto ou não, são elas:

- Ângulo entre o segmento de reta Ombro Esquerdo e Pulso Direito e o segmento de reta ligado pelo Ombro Esquerdo e seu ponto deslocado em relação ao eixo Y da imagem;

Figura 5.1: Posição final do exercício com ângulo de 94,65 graus.



Fonte: Autor

No fragmento de código Python 5.1 é possível ver as equações da biblioteca Numpy usadas para criar a função que calcula o ângulo do braço direito, que recebe com entrada o ponto do pulso direito e do ombro esquerdo respectivamente.

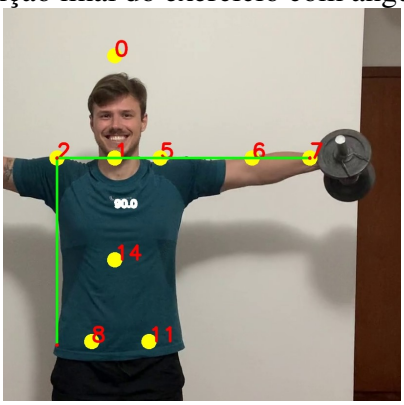
```
def calculate_right_arm_angle(RWrist,LShoulder):
    RWrist = np.array(RWrist)
    LShoulder = np.array(LShoulder)
    radians = np.pi/2 - np.arctan2(RWrist[1]-LShoulder[1], RWrist[0]-
   LShoulder[0])
    angle = np.abs(radians*180.0/np.pi)
    if angle >180.0:
        angle = 360-angle
    return angle

```

Código Python 5.1 – Ângulo do braço direito elevação lateral

- Ângulo entre o segmento de reta Ombro Direito e Pulso Esquerdo e o segmento de reta ligado pelo Ombro Direito e seu ponto deslocado em relação ao eixo Y da imagem;

Figura 5.2: Posição final do exercício com ângulo de 90 graus.



Fonte: Autor

A função que define o ângulo do braço esquerdo é análoga a demonstrada no fragmento de Código Python 5.4, alternando somente os pontos de entrada.

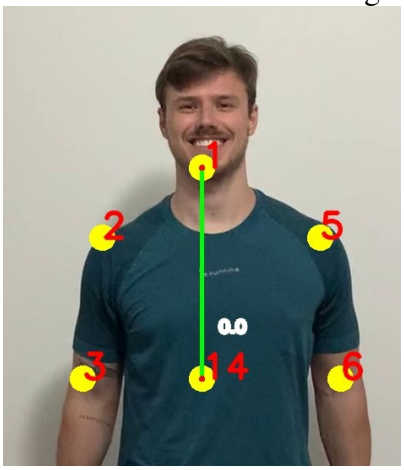
```
def calculate_left_arm_angle(LWrist,RShoulder):
    LWrist = np.array(LWrist)
    RShoulder = np.array(RShoulder)
    radians = np.pi/2 np.arctan2(LWrist[1]-RShoulder[1], LWrist[0]-
   RShoulder[0])
    angle = np.abs(radians*180.0/np.pi)
    if angle >180.0:
        angle = 360-angle
    return angle

```

Código Python 5.2 – Ângulo do braço esquerdo elevação lateral

- Ângulo do segmento de reta gerado pelos pontos Pescoço e Peito em relação ao eixo Y da imagem, gerando a informação de angulação do corpo;

Figura 5.3: Câmera bem alinhada com ângulo de 0 graus.



Fonte: Autor

Figura 5.4: Câmera mal colocada com ângulo de 10,647 graus.

Fonte: Autor

A decisão de usar os pontos do Pescoço e Peito para representar a posição do tronco foi feita por uma questão de robustez da Rede, já que ambos os pontos tem pouca variação durante a execução do exercício. Outro fator predominante foi que tais pontos, dentro da representação do esqueleto já se encontram na posição que pretende representar a posição da coluna do usuário.

No fragmento de código Python 5.3 é possível ver as equações da biblioteca Numpy usadas para criar a função que recebe como entrada o ponto do peito e pescoço respectivamente.

```
1 def calculate_person_angle(Chest,Neck):
2     Chest = np.array(Chest)
3     Neck = np.array(Neck)
```

```
    radians = np.arctan2(Chest[1]-Neck[1], 0) - np.arctan2(Chest[1]-
    Neck[1], Chest[0]-Neck[0])
    angle = np.abs(radians*180.0/np.pi)
    if (b[0] - a[0]) < 0:
        angle = angle * -1
    return angle
```

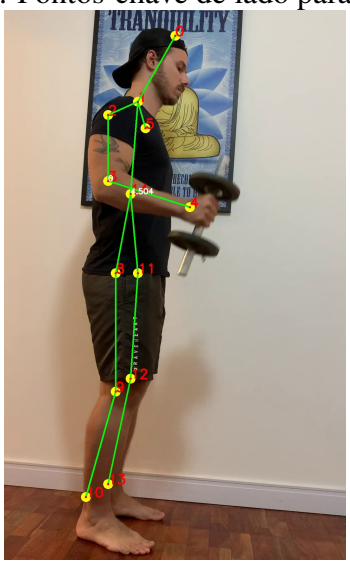
Código Python 5.3 – Ângulo do tronco elevação lateral

Já nos pontos escolhidos para medir a posição dos braços, se tentou primeiro isolar os dois pontos em membros diferentes a fim de evitar possíveis erros ligados a mesma parte da Rede, além disso, se percebeu uma menor variação no erro da Rede devido a maior distância entre os pontos.

5.1.2 Extração de informação do exercício de Bíceps

Alguns exercícios são mais desafiadores que outros, isso se deve pelas próprias características do exercício e as informações que se deseja extrair dele. O exercício de bíceps foi um dos testados que se mostrou possível de implementação somente com uma limitação de posição em que o usuário se encontra. Primeiramente o exercício foi testado com o usuário de lado para a câmera, a fim de extrair informações de somente um dos braços, considerando uma execução alternada como é visto na Figura 5.5.

Figura 5.5: Pontos-chave de lado para a câmera.



Fonte: Autor

O problema dessa posição se encontra no fato da Rede OpenPose detectar muito

mal corpos de lado, não conseguindo determinar se aquele braço é o esquerdo ou direito. Além disso, como destacado na Tabela 5.1 a confiabilidade do cotovelo direito da Figura 5.6 ficou muito baixa, não passando credibilidade para ser usada na POSEXAU.

Figura 5.6: Problema na inferência de pontos-chave de lado para a câmera.

Fonte: Autor

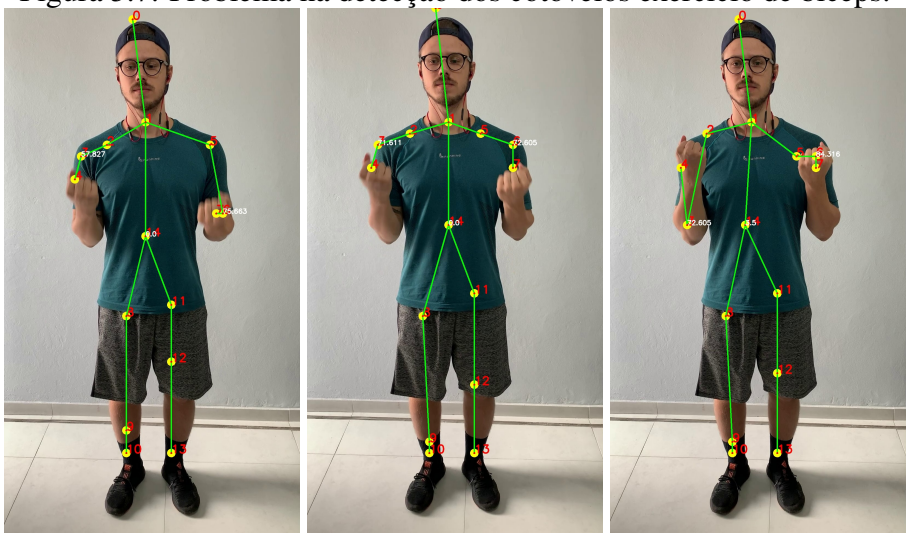
Tabela 5.1: Confiabilidade do exercício de Bíceps de lado

	Confiabilidade
Cabeça	0,344
Pescoço	0,658
Ombro Direito	0,439
Cotovelo Direito	0,150
Pulso Direito	0,367
Ombro Esquerdo	0,447
Cotovelo Esquerdo	0,076
Pulso Esquerdo	0,100
Anca Direita	0,297
Joelho Direito	0,347
Pé Direito	0,603
Anca Esquerda	0,325
Joelho Direito	0,442
Pé Direito	0,319
Peito	0,639
Total	39,67%

Descartada a posição de lado foi testada a posição de frente, que pra esse tipo de exercício não parecia uma boa escolha devido à falta de informação de inclinação do corpo para trás e posição dos cotovelos. Apesar disso, foi testada e descartada devido a problemas na detecção dos cotovelos que demonstraram uma péssima inferência quando

os pulsos estavam na altura do ombro como visto na Figura 5.7.

Figura 5.7: Problema na detecção dos cotovelos exercício de bíceps.



Fonte: Autor

Observa-se um problema na Rede para detectar os joelhos que não foi comentado por não ser o foco de análise desse exercício.

Por fim, foi testado um ângulo de aproximadamente 45 graus em relação a câmera, essa posições se mostrou promissora na confiabilidade dos pontos além de beneficiar a inspeção das posição do tronco e dos cotovelos como mostrado na Figura 5.8.

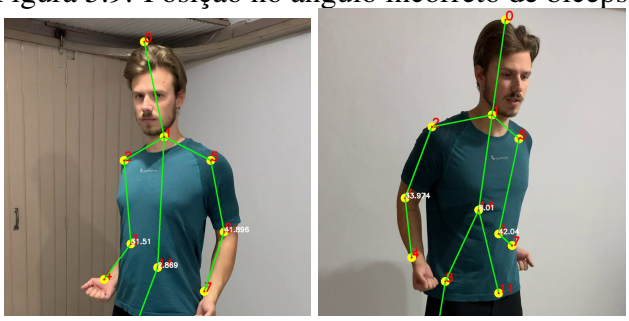
Figura 5.8: Posição mais promissora na Inspeção do Exercício Rosca Direta.

Fonte: Autor

A forma mais simples do usuário conseguir se posicionar dessa maneira, é ficando paralisado no final do movimento do exercício com os braços elevados. Nesse ponto deve

ser possível ver os dois braços e ombros por completo.

Figura 5.9: Posição no ângulo incorreto de bíceps.



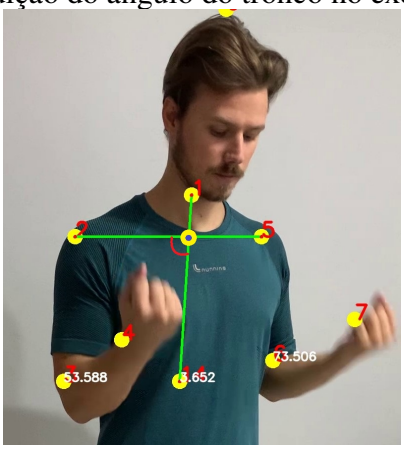
Fonte: Autor

Também é possível notar que ao se encontrar na posição correta do exercício o braço mais próximo à câmera é visto por completo, diferente de quando o usuário se encontra de frente para a câmera.

Após determinar que esta é a melhor posição para inspeção vamos tratar os ângulos que vão determinar se o exercício está correto ou não.

- Ângulo da inclinação do tronco medida pelo ponto do ombro direito, ponto de intersecção entre o segmento de reta ligando os ombros e o segmento de reta ligando pescoço e peito, e o terceiro ponto é o próprio peito. Foram utilizados apenas pontos-chave a fima de evitar possíveis erros gerados pela gravação torta;

Figura 5.10: Medição do ângulo do tronco no exercício de bíceps.



Fonte: Autor

```
def calculate_person_angle(RShoulder,LShoulder,Neck, Chest):
    RShoulder = np.array(RShoulder) # Right Shoulder
    LShoulder = np.array(LShoulder) # Left Shoulder
    Neck = np.array(Neck) # Neck
```

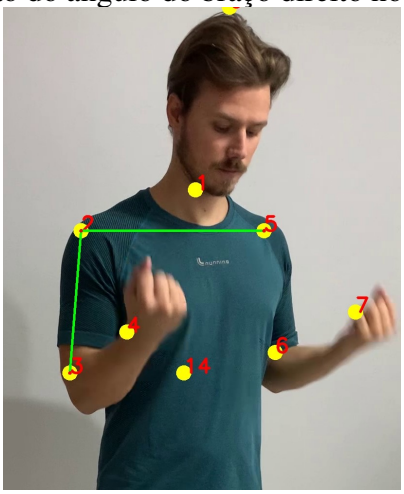
```
    Chest = np.array(Chest) # Chest
    m1 = (LShoulder[1] - RShoulder[1])/(LShoulder[0] - RShoulder[0])
    if ( Chest[0] - Neck[0] ) != 0:
        m2 = (Chest[1] - Neck[1])/(Chest[0] - Neck[0])
        xi = ( m1 * RShoulder[0] - m2 * Neck[0] - RShoulder[1] )/ (m1 -
    m2)
    else:
        xi = Neck[0]
    yi = m1 * (xi - RShoulder[0]) + RShoulder[1]
    radians = np.arctan2(Chest[1]-yi, Chest[0]-xi) -    np.arctan2(
   RShoulder[1]-yi, RShoulder[0]-xi)
    angle = np.abs(radians*180.0/np.pi)
    if angle >180.0:
        angle = 360-angle
    return angle

```

Código Python 5.4 – Ângulo do tronco rosca direta

- Ângulo da inclinação do braço direito medido em relação ao segmento de reta entre o ombro direito e o cotovelo direito, e o segmento de reta de um ombro a outro;

Figura 5.11: Medição do ângulo do braço direito no exercício de bíceps.



Fonte: Autor

```
def calculate_right_arm_angle(RWrist,RShoulder,LShoulder):
    RWrist = np.array(RWrist) # Right Wrist
    RShoulder = np.array(RShoulder) # Right Shoulder
    LShoulder = np.array(LShoulder) # Left Shoulder
    radians = np.arctan2(LShoulder[1]-RShoulder[1], LShoulder[0]-
   RShoulder[0])
```

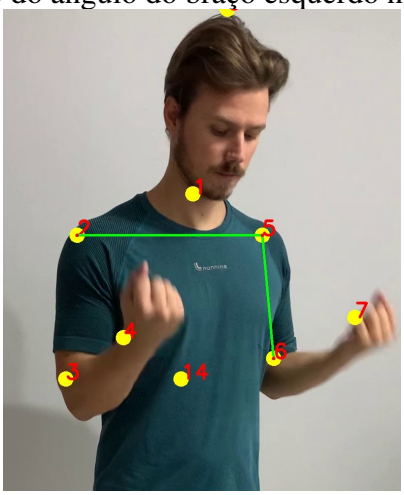
```
    - np.arctan2(RWrist[1]-RShoulder[1], RWrist[0]-RShoulder[0])
    angle = np.abs(radians*180.0/np.pi)
    if angle > 180.0:
        angle = 360-angle
    return angle

```

Código Python 5.5 – Ângulo do braço direito rosca direta

- Ângulo da inclinação do braço esquerdo medido em relação ao segmento de reta entre o ombro esquerdo e o cotovelo esquerdo, e o segmento de reta de um ombro a outro;

Figura 5.12: Medição do ângulo do braço esquerdo no exercício de bíceps.



Fonte: Autor

```
def calculate_left_arm_angle(LWrist,LShoulder,RShoulder):
    LWrist = np.array(LWrist) # Left Wrist
    LShoulder = np.array(LShoulder) # Left Shoulder
    RShoulder = np.array(RShoulder) # Right Shoulder
    radians = np.arctan2(RShoulder[1]-LShoulder[1], RShoulder[0]-
    LShoulder[0]) - np.arctan2(LWrist[1]-LShoulder[1], LWrist[0]-
    LShoulder[0])
    angle = np.abs(radians*180.0/np.pi)
    if angle > 180.0:
        angle = 360-angle
    return angle

```

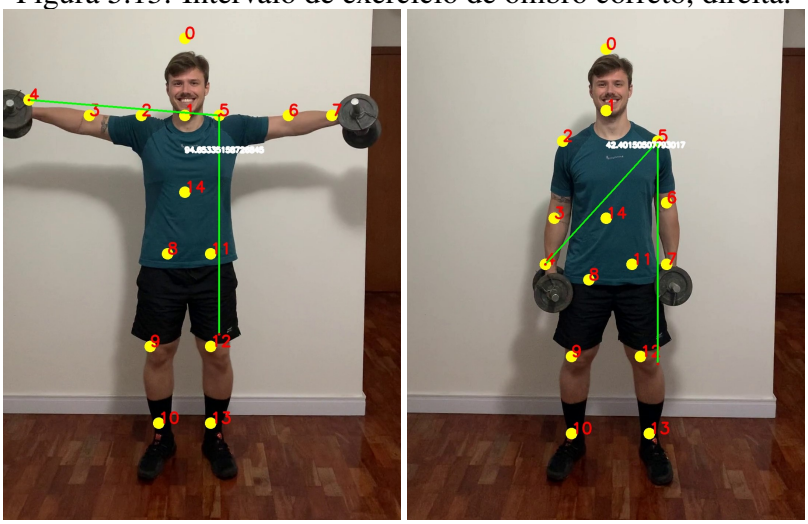
Código Python 5.6 – Ângulo do braço esquerdo rosca direta

5.2 Como foi determinada a maneira correta de fazer o exercício de Elevação Lateral

Ainda que existam adaptações de biomecânica neste exercício de ombro, para se trabalhar diferentes porções de ombro ou maneiras diferentes, uma foi considerada a mais correta levando em consideração o uso do deltoide lateral e um final de movimento comum que invalida o uso desse músculo e pode prejudicar a saúde de quem pratica. O movimento se caracteriza por ultrapassar a posição que os ombros alcancem 90 graus em relação ao tronco. Logo algumas regras foram usadas, como:

1. O braço direito em repouso se encontra colado ao corpo, representando um ângulo de aproximadamente 42 graus como observado na Figura 5.13, é desejável que o angulo desse braço não ultrapasse a altura dos ombros, 90 graus, o que indicaria uma execução incorreta.

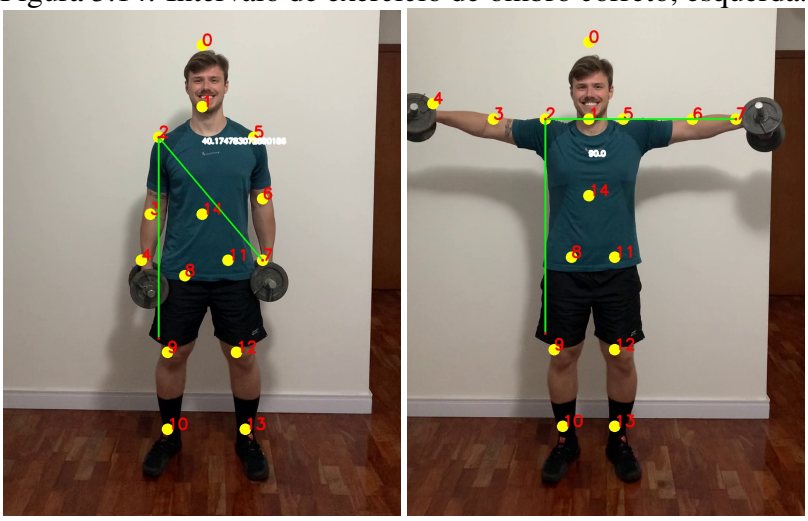
Figura 5.13: Intervalo de exercício de ombro correto, direita.



Fonte: Autor

2. Seguindo a mesma lógica, o braço esquerdo em repouso se encontra colado ao corpo, representando um ângulo de aproximadamente 40 graus como observado na Figura 5.14, da mesma forma este não deve ultrapassar os 90 graus.

Figura 5.14: Intervalo de exercício de ombro correto, esquerda.



Fonte: Autor

3. Os pontos de máximo e mínimo são determinados levando em consideração o ângulo de tronco, a fim de evitar que o usuário necessite posicionar o celular em uma angulo exato de 90 graus em relação ao chão para obter o resultado correto. Como pode ser visto o ângulo calculado na Figura 5.4, e a ponderação resultado na Figura 5.15.

Figura 5.15: Ângulo ajustado; Direita: 106,26° ; Esquerda: 100,64°.

Fonte: Autor

5.3 Como foi determinada a maneira correta de fazer o exercício de Rosca Direta

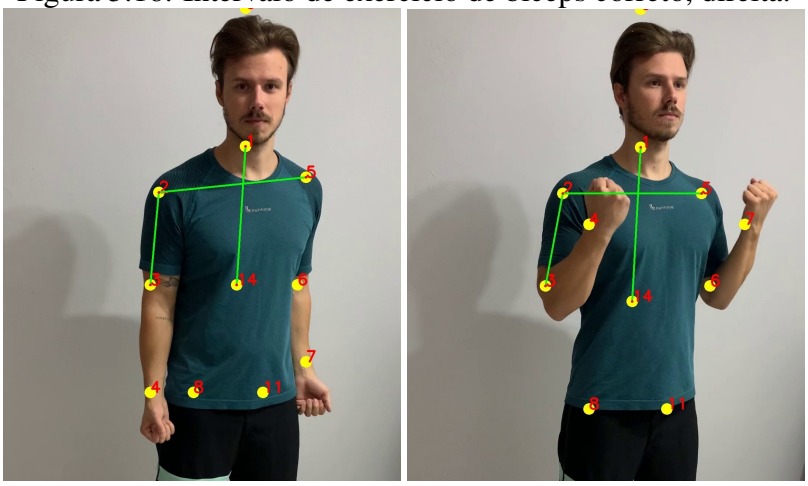
O exercício de Rosca Direta têm um variedade grande de execuções com diferentes tipos de pegada, porém todas são voltadas para o treinamento do grupo muscular bíceps. Existem dois erros muito comuns citados pelos educadores físicos consultados.

O primeiro deles é a hiperextensão lombar, caracterizado pela inclinação do tronco para trás. Esse movimento além de não trazer nenhum benefício muscular, pode causar uma futura lesão na coluna se repetido muitas vezes.

A segunda observação para uma execução correta do exercício é manter os braços alinhados ao tronco, o que torna o esforço maior para o músculo do bíceps que é o alvo do exercício.

1. O braço direito em repouso se encontra colado ao corpo, paralelo ao ângulo do tronco. O movimento segue até o pulso alcançar o peito, durante todo o tempo a inclinação deve ser idêntica ao do tronco como observado na Figura 5.16 . Logo, existe uma variável que determina um desvio de ângulo aceitável pela inspeção que deve ser determinado por análise de um educador físico.

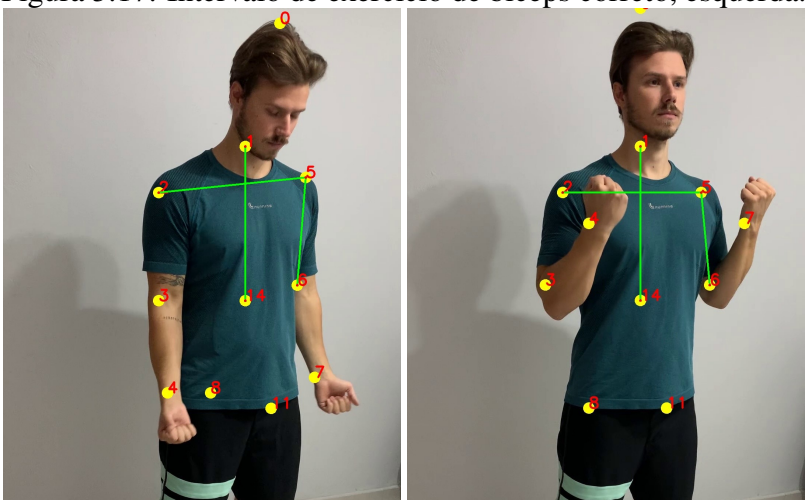
Figura 5.16: Intervalo de exercício de bíceps correto, direita.



Fonte: Autor

2. A lógica de inspeção do braço esquerdo é análoga a do braço direito.

Figura 5.17: Intervalo de exercício de bíceps correto, esquerda.

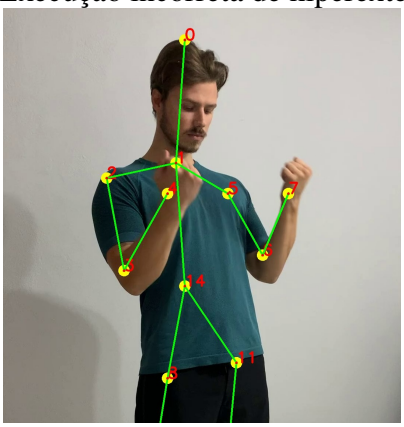


Fonte: Autor

3. A hiperextensão lombar é determinada quando o ângulo do tronco ultrapassa 92

graus. Como é visto na Figura 5.18 onde o corpo foi inclinado e o ângulo alcançado foi 94,24 graus.

Figura 5.18: Execução incorreta de hiperextensão lombar.



Fonte: Autor

5.4 Influência da altura e distância da posição da câmera Elevação Lateral

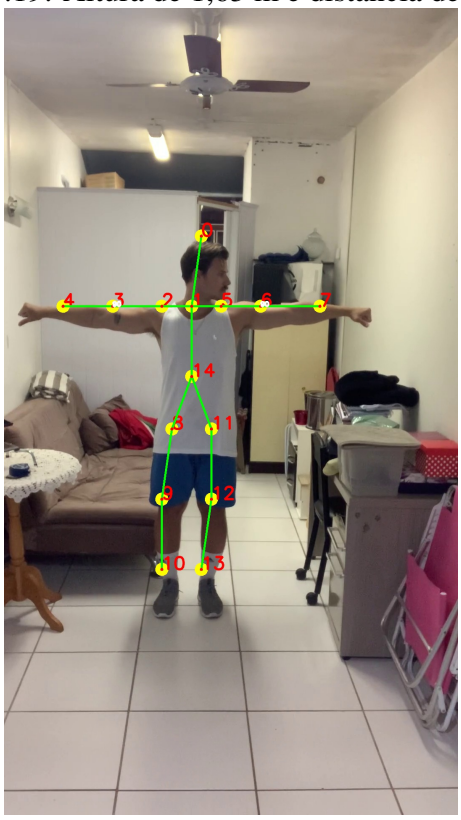
Uma das variáveis do sistema é quais as possíveis alterações no resultado poderiam ser geradas por influência da distância entre a pessoa que está fazendo o exercício e a câmera. Pensando mais uma vez na robustez do sistema e usabilidade, algum usuário poderia necessitar que a câmera ficasse mais perto devido ao pouco espaço que lhe é oferecido ou mais distante devido a envergadura do seu corpo ser maior.

Outra variável importante de ser testada era se a altura que a câmera está posicionada influencia no resultado, devido à necessidade do usuário de posicioná-la conforme sua disponibilidade, seja no chão, seja em um suporte de ventosa por exemplo.

Portanto, o objetivo principal era descobrir se o ângulo gerado entre os braços e o tronco permaneceria o mesmo com ambas as influências. Para isso, no exercício de elevação lateral foram gravados dois vídeos mudando a distância da câmera posicionada em uma altura de 1 metro e 63 centímetros, altura bem alta se considerarmos a altura média dos brasileiros que é de 1,65 metros (LANCET, 2020). E dois vídeos mudando a distância da câmera posicionada no chão. As medições de distância e altura foram feitas usando uma trena e o ângulo dos braços foram acompanhados por um educador físico.

1. Primeiro teste em uma altura de 1 metro e 63 centímetros.

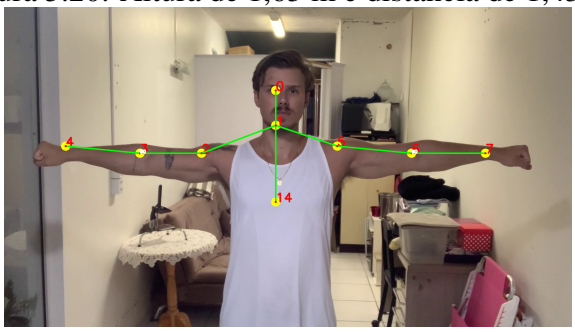
Figura 5.19: Altura de 1,63 m e distância de 3,58 m.



Fonte: O Autor

A Figura 5.19 foi colocada na rede e medida por inspeção analítica a partir do pontos de interesse e o algoritmo provou uma precisão na medição física e apontou exatos 90 graus.

Figura 5.20: Altura de 1,63 m e distância de 1,43 m.



Fonte: O Autor

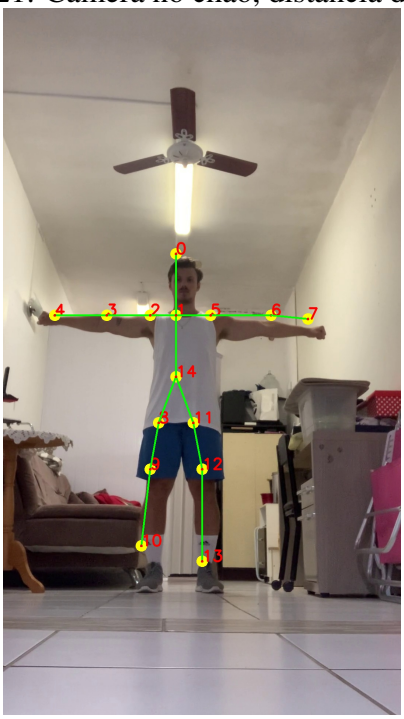
Para a distância de 1,43 metros já foi necessário usar a tela na posição horizontal devido a posição dos braços no exercício. A imagem 5.20 foi processada na POSE-XAU e o ângulo apresentado foi 90 graus, provando que a distância não interfere na medição usada no ângulo dos braços.

A rede fica confusa quando está visível somente metade do corpo, detectando ape-

nas as partes superiores. Entretanto, isso não afeta o resultado do exercício, pois os pontos-chave essenciais são determinados de forma correta, podendo a câmera ser usada nessa posição para esse exercício.

2. Primeiro teste com a câmera posicionada no chão.

Figura 5.21: Câmera no chão, distância de 3,58 m.



Fonte: O Autor

Como esperado o ângulo indicado pela Rede permaneceu 90 graus, mesmo com a distância elevada. Nesse caso observou-se pela grande distância e pela má iluminação, vários erros de detecção em alguns dos *frames*. Portanto, quanto mais distante da câmera pior será a qualidade de definição do corpo no vídeo e maior a chance de um resultado errado.

Figura 5.22: Câmera no chão, distância de 1,43 m.

Fonte: O Autor

Mantendo a câmera no chão e se aproximando dela se observou uma grave alteração

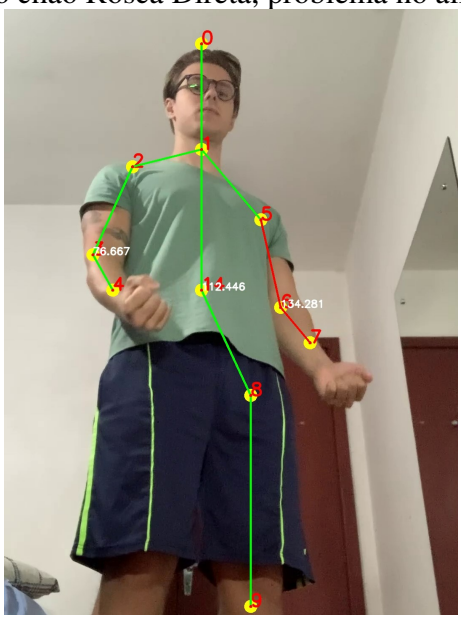
na detecção de pontos-chave. Isso se deve pela câmera apresentar uma angulação muito elevada em relação aos membros superiores, a captura do vídeo e medição da rede ficam severamente prejudicadas. Logo, o usuário é aconselhado a jamais utilizar essa posição de gravação. Conclui-se que a baixa altura da câmera combinada com uma pequena distância gera problemas de identificação.

5.5 Influência da altura da câmera Rosca Direta

Na Seção 5.4 conseguimos mostrar que a distância não causa influência nos ângulos medidos, entretanto a gravação próxima do chão se mostrou problemática.

No exercício de Rosca Direta foi efetuado o mesmo teste com a intenção de descartar erros. Assim como no exercício de elevação lateral, observam-se problemas nessa posição de gravação. Os pontos-chave dos ombros ficaram desalinhados como visto na Figura 5.23, e por isso não geraram dados confiáveis passiveis de inspeção.

Figura 5.23: Câmera no chão Rosca Direta, problema no alinhamento dos ombros.



Fonte: O Autor

Logo, recomenda-se fortemente que na gravação desse exercício o usuário se grave com a câmera em uma altura superior ao quadril.

5.6 Confiabilidade da Rede

Uma confiabilidade alta da Rede OpenPose é essencial para o funcionamento correto da POSEXAU. Considerando que as funções utilizadas na inspeção são determinísticas, se a entrada da POSEXAU se mantiver confiável o resultado será confiável da mesma forma. Visando desconsiderar resultados em que a alimentação tenha sido de baixa confiabilidade a fim de evitar possíveis falsos aprovados, foram testadas diferentes iluminações em diferentes ambientes obtendo uma perspectiva do potencial da rede OpenPose. Imagens de *frames* de diferentes vídeos para testar a confiabilidade da rede são mostrados na Figura 5.24.

Figura 5.24: *Frames* usados para teste de confiabilidade.

Fonte: O Autor

Na Tabela 5.2 é possível ver a confiabilidade dos pontos-chave dos respectivos *frames* da Figura 5.24, confiabilidade média de todos os pontos, confiabilidade média dos pontos-chave alvos do exercício.

Tabela 5.2: Amostras de confiabilidade.

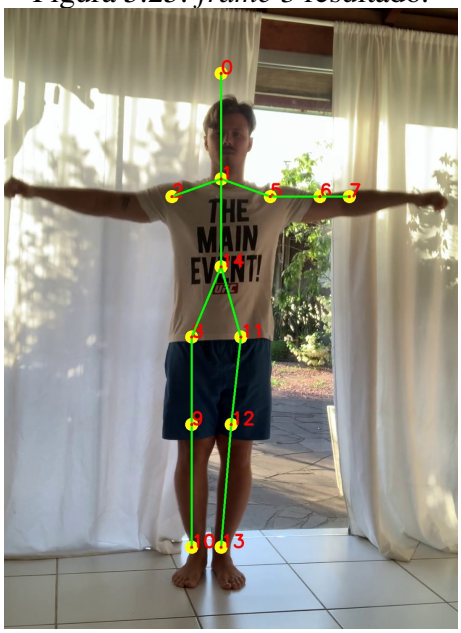
	frame 1	frame 2	frame 3	frame 4	frame 5	frame 6
Cabeça	0,800	0,539	0,692	0,761	0,721	0,691
Pescoço	0,706	0,755	0,763	0,804	0,753	0,793
Ombro Direito	0,695	0,744	0,611	0,707	0,829	0,771
Cotovelo Direito	0,618	0,471	0,163	0,738	0,516	0,704
Pulso Direito	0,718	0,516	0,047	0,462	0,277	0,787
Ombro Esquerdo	0,712	0,705	0,643	0,671	0,764	0,761
Cotovelo Esquerdo	0,524	0,707	0,425	0,691	0,572	0,823
Pulso Esquerdo	0,363	0,705	0,213	0,514	0,368	0,842
Anca Direita	0,469	0,312	0,518	0,551	0,576	0,600
Joelho Direito	0,584	0,134	0,715	0,675	0,335	0,727
Pé Direito	0,759	0,017	0,866	0,700	0,488	0,914
Anca Esquerda	0,558	0,343	0,456	0,581	0,632	0,582
Joelho Direito	0,609	0,134	0,480	0,627	0,437	0,648
Pé Direito	0,731	0,034	0,730	0,813	0,667	0,807
Peito	0,701	0,565	0,669	0,765	0,840	0,743
Total	63,65%	44,55%	53,26%	67,06%	58,51%	74,63%
Interesse	64,86%	63,42%	46,94%	67,91%	62,67%	76,84%

Fazendo esse teste foi possível tirar algumas conclusões a respeito dessa integração:

1. Valores de confiabilidade menores que 0,2 não são mostrados no resultado, conforme parâmetro configurável no código. Como visto na Figura 5.25 e os pontos destacados na Tabela 5.2.
2. Valores de confiabilidade acima de 0,35 podem ser considerados bons.
3. A média total dos pontos não representa bem a confiabilidade necessária para o exercício alvo. Portanto deve-se considerar apenas os pontos de interesse.
4. O *frame* 2 teve uma péssima média total de confiabilidade devido a posição da câmera, o que fortalece a ideia de direcionar os pontos para as peculiaridades do exercício.
5. O *frame* 3 teve uma péssima média de confiabilidade de pontos de interesse, o que se refletiu no resultado mostrado na Figura 5.25.
6. O *frame* 4 que possuía fundo monocromático não foi o que obteve melhor confia-

bilidade, mas sim o *frame* 6.

Figura 5.25: *frame* 3 resultado.



Fonte: O Autor

Analisando a média entre os *frames* que tiveram um resultado passível de inspeção, conclui-se que acima de 60% o vídeo em questão deve ser considerado bom. Qualquer coisa abaixo disso deve ser desconsiderada, alertando o praticante do exercício o motivo de tal.

No fragmento de Código Python 5.4, é possível ver como é calculada a confiabilidade para ambos os exercícios, a função recebe como entrada a probabilidade miníma para validação escolhida pelo usuário e um vetor contendo as confiabilidades dos pontos, onde o índice do vetor representa a parte do corpo mencionada na Tabela 4.1.

```
def confiability_upper(confs,prob_min_lateral_raise):
    Num_Points = 9
    Sum_Confiability = (confs[BODY_PARTS["Head"]] + confs[BODY_PARTS[
                    "Neck"]]+ confs[BODY_PARTS["RShoulder"]]
                    + confs[BODY_PARTS["RElbow"]] + confs[BODY_PARTS[
                    "RWrist"]]+ confs[BODY_PARTS["LShoulder"]]
                    + confs[BODY_PARTS["LElbow"]] + confs[BODY_PARTS["
                    LWrist"]] + confs[BODY_PARTS["Chest"]])
    Confiability = ((Sum_Confiability)/Num_Points)
    return False if (Confiability <= prob_min_lateral_raise) else True
```

5.7 Imprecisões da rede OpenPose

Nessa seção serão mencionados alguns casos observados pelo autor que mostraram problemas na inferência da Rede OpenPose. O primeiro caso que a confiabilidade não detecta e pode gerar problemas no resultado são algumas variações repentinas de pontos de um *frame* para outro como é visto nas Figuras 5.26, 5.27 e 5.28.

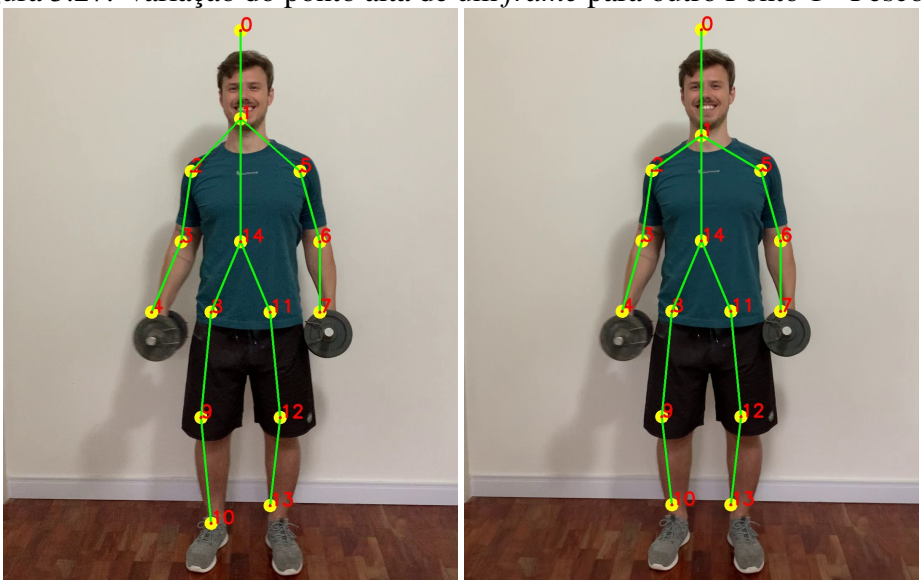
- Ponto zero, que identifica a posição da cabeça, porém não é crítico pelo fato de não ser usado na análise.

Figura 5.26: Variação do ponto alta de um *frame* para outro Ponto 0 - Cabeça.

Fonte: O Autor

- Ponto um, ou pescoço, também acaba por ter uma variação observada, às vezes aparecendo alinhada com os ombros e às vezes mais alta na posição do queixo. Apesar desse ponto ser usado para identificar possíveis câmeras tortas não foi identificado problema nessa variação vista que ocorre no eixo y, não tendo influencia no calculo explicado na Seção 5.1.

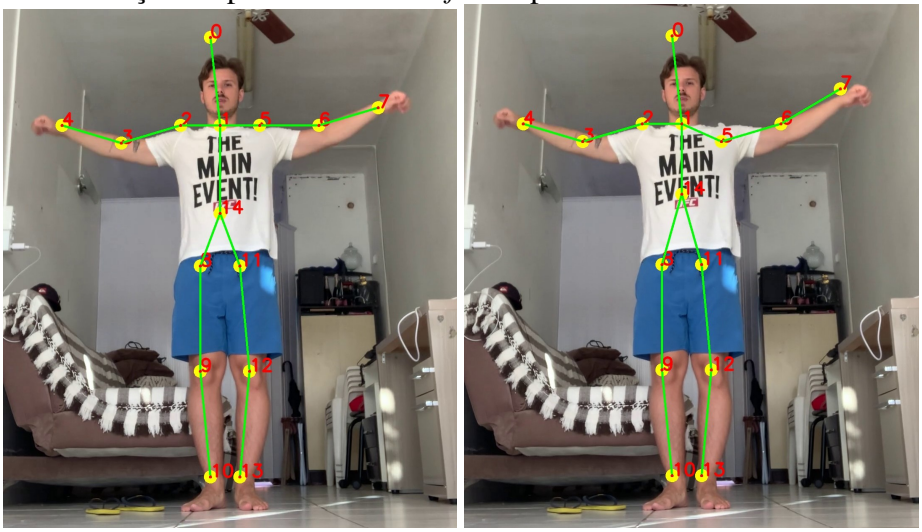
Figura 5.27: Variação do ponto alta de um *frame* para outro Ponto 1 - Pescoço.



Fonte: O Autor

- Um deles é o ponto que identifica a posição dos ombros, em algumas ocasiões pode ser um fator que causa um falso negativo.

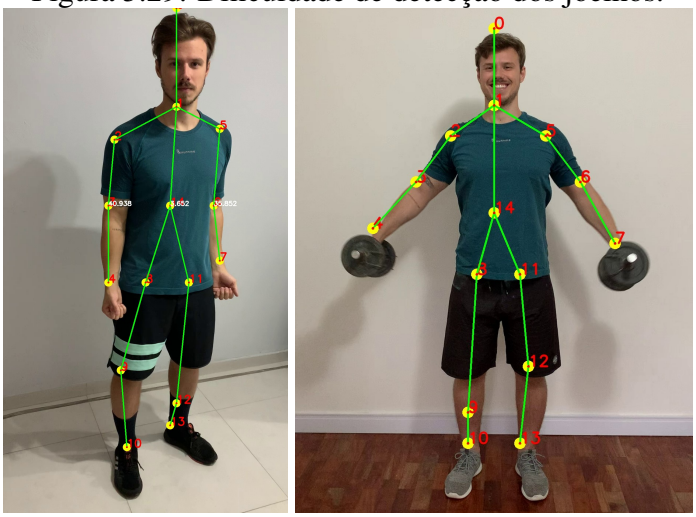
Figura 5.28: Variação do ponto alta de um *frame* para outro Ponto 5 - Ombro Esquerdo.



Fonte: O Autor

- Problemas para detectar os joelhos mostrado na Figura 5.29.

Figura 5.29: Dificuldade de detecção dos joelhos.

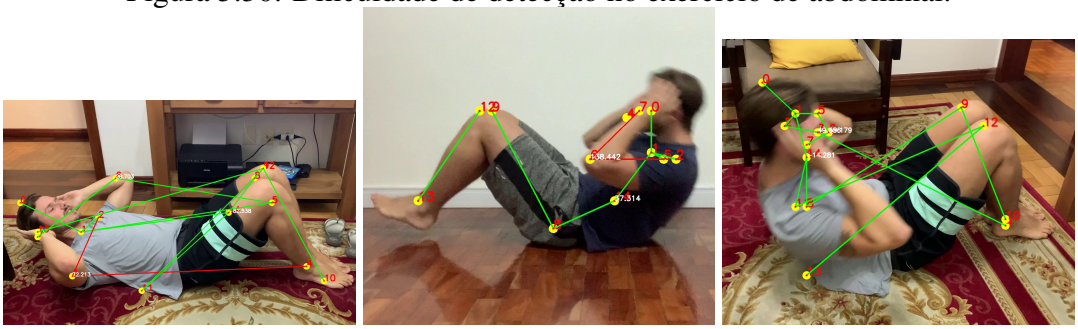


Fonte: O Autor

Alguns exercícios tiveram tentativas de serem adicionados mas fracassaram.

- Exercícios deitado em que o usuário se encontra de lado para a câmera como no exercício de abdominal mostrado na Figura 5.30.

Figura 5.30: Dificuldade de detecção no exercício de abdominal.



Fonte: O Autor

- Exercícios deitado de bariga para baixo em que o usuário se encontra de lado para a câmera como no exercício de flexão de braço mostrado na Figura 5.31.

Figura 5.31: Dificuldade de detecção no exercício de flexão.

Fonte: O Autor

6 ALGORITMOS

A linguagem de programação escolhida foi Python devido a sua vasta aplicabilidade em soluções de Redes Neurais. Essa escolha facilitou os testes iniciais em diferentes tipos de Redes Neurais de Detecção de Posição Corporal. Além disso, com a linguagem Python é possível usar a distribuição Anaconda (ANACONDA, 2020) que será melhor explicada na Seção 6.1.

6.1 Anaconda

Um problema comum enfrentado em algumas linguagens como Python é criar um ambiente de programação idêntico ao da solução que se deseja usar, porque envolve a instalação de muitas bibliotecas e pacotes, que geralmente possuem versões específicas que devem ser respeitadas. No Anaconda esse processo de criação de ambientes se torna muito mais simples, pois além de conseguir criar vários ambientes com versões diferentes sem gerar conflitos e problemas de compatibilidade, a transição entre eles é muito simples.

Com ele também é possível importar um ambiente já criado de maneira rápida, possibilitando que pessoas reproduzam a sua solução sem muito esforço. Para a criação de um ambiente compatível com a solução proposta na POSEXAU basta seguir os seguintes passos:

- *1*. Baixar o Anaconda gratuitamente no site oficial (ANACONDA, 2020).
- *2*. Criar uma ambiente de programação com python 3.7.11
  conda create –name POSEXAU python=3.7.11
- *3*. Instalar os pacotes dependentes:
  conda install -c anaconda jupyter
  conda install -c conda-forge opencv=4.5.3
  conda install -c anaconda numpy=1.21.2
  conda install -c jmcmurray os=0.1.4
  pip install import-ipynb=0.1.3

6.2 Arquitetura da POSEXAU

O código completo está disponível no meu Github pessoal (LAUCK, 2022). O corpo principal pode ser encontrado com o nome OpenPose, nele está contida a parte de inicialização de parâmetros, detecção dos pontos, criação do vídeo final e chamada para a função do POSEXAU.

Dentro do módulo POSEXAU tem duas funções, uma de inspeção se o exercício está correto ou não e dentro dela ocorre uma chamada para a função que verifica se a confiabilidade escolhida pelo usuário é respeitada e se pode ser validado o exercício.

O módulo principal da POSEXAU carrega dois módulos externos, lateral_raise que é responsável por fazer as manipulações matemáticas necessárias para tomar a decisão da corretude do exercício de elevação lateral, barbell_curl também responsável por verificar a corretude do exercício de rosca direta, e print_result responsável por desenhar os resultados no *frame*.

Até o momento a POSEXAU possui duas funções principais que são responsáveis por fazer a inspeção, verificação de confiabilidade e desenho na imagem resultado. Ambas possuem seis parâmetros de entrada.

"posexau.inspec_lateral_raise"

1. O primeiro é o vetor com a confiabilidade dos pontos onde o seus índice segue as convenções da Tabela 4.1.
2. Confiabilidade miníma aceitável escolhida pelo usuário.
3. *Frame* do vídeo.
4. Vetor contendo os pontos do corpo onde o índice segue a Tabela 4.1.
5. Ângulo máximo entre o tronco e o braço escolhido pelo usuário.
6. Parâmetro booliano que define se os ângulos do corpo e braços serão desenhados na imagem ou não.

"posexau.inspec_barbell_curl"

1. O primeiro é o vetor com a confiabilidade dos pontos onde o seus índice segue as convenções da Tabela 4.1.
2. Confiabilidade miníma aceitável escolhida pelo usuário.
3. *Frame* do vídeo.
4. Vetor contendo os pontos do corpo onde o índice segue a Tabela 4.1.

5. Ângulo máximo de inclinação do corpo aceitável.
6. Margem aceitável no ângulo dos braços para reprovar o movimento.
7. Parâmetro booliano que define se os ângulos do corpo e braços serão desenhados na imagem ou não.

6.3 Adicionando novos exercícios.

A adição de novos exercícios é essencial para o sucesso e progressão da ferramenta POSEXAU. Porém essa não é uma tarefa simples, nessa seção serão discutidos os passos que foram tomados tanto para se chegar ao sucesso da adição do execício de Elevação Lateral e Rosca Direta, como da falha na tentativa de adicionar alguns exercícios que infelizmente ainda não são possíveis de ser programados devido as condições oferecidas pelas redes de detecção corporal 2D.

Primeiro, a adição de qualquer exercício começa pela tarefa exaustiva de testar a gravação de diferentes ângulos de execução, distâncias e alturas. Deve ser levado em questão a rede atual que se está usando, e se ela é capaz de gerar resultados confiáveis levando em conta a preservação da saúde do praticante.

Pontos positivos dessa etapa:

- *1*. A escolha da Rede pode ser feita pelo usuário.
- *2*. Há uma quantidade grande de exercícios que podem ser testados.
- *3*. A câmera móvel permite criatividade e maior possibilidade de sucesso.
- *4*. Prática de exercícios durante o processo.
- *5*. Gerar tecnologia para auxiliar na saúde das pessoas.

Pontos negativos:

- *1*. Exercícios que obstruem parte do corpo ainda são um desafio para as redes atuais.
- *2*. Pode ser um pouco decepcionante após a gravação de vários vídeos tomar a decisão de desclassificar o exercício.
- *3*. A sua câmera pode determinar uma parte do seu sucesso nessa etapa.

Se provado que a rede tem robustez suficiente para gerar bons resultados, ainda que com limitação de posição de câmera, se parte para os testes seguintes, agora envolvendo os resultados.

Nessa etapa é onde se encontra a criatividade do programador, quais informações

são passíveis de serem retiradas do resultado gerado, ângulos, distâncias, proporções. O resultado gerado nada mais é que pontos em gráfico 2D e o seu objetivo e retirar informações disso.

Aplicada suas regras matemáticas para aquele exercício será possível testá-la nos vídeos gravados previamente. Revisite os vídeos agora gerando resultados de inspeção dos exercícios, corrija os problemas até obter resultados concisos.

Por fim, determine uma função para desenhar os pontos de forma a destacar com vermelho as partes do corpo onde o exercício está sendo feito errado e gere mensagens de erro, é possível se inspirar nas já implementadas. Qualquer colaboração no Github (LAUCK, 2022) será de suma importância para o sucesso da ferramenta POSEXAU.

7 CONCLUSÃO

Por muito tempo a inspeção analítica fez coisas incríveis e ajudou no desenvolvimento de muitas tecnologias inovadoras. Depois que seu sucesso já havia se consolidado as Redes Neurais começaram a trilhar o seu caminho para o sucesso, muitos problemas simples de ser implementados com inspeção analíticas foram falhos quando tentados resolver com Inteligência Artificial, assim como a Inteligência Artificial resolveu muitos problemas que a inspeção analítica não resolvia, gerando novas possibilidades. Nesse trabalho vemos que a combinação de ambos se mostra muito promissora, a inteligência artificial é capaz de gerar entradas que seriam impossíveis de serem geradas com inspeção analítica. Por outro lado, a inspeção analítica é determinística e necessita de pouco processamento. O determinismo da inspeção, combinado com a confiabilidade da rede, produz resultados surpreendentes.

Um ponto positivo da POSEXAU é não depender do uso exclusivo de um sistema de detecção de pose corporal. Portanto, qualquer Rede Neural que tenha como saída os pontos-chave utilizados para verificação do exercício podem ser usadas, se assim surgir uma rede mais precisa ou por necessidade da aplicação uma rede com menor processamento.

É sabido que o uso da POSEXAU ainda é pouco atrativo para pessoas que querem utilizá-lo apenas para corrigir os exercícios, devido à necessidade de ter que usar uma máquina com um bom poder computacional e ter noções básicas do funcionamento de Python, Anaconda e Jupyter Notebook. O ideal seria abstrair a parte de códigos e tornar a aplicação uma caixa preta para pessoas interessadas somente no seu uso e não desenvolvimento.

Apesar das inspeções feitas nos exercícios englobarem uma parte dos erros, ainda existem alguns casos em que uma execução incorreta pode ser apontada como correta. Existe conhecimento desses casos pelo autor, e portanto deve-se aumentar o número de inspeções a fim de evitar esses casos. Portanto, os exercícios adicionados no OpenPose ainda devem ser retrabalhados a fim de torná-los o mais confiável possível.

Foi constatado que grande parte dos educadores físicos já aderiram ao acompanhamento a distância (UNICESUMAR, 2022) e alguns fazem correções dos exercícios enviados pelos seus alunos em *tablets* e *smarphones* desenhando sobre o corpo do mesmo. Porém, por ser um processo muito mecânico, acaba por tomar bastante tempo dos profissionais que reflete no custo do serviço para os usuários. A ferramenta otimizaria esse

processo, tornado o trabalho mais rápido para o educador físico e permitindo uma diminuição no custo de horas.

Aplicativos como Freeletics Bodyweight (FREELETICS..., 2022) são famosos por oferecer treinos prontos e acompanhamento de alguns profissionais que lhe auxiliam nessa jornada. Demonstrando que o público alvo não possui mais tanta resistência ao acompanhamento à distância como antigamente.

Observa-se que em muitos exercícios ainda é impossível de se obter resultados satisfatórios da detecção da pose humana pelo OpenPose, isso limitou muito a escalabilidade desse trabalho. O exercício de elevação lateral, foi o exercício mais robusto que se encontrou para iniciar as pesquisas. Grande parte do seu sucesso se deve pelo fato da posição do exercício beneficiar a visão clara de todos os membros e por ter grandes semelhanças com a maioria dos casos usados para treinamento da rede. Outros exercícios como flexão de braço e abdominal, se mostraram muito ruins pelo fato da posição do exercício não gerar boas saídas na rede OpenPose. O problema poderia ser contornado utilizando outras redes de detecção capazes de gerar bons resultados de saída em posições de lado e deitado. O exercício de Rosca Direta apresentou bons resultados apesar do usuário precisar ficar em posição diagonal para câmera, o que pode gerar mais instabilidade nas gravações.

No entanto o problema de posicionamento não invalida a solução, uma rede como OpenPose não é treinada visando esse tipo específico de detecção envolvendo exercícios. Alguns testes poderiam ser feitos usando um treinamento com dados de pessoas fazendo exercícios, redefinindo os pesos da Rede e chegando em resultado com melhores detecções.

8 TRABALHOS FUTUROS

Eu acredito que o primeiro passo seja difundir a ferramenta, aumentar a quantidade de exercícios possíveis de serem corrigidos pela POSEXAU, e a quantidade de inspeções feitas nos exercícios já adicionados para evitar outros erros. Com isso já será possível alimentar a rede com vídeos de pessoas que reproduziram o ambiente de programação em suas máquinas, podendo assim ter mais amostras em diferentes câmeras e uma validação concreta de todas elas por um educador físico capacitado.

Um passo um pouco maior, mas que foi a ideia inicial do projeto, seria criar uma plataforma mobile para tornar a ferramenta acessível ao público em geral. O autor esteve em contato com alguns educadores físicos que prestam serviço a distância e muitos se mostraram a favor da tecnologia para auxiliá-los em suas aulas. Muitas são as vantagens para os profissionais da área, pois além de tornar o alcance maior da prática esportiva, poderia tornar mais acessível o acompanhamento periódico de treino feito em casa visto que os educadores teriam a ajuda da Inteligência Artificial para corrigir possíveis erros de execução, aumentando sua capacidade de atendimento e sua qualidade.

Para isso seria necessário criar um aplicativo inicialmente na Plataforma Android. Esse aplicativo teria uma interface intuitiva onde o usuário seria capaz de anexar o vídeo praticando o exercício, depois de enviado, o exercício seria processado em uma máquina virtual na nuvem, em um servidor local, ou até mesmo ser processado pelo próprio *smartphone* dependendo do seu poder de processamento, uma vez que só seriam feitas inferências. Depois disso, seria gerada a realimentação para o usuário do vídeo contendo informação de possíveis erros na execução.

Pensando em melhorar a robustez da detecção da pose humana, e aumentar o número de exercícios que a POSEXAU é capaz de processar, seria um trabalho muito interessante fazer testes em redes já consolidadas usando uma base de dados própria. Para isso seria necessário pessoas dispostas a gravarem vídeos executando os exercícios e redefinir os pesos usados na Rede.

REFERÊNCIAS

ANACONDA. 2020. Available from Internet: <https://www.anaconda.com/>.

ANG, C. The growth of home fitness apps. *Visual Capitalist*, v. 1, n. 1, p. 1, 2021.

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

FANG, H.-S. et al. RMPE: Regional multi-person pose estimation. In: *ICCV*. [S.l.: s.n.], 2017.

FREELETICS Bodyweight. 2022. Available from Internet: <https://www.freeletics.com/pt/>.

GONZALEZ, R. E. W. R. C. *Digital Image Processing*. 2nd ed. ed. [S.l.]: Prentice Hall, 2002. ISBN 9780201180756,0201180758,0130946508,9780130946508.

KENDALL, A.; GRIMES, M.; CIPOLLA, R. *PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization*. 2016.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F. et al. (Ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2012. v. 25. Available from Internet: <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.

LANCET, T. *Height and body-mass index trajectories of school-aged children and adolescents from 1985 to 2019 in 200 countries and territories: a pooled analysis of 2181 population-based studies with 65 million participants*. 2020. Available from Internet: <https://www.thelancet.com/journals/lancet/article/PIIS0140-6736(20)31859-6/fulltext>.

LAUCK, L. *Repositório POSEXAU*. 2022. Available from Internet: <https://github.com/LucasLauckP/POSEXAU>.

LECUN, Y. B. Y. *Pattern Recognition and Neural Networks*. [S.l.: s.n.], 1995.

NETO, M. C. da S. Ensinando cinemática através da análise de movimentos em vídeos de captura de games. *UFF*, 2016.

OPENCV site oficial. 2022. Available from Internet: <https://opencv.org/>.

SHI, S. et al. Benchmarking state-of-the-art deep learning software tools. *CoRR*, abs/1608.07249, 2016. Available from Internet: <http://arxiv.org/abs/1608.07249>.

ULJAK, I. et al. A brief introduction to opencv. *2012 Proceedings of the 35th International Convention MIPRO*, p. 1725–1730, 2012.

UNICESUMAR. *Pandemia faz com que cresça a demanda de educadores físicos*. 2022. Available from Internet: <https://g1.globo.com/mg/centro-oeste/especial-publicitario/unicesumar/educacao-a-distancia/noticia/2021/03/17/pandemia-faz-com-que-cresca-a-demanda-de-educadores-fisicos.ghtml>.

WALIA, M. S. A comprehensive guide on human pose estimation. *Data Science Blogathon*, 2022.