

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**JOÃO PEDRO MENEGAS ZANKOSKI**

**VISÃO COMPUTACIONAL PARA AUXÍLIO NA AVALIAÇÃO DAS  
COMPRESSÕES TORÁCICAS DURANTE A RCP**

**PATO BRANCO**

**2025**

**JOÃO PEDRO MENEGAS ZANKOSKI**

**VISÃO COMPUTACIONAL PARA AUXÍLIO NA AVALIAÇÃO DAS  
COMPRESSÕES TORÁCICAS DURANTE A RCP**

**Computer Vision to Support the Evaluation of Chest Compressions during  
CPR**

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia de Computação do Curso de Bacharelado em Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Rubens Andreas Sautter

**PATO BRANCO**

**2025**



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**JOÃO PEDRO MENEGAS ZANKOSKI**

**VISÃO COMPUTACIONAL PARA AUXÍLIO NA AVALIAÇÃO DAS  
COMPRESSÕES TORÁCICAS DURANTE A RCP**

Trabalho de Conclusão de Curso de Graduação  
apresentado como requisito para obtenção  
do título de Bacharel em Engenharia de  
Computação do Curso de Bacharelado em  
Engenharia de Computação da Universidade  
Tecnológica Federal do Paraná.

Data de aprovação: 24/Junho/2025

---

Rubens Andreas Sautter  
Doutor  
Universidade Tecnológica Federal do Paraná

---

Jefferson Tales Oliva  
Doutor  
Universidade Tecnológica Federal do Paraná

---

Luciene De Oliveira Marin  
Doutora  
Universidade Tecnológica Federal do Paraná

**PATO BRANCO  
2025**

## RESUMO

As doenças cardiovasculares permanecem entre as principais causas de mortalidade no Brasil, o que ressalta a relevância da Reanimação Cardiopulmonar (RCP) como intervenção crítica em situações de emergência. Nesse contexto, a capacitação adequada de profissionais e leigos torna-se essencial para o aumento das taxas de sobrevivência. Este trabalho propõe o desenvolvimento de uma ferramenta computacional voltada ao apoio no ensino e na avaliação prática da RCP. A abordagem baseia-se na análise automatizada de vídeos de treinamentos, com o objetivo de identificar falhas na execução do procedimento, em especial no que se refere à frequência das compressões torácicas. Essa análise permite a geração de *feedbacks* objetivos e sistemáticos, facilitando intervenções pedagógicas mais precisas por parte dos instrutores. Adicionalmente, busca-se implementar um sistema de classificação capaz de distinguir, com acurácia, execuções corretas e incorretas da RCP. O principal desafio deste trabalho é o número limitado de vídeos disponíveis, o que é abordado por meio de técnicas de aumento de dados, visando à generalização e robustez do modelo proposto. Para tanto, foram avaliados 16 vídeos de treinamento com execuções corretas e incorretas, os pontos-chave corporais foram extraídos com MediaPipe e YOLO, transformados em séries temporais, analisados em termos de periodicidade por meio da *Short-Time Fourier Transform* (STFT) e, em seguida, empregados no treinamento de uma rede *Long short-term memory* (LSTM) otimizada via Optuna. Nos testes finais o classificador alcançou, no caso de maior sucesso, F1-score de 0,95 via YOLO e 0,91 via MediaPipe, com precisão e recall médios superiores a 0,89. A expectativa é que os resultados obtidos contribuam para a melhoria dos processos de ensino-aprendizagem em RCP, com potencial impacto na formação de indivíduos mais bem preparados para atuar em emergências cardiovasculares.

**Palavras-chave:** reanimação cardiopulmonar; aprendizado de máquina; visão computacional.

## ABSTRACT

Cardiovascular diseases remain among the leading causes of mortality in Brazil, underscoring the importance of Cardiopulmonary Resuscitation (CPR) as a critical emergency intervention. In this context, the proper training of both healthcare professionals and laypersons is essential for increasing survival rates. This work proposes the development of a computational tool designed to support the teaching and practical assessment of CPR. The approach is based on the automated analysis of training videos, with the aim of identifying execution errors—particularly those related to chest compression rate. Such analysis enables the generation of objective, systematic feedback, thereby facilitating more precise pedagogical interventions by instructors. Additionally, we seek to implement a classification system capable of accurately distinguishing between correct and incorrect CPR performances. The main challenge of this study is the limited number of available videos, which we address through data augmentation techniques to enhance the proposed model's generalization and robustness. We expect that the results will contribute to the improvement of CPR teaching–learning processes, with the potential to better prepare individuals for responding to cardiovascular emergencies. To this end, 16 training videos containing both correct and incorrect executions were captured; body keypoints were extracted with MediaPipe and YOLO, converted into time-series data, analysed for periodicity via the Short-Time Fourier Transform (STFT), and subsequently used to train a Long Short-Term Memory (LSTM) network optimised with Optuna. In the final tests, the classifier achieved its best performance with an F1-score of 0.95 using YOLO and 0.91 using MediaPipe, with mean precision and recall values exceeding 0.89.

**Keywords:** cardiopulmonary resuscitation; machine learning; computer vision.

## LISTA DE FIGURAS

<b>Figura 1 – Arquitetura básica de uma célula LSTM . . . . .</b>	<b>18</b>
<b>Figura 2 – Exemplo de Matriz de confusão . . . . .</b>	<b>19</b>
<b>Figura 3 – Frame de exemplo dos vídeos capturados . . . . .</b>	<b>25</b>
<b>Figura 4 – Fluxograma do Sistema de Avaliação e Assistência de RCP . . . . .</b>	<b>26</b>
<b>Figura 5 – Exemplo de série temporal gerada a partir do vetor de coordenadas extraído de um video de RCP . . . . .</b>	<b>27</b>
<b>Figura 6 – Espectrograma do movimento completo e Série temporal (Execução Correta) . . . . .</b>	<b>34</b>
<b>Figura 7 – Espectrograma do movimento completo e Séries temporal (Execução Incorreta) . . . . .</b>	<b>34</b>
<b>Figura 8 – Matrizes de confusão para MediaPipe e YOLO em diferentes janelas de entrada (128, 256 e 512 quadros) . . . . .</b>	<b>36</b>

## LISTA DE ABREVIATURAS E SIGLAS

### Siglas

AHA	<i>American Heart Association</i>
CNN	<i>Rede Neural Convolucional</i>
FPS	<i>Frames por segundo</i>
LSTM	<i>Long short-term memory</i>
RCP	Reanimação Cardiopulmonar
SMOTE	<i>Synthetic Minority Over-sampling Technique</i>
STFT	<i>Short-Time Fourier Transform</i>
YOLO	<i>You Only Look Once</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>8</b>
1.1	Objetivos	9
1.2	Objetivos Específicos	9
1.3	Estrutura do Trabalho	10
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>11</b>
2.1	Trabalhos Relacionados	11
2.2	Métodos de Ensino de RCP e Seus Desafios	12
2.3	Visão Computacional para Educação em Saúde	12
2.3.1	MediaPipe	12
2.3.2	YOLO	13
2.4	Técnicas de Aumento de dados e Balanceamento de Dados	13
2.4.1	SMOTE	14
2.4.2	Surrogate	14
2.5	Processamento de Sinais para Análise de Frequências	15
2.5.1	<i>Short-Time Fourier Transform</i> (STFT)	16
2.6	Redes Neurais para Classificação de Sequências	16
2.6.1	Redes Neurais Recorrentes e LSTM	17
2.7	Métricas de Avaliação e Validação	19
2.7.1	Matriz de Confusão	19
2.7.2	Acurácia	20
2.7.3	Precisão e Revocação	20
2.7.4	F1-Score	20
2.7.5	Validação Cruzada	21
2.8	Otimização de Hiperparâmetros	21
2.8.1	Optuna	22
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>24</b>
3.1	Materiais	24
3.1.1	Base de Dados	24
3.1.2	Linguagens de Programação e Bibliotecas	25
3.1.3	Ambiente de Desenvolvimento	25

<b>3.2</b>	<b>Métodos</b>	<b>26</b>
3.2.1	Preparação dos Dados	26
3.2.2	Extração de Características	26
3.2.3	Aumento de Dados	28
3.2.4	Análise de Periodicidade	28
3.2.5	Criação e Treinamento do Modelo	29
3.2.6	Validação e Testes	30
<b>4</b>	<b>RESULTADOS</b>	<b>31</b>
4.1	Escopo do sistema	31
4.2	Extração de Características	31
4.3	Análise de Frequência	33
4.4	Aumento e Balanceamento de dados	35
4.5	Treino Final e Avaliação no Conjunto de Teste	35
<b>5</b>	<b>CONCLUSÃO</b>	<b>37</b>
	<b>REFERÊNCIAS</b>	<b>38</b>

## 1 INTRODUÇÃO

As doenças cardiovasculares representam uma das principais causas de mortalidade global, inclusive no Brasil (Oliveira *et al.*, 2024). Segundo a *American Heart Association* (AHA), as doenças cardiovasculares são responsáveis por aproximadamente 19 milhões de mortes anuais (Tsao *et al.*, 2022). Em geral, essas doenças estão associadas ao estilo de vida contemporâneo, incluindo fatores como alimentação inadequada, sedentarismo e baixa qualidade do sono. No contexto de emergências cardiovasculares, o tempo é um elemento crucial para a sobrevivência durante procedimentos de RCP (Feneley *et al.*, 1988). Apesar da importância desse procedimento, as estatísticas mais recentes mostram que o número de RCP realizados está muito abaixo do ideal (Martin *et al.*, 2024). Nesse sentido, o desenvolvimento de ferramentas que auxiliam no ensino e disseminação das técnicas de RCP é essencial. Este trabalho tem como objetivo o desenvolvimento de uma ferramenta computacional para apoiar o ensino do procedimento de RCP.

A pesquisa é desenvolvida no âmbito do projeto PIPE-FAPESP nº 2021-15114-8, o objetivo central do projeto é a criação de uma plataforma de ensino a distância voltada para a capacitação em diferentes tipos de procedimentos, incluindo práticas da área médica. Como diferencial, a plataforma busca integrar ferramentas baseadas em Inteligência Artificial (IA), com o propósito de auxiliar na avaliação automática desses procedimentos. Neste contexto, o presente trabalho contribui com o desenvolvimento de uma ferramenta computacional voltada à classificação e à avaliação de execuções de RCP. Os dados utilizados foram coletados e disponibilizados pela empresa Hipocampus, principal entidade envolvida no projeto.

Embora o uso de métodos computacionais no ensino seja objeto de debate (Ghanbaryan; Saeidi, 2023; Vaona A; Moja, 2018; Stephan *et al.*, 2018), o uso de ferramentas computacionais avançadas para o ensino é raro na área. Em geral, a comparação é realizada entre vídeos educativos e o ensino com um instrutor e equipamentos especializados, e esses estudos indicam que o ensino assistido apenas por vídeos apresenta eficácia inferior quando comparado ao treinamento com manequins (Ghanbaryan; Saeidi, 2023). Entretanto, nas últimas duas décadas, o avanço de ferramentas de reconhecimento de poses trouxe novas possibilidades para o ensino de procedimentos complexos, como a RCP. A detecção de poses exigia equipamentos especializados, como dispositivos com sensores infravermelhos, por exemplo o *Kinect* (Obdržálek *et al.*, 2012), dificultando o acesso e uso desta abordagem. Contudo, com a utilização de técnicas computacionais, como *Rede Neural Convolucional* (CNN), torna-se viável a realização da detecção de poses em imagens e vídeos capturados por câmeras comuns, ampliando o alcance e acessibilidade dessas ferramentas. Atualmente, existem diversas bibliotecas de reconhecimento de pose abertas ao público em geral, como o *MediaPipe* e *You Only Look Once* (YOLO) (Debalaxmi; Vishwakarma; Ranga, 2024).

Neste contexto, a adoção de técnicas de análise mais aprofundadas é fundamental para fornecer um *feedback* detalhado e confiável aos alunos e profissionais que buscam aprimorar

suas práticas de RCP. Por meio de algoritmos capazes de avaliar a execução das compressões torácicas, torna-se possível identificar ajustes que poderiam passar despercebidos em um simples treinamento por vídeo (Li *et al.*, 2024). Nesse sentido, a análise computacional dos movimentos não apenas apoia o processo de aprendizagem, mas também tende a ampliar a segurança na execução dos procedimentos, ao permitir uma correção mais rápida e objetiva das falhas, buscando o aumento das taxas de sobrevivência em situações de emergência.

Do ponto de vista computacional, o presente trabalho fundamenta-se na captura de dados em vídeo e na subsequente extração de poses corporais por meio de bibliotecas de visão computacional e na aplicação de técnicas de aprendizado de máquina para classificar a qualidade da execução do procedimento. A abordagem proposta combina a detecção de poses com a análise de séries temporais por meio de redes neurais recorrentes, ilustrando a sinergia entre reconhecimento de imagem e processamento de sinais temporais (Luo *et al.*, 2018). O sistema desenvolvido neste trabalho avalia automaticamente o desempenho de operadores durante a realização da RCP a partir de vídeos, de forma precisa e acessível, sem a necessidade de sensores dedicados ou equipamentos especializados.

## 1.1 Objetivos

Desenvolver uma ferramenta de visão computacional para a avaliação automática da qualidade da RCP realizada em vídeos, que fornecerá informações visuais e quantitativas por meio de gráficos de frequência e de movimento.

## 1.2 Objetivos Específicos

- Analisar a periodicidade dos movimentos durante as compressões torácicas para verificar se a técnica está correta, possibilitando a monitoramento do movimento ao longo do tempo.
- Classificar a qualidade da técnica de RCP em correto e incorreto, buscando diferenciar as execuções.
- Realizar medidas de qualidade baseadas em padrões profissionais para serem usadas em avaliações e certificações, de acordo com as diretrizes da AHA (Merchant *et al.*, 2020).
- Avançar na aplicação de tecnologias de aprendizado de máquina e visão computacional na área médica.

### 1.3 Estrutura do Trabalho

O presente trabalho está organizado em seis capítulos, conforme descrito a seguir:

Capítulo 1: Introdução – Apresenta as considerações iniciais, os objetivos do estudo, a justificativa para a realização da pesquisa e a organização do trabalho.

Capítulo 2: Referencial Teórico – Discute os conceitos fundamentais relacionados a Reanimação Cardiopulmonar (RCP), visão computacional e técnicas de aprendizado de máquina.

Capítulo 3: Trabalhos Relacionados – Analisa estudos anteriores e pesquisas similares, destacando as contribuições e lacunas existentes que justificam a realização deste trabalho.

Capítulo 4: Materiais e Métodos – Descreve os materiais utilizados e os métodos adotados para o desenvolvimento da ferramenta, incluindo a coleta e preparação dos dados, técnicas de aumento de dados e balanceamento, extração de características, criação e treinamento do modelo, além das estratégias de validação e testes.

Capítulo 5: Resultados – Apresenta os resultados obtidos a partir da aplicação das técnicas e métodos descritos na metodologia, incluindo análises das métricas de desempenho do modelo e a eficácia da ferramenta desenvolvida.

Capítulo 6: Conclusão – Resume os principais achados do estudo, discute suas implicações, destaca as contribuições para a área de saúde e sugere direções para pesquisas futuras.

## 2 REFERENCIAL TEÓRICO

### 2.1 Trabalhos Relacionados

Esta seção apresenta os principais trabalhos relacionados ao uso de técnicas de captura de pose e redes neurais para a avaliação de movimentos humanos, com ênfase em aplicações voltadas ao treinamento e análise da RCP. A seguir, discutem-se os métodos utilizados, os recursos tecnológicos empregados e os resultados obtidos, sintetizados na Tabela 1.

A aplicação de técnicas de visão computacional para análise de movimento humano evoluiu de sistemas baseados em sensores especializados para soluções puramente baseadas em vídeo. Cao *et al.* (2017) introduziram o *OpenPose*, uma das primeiras bibliotecas de estimativa de pose em tempo real a partir de câmeras comuns, capaz de extrair coordenadas articulares com boa precisão e baixo custo computacional. Logo em seguida, Lugaresi *et al.* (2019) lançaram o *MediaPipe Pose*, otimizado para dispositivos móveis, ampliando ainda mais o acesso a ferramentas de análise de movimento em aplicações de saúde e esporte.

Em aplicações específicas à reanimação cardiopulmonar, Xie *et al.* (2020) empregaram o sensor de profundidade *Kinect* para capturar o esqueleto tridimensional do profissional e calcular métricas como profundidade e frequência das compressões torácicas. Seu sistema demonstrou que, mesmo com limitações de hardware, era possível fornecer *feedback* automático durante o treinamento de RCP sem a necessidade de manequins especiais.

Com o avanço das redes neurais convolucionais (CNN), Huang *et al.* (2024) propuseram o *SmartCPR*, um *framework* que utiliza CNNs acopladas a um modelo de estimativa de pose para avaliar compressões apenas por vídeo de smartphone. Esse trabalho apresentou resultados promissores de classificação em tempo real do desempenho em RCP, incluindo indicadores de ritmo e postura, sem depender de sensores externos.

Para além das métricas físicas, Wang *et al.* (2024) desenvolveram o *CPR-Coach*, que utiliza abordagens de detecção de ação em vídeo para reconhecer tipos específicos de erro (braços dobrados, compressões assimétricas, posição incorreta das mãos) em um conjunto de dados anotados manualmente *CPR-Coach dataset*. Esse foi o primeiro estudo a empregar visão computacional para análise granular de falhas durante a RCP, fornecendo *feedback* detalhado aos aprendizes.

O presente trabalho concentra-se em avaliar a frequência e a regularidade das compressões, usando apenas a série temporal das coordenadas articulares extraídas por bibliotecas de pose em vídeo, sem recorrer a sensores de profundidade ou à análise espacial detalhada do posicionamento das mãos. Em suma, enquanto outros trabalhos buscam a correção postural ou a detecção de erros específicos, nosso sistema foca na estimativa de ritmo entre 100 e 120 compressões por minutos (CPM) servindo como complemento objetivo a ensinamentos tradicionais.

**Tabela 1 – Comparação dos principais trabalhos relacionados à avaliação de RCP por visão computacional**

Estudo (ano)	Aquisição / Dispositivo	Técnica principal	Métrica-alvo	Desempenho reportado
Cao <i>et al.</i> (2017)	Câmera RGB	<i>Part Affinity Fields</i> (CNN, bottom-up)	Estimativa geral de pose	primeiro lugar no COCO 2016 <i>Key-points challenge</i>
Xie <i>et al.</i> (2020)	Kinect	Esqueleto 3D + ajuste senoidal	Frequência e profundidade	Erro mediano $\pm$ 2,9 cpm; variação 4,1 %
Huang <i>et al.</i> (2024)	Smartphone	<i>MoveNet</i> + heurísticas	Profundidade, frequência e postura	Erro Médio 0,33 cm (profundidade), acurácia global 84,2 %
Wang <i>et al.</i> (2024)	4 câmeras 4K	CPR-Coach dataset + <i>ImagineNet</i>	13 erros simples e 74 compostos	Precisão média 0,705
Este trabalho (2025)	Câmera comum	YOLO/MediaPipe	Frequência e movimento geral de partes do corpo	F1-score 0,95 (YOLO) / 0,91 (MediaPipe); precisão e revocação > 0,89

## 2.2 Métodos de Ensino de RCP e Seus Desafios

O ensino da RCP tradicionalmente envolve a combinação de instruções teóricas, demonstrações práticas e treinamentos com manequins. Essa abordagem prática permite aos aprendizes executar compressões torácicas sob supervisão direta, recebendo *feedback* imediato sobre a qualidade das compressões. No entanto, estudos indicam que métodos baseados exclusivamente em vídeo ou autoaprendizagem apresentam eficácia inferior devido à falta de interatividade e correção em tempo real (Ghanbaryan; Saeidi, 2023).

## 2.3 Visão Computacional para Educação em Saúde

Com o avanço das tecnologias de visão computacional, novas abordagens têm surgido para aprimorar o ensino de procedimentos médicos. Bibliotecas como MediaPipe e YOLO permitem a detecção e rastreamento de poses humanas em tempo real, facilitando a análise automática de movimentos (Debalaxmi; Vishwakarma; Ranga, 2024). Essas ferramentas utilizam CNNs para identificar pontos-chave no corpo humano, como ombros e cotovelos, essenciais para a execução correta das compressões torácicas durante a RCP.

### 2.3.1 MediaPipe

O MediaPipe é uma biblioteca desenvolvida pelo Google que oferece soluções prontas para processamento de multimídia, incluindo a detecção de poses humanas. Utilizando modelos

avançados de CNN, o MediaPipe PoseLandmarker é capaz de identificar até 33 pontos-chave do corpo por pessoa em tempo real. Além das coordenadas  $(x,y)$ , cada *keypoint* retorna dois scores de confiança (presença e visibilidade), permitindo descartar detecções pouco confiáveis (Team, 2024).

### 2.3.2 YOLO

YOLO é uma família de modelos de detecção de objetos que opera em uma única passagem pela imagem, oferecendo alta velocidade e boa precisão (Ultralytics; Contributors, 2024). A extensão YOLO-Pose adapta essa arquitetura para também estimar pontos-chave corporais. Nesse contexto, o modelo YOLOv8n-pose produz 17 pontos-chave e devolve um *score* de confiança por *keypoint*, indicando a certeza da presença e localização daquele ponto. Essa abordagem, descrita em (Maji *et al.*, 2022), modifica a saída original da rede para incluir coordenadas e probabilidades associadas, sendo fundamental em cenários de análise detalhada de postura, como no treinamento de RCP.

As principais diferenças entre as duas soluções de extração de pontos-chave estão resumidas na Tabela 2.

**Tabela 2 – Comparativo entre MediaPipe e YOLOv8n-pose**

Característica	MediaPipe	YOLOv8n-pose
Versão	0.10.7	v8n-pose
# Parâmetros (M)	3.37	3.15
# Pontos-chave	33	17
Tamanho do modelo	≈18 MB	≈30 MB
Resolução de entrada	até 1920×1080	até 1280×1280
FPS (CPU)	~30 fps (1080p)	~15 fps (640×360)
FPS (GPU)	~100 fps	~80 fps
Score por <i>keypoint</i>	probabilidade de presença e visibilidade	confiança
Detecta múltiplos esqueletos	Não	Sim
Plataformas suportadas	Android, iOS, Linux, Windows	Linux, Windows, MacOS

## 2.4 Técnicas de Aumento de dados e Balanceamento de Dados

Para superar a limitação de dados disponíveis, uma opção é recorrer a estratégias que ampliem a diversidade e a cobertura do conjunto de treinamento sem coletar novos dados, para isso são aplicadas técnicas de aumento de dados e balanceamento de classes, como o *Synthetic Minority Over-sampling Technique* (SMOTE), e o Surrogate, que criam dados sintéticos a partir dos dados originais de modo a simular dados distintos, além de janelar as séries temporais em diferentes tamanhos. Assim, ao enriquecer o conjunto de dados e garantir equilíbrio

entre os diferentes tipos de classes, aumenta-se a confiabilidade e a robustez do classificador final.

#### 2.4.1 SMOTE

O SMOTE é utilizado para lidar com desequilíbrios de classes, balanceando o conjunto de dados ao gerar amostras da classe minoritária interpolando entre cada amostra e seus  $k$  vizinhos mais próximos, escolhe-se aleatoriamente um desses vizinhos e gera-se um novo ponto sintético (Chawla *et al.*, 2002). Isso é crucial para evitar que o modelo a ser treinado se torne tendencioso em relação à classe majoritária.

Matematicamente, seja  $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  o conjunto de  $N$  vetores de características da classe minoritária. Para cada  $\mathbf{x}_i \in \mathcal{X}$ :

1. Calcula-se os  $k$  vizinhos mais próximos  $N_k(\mathbf{x}_i) = \{\mathbf{x}_{i_1}, \dots, \mathbf{x}_{i_k}\}$  no espaço de características (por distância Euclidiana).
2. Seleciona aleatoriamente um vizinho  $\mathbf{x}_{i_j}$  de  $N_k(\mathbf{x}_i)$ .
3. Gera-se um escalar  $\lambda$  em  $[0,1]$  e constrói a amostra sintética

$$\mathbf{x}_{\text{smote}} = \mathbf{x}_i + \lambda(\mathbf{x}_{i_j} - \mathbf{x}_i). \quad (1)$$

4. Atribua a  $\mathbf{x}_{\text{new}}$  o mesmo rótulo de  $\mathbf{x}_i$ .

Repete-se esse procedimento até que o número de exemplos sintéticos se equilibre ao tamanho da classe majoritária.

#### 2.4.2 Surrogate

A geração de dados *Surrogate* é uma técnica de síntese de séries temporais cujo objetivo é criar réplicas artificiais que preservam o espectro de frequência do sinal original, ao mesmo tempo em que randomizam as fases das componentes, alterando a ordem temporal dos eventos (Theiler *et al.*, 1992). O procedimento básico segue os passos:

1. Transformada de Fourier: dado um sinal  $x_n$ , para  $n = 0, \dots, N - 1$ , calcula-se

$$\hat{X}_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N}, \quad k = 0, \dots, N - 1. \quad (2)$$

2. Magnitude do espectro: determina-se

$$|\hat{X}_k| = \sqrt{\hat{X}_k \hat{X}_k^*} \quad (3)$$

3. Randomização de fase: mantém-se o módulo espectral  $|\hat{X}_k|$  conforme Equação 3 e, em seguida, substituem-se as fases originais por  $\psi_k$  sorteadas uniformemente em  $[0, 2\pi]$ , conforme Equação 4, antes de aplicar a transformada inversa para gerar o *Surrogate* (Theiler *et al.*, 1992).

$$\tilde{X}_k = |\hat{X}_k| e^{i\psi_k}, \quad \psi_k \sim \mathcal{U}(0, 2\pi) \quad (4)$$

4. Transformada Inversa de Fourier: reconstrói-se no domínio do tempo

$$x_n^{\text{sur}} = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}_k e^{i2\pi kn/N}, \quad n = 0, \dots, N-1. \quad (5)$$

O *Surrogate*  $x_n^{\text{sur}}$  tem exatamente o mesmo módulo do espectro original, mas fases completamente aleatórias, garantindo que o conteúdo de frequência seja preservado sem manter a ordem temporal dos picos e vales do sinal original.

Embora esse método simples preserve o espectro, ele nem sempre reproduz perfeitamente a distribuição de amplitudes do sinal real. Para contornar isso, Chabaud *et al.* (Schreiber; Schmitz, 2000) propuseram o método AAFT (*Amplitude Adjusted Fourier Transform*), que acrescenta três etapas:

1. Normalização ranqueada: reordena-se o sinal original para seguir uma distribuição gaussiana simulada, preservando apenas o ranking dos valores.
2. *Surrogate* espectral: gera-se um *Surrogate* pelo procedimento descrito nos itens 1–4, garantindo o mesmo espectro de frequência.
3. Reatribuição de amplitudes: ordena-se o *Surrogate* obtido e, posição a posição, substitui-se cada amostra pelo valor correspondente (mesmo ranking) da série original.

Com isso, o AAFT produz réplicas que mantêm simultaneamente o espectro de potência e amplitudes, oferecendo novas séries temporais mais realistas.

## 2.5 Processamento de Sinais para Análise de Frequências

A análise das frequências das compressões torácicas é fundamental para garantir que as execuções de RCP estejam de acordo com as diretrizes recomendadas da AHA, que esta-

belecem uma taxa de 100 a 120 CPM (Feneley *et al.*, 1988). Para isso, será empregada a STFT ou Transformada de Fourier de Tempo Curto (Owens; Murphy, 1988), uma técnica que permite a análise das frequências de um sinal ao longo do tempo.

### 2.5.1 Short-Time Fourier Transform (STFT)

A STFT divide o sinal  $x(t)$  em segmentos menores, ou janelas  $w(t)$ , antes de aplicar a Transformada de Fourier a cada segmento individualmente. A representação resultante, frequentemente exibida como espectrograma, mostra a evolução do conteúdo em frequência ao longo do tempo (Boashash, 2003). Essa abordagem permite a análise de sinais longos que variam ao longo do tempo, como é o caso das compressões torácicas durante a RCP.

A STFT é definida matematicamente como:

$$\{x(t)\}(\tau, \omega) = \int_{-\infty}^{+\infty} x(t) \cdot w(t - \tau) \cdot e^{-j\omega t} dt \quad (6)$$

Onde  $x(t)$  é o sinal de entrada,  $w(t - \tau)$  é a função janela deslocada até o instante  $\tau$ ,  $\omega$  representa a frequência angular associada.

Com essas configurações, a STFT permite monitorar, por exemplo, a taxa de compressões em um RCP, identificando variações de ritmo e até mesmo detectar episódios de fadiga, fornecendo *feedbacks* essenciais para treinamentos de RCP.

## 2.6 Redes Neurais para Classificação de Sequências

Um classificador é o modelo resultante de um algoritmo de aprendizado de máquina (Bishop, 2006), que de modo geral, recebe como entrada um vetor de características extraídas de uma base dados e busca atribuir uma das categorias a uma amostra nova, com base em padrões que aparecem na base de dados aprendidos durante a fase de treinamento. Existem diversos tipos de classificadores:

- Lineares: procuram hiperplanos que separam as classes no espaço de atributos (Rosenblatt, 1958).
- Baseados em árvore: particionam recursivamente o espaço em regiões homogêneas (Loh, 2011).
- Probabilísticos: estimam distribuições de probabilidade para cada classe (Murphy, 2012).
- Neurais: utilizam camadas de unidades interligadas e funções de ativação para aprender fronteiras de decisão complexas (Goodfellow; Bengio; Courville, 2016).

A escolha do método leva em conta características do problema, como o tipo de dado, o grau de linearidade, a necessidade de memória temporal e a interpretabilidade do modelo.

### 2.6.1 Redes Neurais Recorrentes e LSTM

As Redes Neurais Recorrentes (RNNs) são uma classe de redes neurais artificiais projetadas para lidar com dados sequenciais. As RNNs possuem conexões recorrentes que permitem manter informações de estados anteriores, tornando-as ideais para tarefas que envolvem dependências temporais (Sutskever, 2013), como séries temporais e reconhecimento de padrões dinâmicos.

A estrutura de uma RNN básica consiste em uma camada oculta que recebe como entrada não apenas o estado atual  $x_t$ , mas também um estado oculto anterior  $h_{t-1}$ . Esse mecanismo permite que a rede processe informações de forma contínua, levando em consideração o contexto das entradas anteriores. A arquitetura básica de uma RNN combina a entrada  $x_t$  com o estado oculto da etapa anterior  $h_{t-1}$ , produzindo um novo estado  $h_t$  como na Equação (7) e, então uma saída  $y_t$  a partir da Equação (8).

$$h_t = \tanh(W_h h_{t-1} + W_x x_t + b) \quad (7)$$

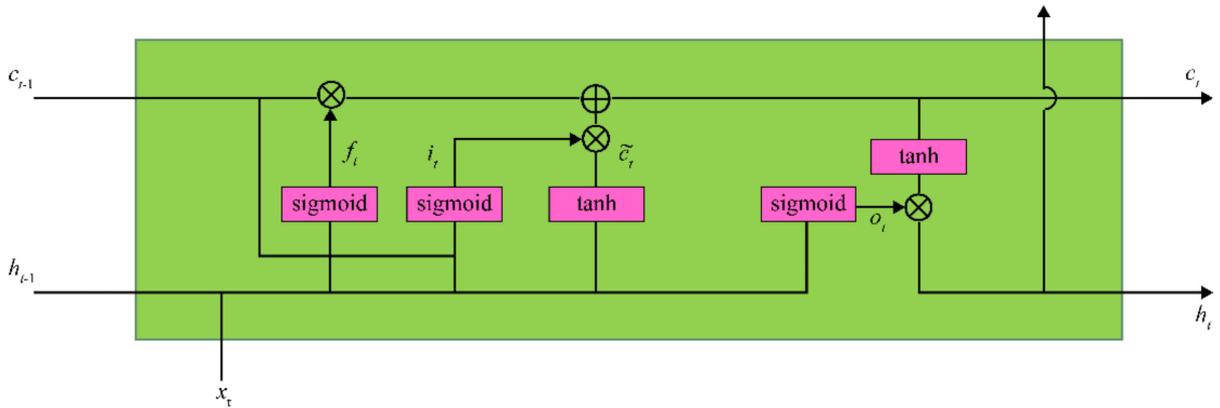
$$y_t = W_y h_t + b_y \quad (8)$$

onde  $h_t$  é o estado oculto no tempo  $t$ ,  $x_t$  representa a entrada no tempo  $t$ ,  $W_h, W_x, W_y$  são os pesos da rede que junto do viés ( $b$ ) são cruciais para que a rede evolua suas decisões, e  $y_t$  é a saída da rede.

Apesar de sua utilidade, as RNNs tradicionais enfrentam o problema do *vanishing gradient* (Noh, 2021), onde gradientes muito pequenos fazem com que a rede tenha dificuldades em aprender padrões de longo prazo, o que impacta diretamente a capacidade de modelar sequências longas, como os movimentos durante a execução da RCP.

Para mitigar o *vanishing gradient*, surgiu o modelo LSTM, que introduz uma arquitetura mais elaborada, permitindo a preservação de informações relevantes por períodos maiores (Hochreiter; Schmidhuber, 1997). As LSTMs incluem mecanismos chamados de portas de entrada, esquecimento e saída, que regulam a adição e remoção de informações do estado interno da célula como mostrado na Figura 1. Essas portas e estados permitem que a LSTM mantenha e atualize informações relevantes ao longo das sequências temporais, melhorando a capacidade de modelagem ao longo prazo.

Figura 1 – Arquitetura básica de uma célula LSTM



$f_t$ : porta de esquecimento,  $i_t$ : porta de entrada,  $\tilde{C}_t$ : candidato ao estado,  $o_t$ : porta de saída

Fonte: Adaptado de (Noh, 2021).

A Equação (9) representa a porta de esquecimento ( $f_t$ ), que determina quais informações do estado anterior  $C_{t-1}$  devem ser descartadas.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (9)$$

A porta de entrada ( $i_t$ ), descrita na Equação (10), decide quais novas informações serão adicionadas ao estado da célula.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (10)$$

A Equação (11) define o candidato ao estado da célula ( $\tilde{C}_t$ ), que introduz novas informações através da função tanh.

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (11)$$

O estado da célula atualizado  $C_t$ , conforme a Equação (12), combina as informações esquecidas e as novas informações.

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (12)$$

A porta de saída ( $o_t$ ) na Equação (13) controla a quantidade de informação do estado da célula que será transmitida para a saída final  $H_t$ , definida na Equação (14).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (13)$$

$$H_t = o_t \odot \tanh(C_t) \quad (14)$$

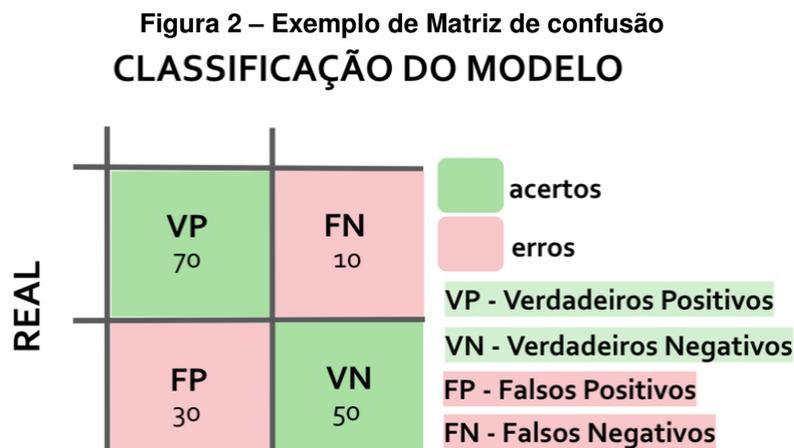
## 2.7 Métricas de Avaliação e Validação

Avaliar e validar os modelos de classificação é fundamental para verificar se o sistema proposto atende aos objetivos de maneira eficaz. O método de validação utilizado influencia diretamente a robustez e a confiabilidade dos resultados (Powers, 2020). Nesta seção, são apresentadas as principais métricas de avaliação e o papel da validação cruzada no processo.

### 2.7.1 Matriz de Confusão

A matriz de confusão é uma ferramenta fundamental para analisar o desempenho de classificadores, oferecendo uma visão detalhada sobre os erros cometidos em cada classe (Devopedia, 2019), ela é ilustrada na Figura 2 e organizada da seguinte forma:

- Verdadeiros Positivos (TP): casos em que o modelo previu “correto” e o rótulo real também era “correto”.
- Falsos Positivos (FP): casos em que o modelo previu “correto” mas o rótulo real era “incorreto”.
- Falsos Negativos (FN): casos em que o modelo previu “incorreto” mas o rótulo real era “correto”.
- Verdadeiros Negativos (TN): casos em que o modelo previu “incorreto” e o rótulo real também era “incorreto”.



Fonte: Adaptado de (Ruback; Avila; Cantero, 2021).

A partir desses quatro valores, derivam-se métricas importantes, Acurácia, Precisão, Revocação e F1-score, essas métricas ajudam a identificar se o classificador está cometendo erros ao definir classes incorretas como corretas (FP), o que poderia gerar um falso senso de segurança, ou ao penalizar classes corretas (FN), o que reduz a veracidade do sistema.

### 2.7.2 Acurácia

A acurácia mede a proporção de previsões corretas tanto verdadeiros positivos quanto verdadeiros negativos, em relação ao total de amostras avaliadas. Embora amplamente utilizada, a acurácia pode ser insuficiente em cenários de classes desbalanceadas, pois um modelo que classifique a maioria das instâncias como pertencentes à classe majoritária ainda pode exibir uma acurácia elevada (Tan; Steinbach; Kumar, 2019).

$$\text{Acurácia} = \frac{\text{Verdadeiros Positivos} + \text{Verdadeiros Negativos}}{\text{Total de Amostras}} \quad (15)$$

### 2.7.3 Precisão e Revocação

A Precisão indica a proporção de predições positivas que estão corretas, calculada pela razão entre verdadeiros positivos (TP) e o total de predições positivas (TP + FP), conforme a Equação (16). Já a revocação mede a capacidade do classificador de identificar todos os casos positivos reais, obtido pela razão entre verdadeiros positivos (TP) e a soma de verdadeiros positivos com falsos negativos (FN), conforme a Equação (17). Em outras palavras, a precisão reflete o grau de confiança nas predições positivas, enquanto a revocação quantifica a cobertura do modelo sobre os casos positivos no conjunto de dados (Tatbul *et al.*, 2018).

$$\text{Precisão} = \frac{TP}{TP + FP} \quad (16)$$

$$\text{Revocação} = \frac{TP}{TP + FN} \quad (17)$$

### 2.7.4 F1-Score

O F1-Score é uma métrica que combina a Precisão e a Revocação em uma única medida, por meio da média harmônica dessas duas métricas. Ele é especialmente útil em situações com classes desbalanceadas, nas quais uma alta acurácia pode ser enganosa. O F1-Score oferece um equilíbrio entre a capacidade do modelo de evitar falsos positivos (representados pela Precisão) e falsos negativos (representados pela revocação) (Sasaki, 2007). Seus valores variam entre 0 e 1, sendo 1 o valor ideal, que representa máxima Precisão e Revocação simultaneamente.

$$F1 = 2 \cdot \frac{\text{Precisão} \times \text{Revocação}}{\text{Precisão} + \text{Revocação}} \quad (18)$$

### 2.7.5 Validação Cruzada

Para estimar de forma confiável a capacidade de generalização do modelo, emprega-se a validação cruzada, uma técnica que divide o conjunto de dados em múltiplas partições (*folds*), garantindo que cada partição seja usada como conjunto de validação em uma das rodadas de treinamento (Nti; Nyarko-Boateng; Aning, 2021). Dessa forma, cada instância do conjunto de dados participa tanto do treinamento quanto da validação, reduzindo o viés de uma única partição de teste, fornecendo visão mais completa do comportamento do modelo.

- *K-Fold Cross-Validation*: No método *K-fold*, o conjunto de dados é dividido em  $k$  partições. Em cada rodada, o modelo é treinado na posição  $k - 1$  e validado na partição restante. Ao final, a média dos resultados de cada uma das  $k$  rodadas reflete o desempenho geral do modelo.
- *Stratified K-Fold* Variante do *K-fold* que preserva a proporção de cada classe em todos os folds. Fundamental em problemas com classes desbalanceadas, evita que algumas iterações fiquem sem exemplos de uma das categorias (Kohavi, 1995).
- *Leave-One-Out (LOO)* Caso extremo de *K-fold* onde  $k = N$ , o número de amostras. Em cada rodada, usa-se  $N - 1$  amostras para treinar e 1 para validar. Fornece estimativas quase não tendenciosas, mas é computacionalmente caro para bases grandes e apresenta alta variância das métricas (Allen, 2012), não ideal para bases de dados pequenas.

Para estimar de forma confiável a capacidade de generalização do modelo no projeto atual, utilizou-se a técnica de *K-Fold Cross-Validation* com  $k = 10$ . Nesse procedimento, o conjunto de dados é dividido em dez partições de tamanho similar, em cada uma das dez iterações, o modelo é treinado em nove *folds* e validado no *fold* restante, que permite aproveitar ao máximo todos os exemplos de treinamento, importante quando a base é limitada.

## 2.8 Otimização de Hiperparâmetros

A definição adequada de hiperparâmetros é um fator determinante para o bom desempenho de modelos de aprendizado de máquina, em especial quando se trata de arquiteturas mais complexas, como as redes neurais. Hiperparâmetros são variáveis que controlam aspectos cruciais do processo de treinamento e da estrutura do modelo (Goodfellow; Bengio; Courville, 2016). No caso específico de redes LSTM, os principais hiperparâmetros a serem considerados incluem:

- Número de camadas LSTM: Define quantas camadas empilhadas de memória longa-curta serão utilizadas.

- Número de unidades ocultas (neurônios) por camada: Afeta diretamente a capacidade de representação da rede.
- Taxa de aprendizado: Controla o tamanho dos passos durante a atualização dos pesos.
- Otimizador: Algoritmos como Adam, RMSprop ou SGD são comumente usados (Kingma; Ba, 2015).
- Tamanho do lote: Número de amostras processadas antes da atualização dos pesos.
- Número de épocas: Total de vezes que o modelo percorre todo o conjunto de dados de treinamento.
- Taxa de *dropout*: Percentual de unidades (neurônios) que serão aleatoriamente desligadas durante cada iteração do treinamento, impedindo que participem da atualização dos pesos naquela etapa. Essa técnica atua como uma forma eficaz de regularização, forçando a rede a não depender excessivamente de caminhos específicos, aumentando a capacidade de generalização do modelo (Srivastava *et al.*, 2014).

### 2.8.1 Optuna

Ajustar manualmente cada hiperparâmetro pode ser um processo demorado e sujeito a erros, principalmente em cenários que envolvem múltiplas variáveis de configuração, para tornar o ajuste de hiperparâmetros mais sistemático e eficiente, foi utilizada a biblioteca Optuna, que implementa métodos de busca como otimização bayesiana e pesquisa em árvore de Parzen (TPE), possibilitando a seleção automática de combinações ótimas de parâmetros com base no desempenho do modelo (Koyama, 2019).

- Automação do Processo de Ajuste: O Optuna monitora o desempenho do modelo em diferentes combinações de hiperparâmetros e direciona as próximas tentativas para regiões mais promissoras do espaço de busca (Team, 2023).
- Integração com Validação Cruzada: Cada configuração de hiperparâmetros pode ser avaliada utilizando a *K-Fold Cross-Validation* ou um conjunto de validação independente, registrando automaticamente métricas por tentativas e fornecendo relatórios e para monitoramento, tornando a estimativa da performance mais confiável.
- *Pruning* (Interrupção Antecipada): Para economizar tempo computacional, o Optuna suporta *pruners* que interrompem tentativas cujas curvas de validação mostram desempenho inferior ao esperado em estágios iniciais.
- Seleção do Melhor Modelo: Ao término do processo, o Optuna informa qual conjunto de hiperparâmetros obteve a melhor pontuação na métrica selecionada, bem como o histórico completo das tentativas testadas.

Optuna pode otimizar hiperparâmetros do treinamento e de arquiteturas LSTM (número de camadas, tamanho do estado oculto, *dropout*), com essa abordagem busca-se obter um modelo de classificação para as execuções de RCP que apresente alta acurácia, revocação e precisão. O ajuste orientado de hiperparâmetros, aliado as métricas de avaliação e a validação cruzada, fornece uma base sólida para garantir que o modelo final seja confiável.

### 3 MATERIAIS E MÉTODOS

O processo de ensino de RCP (Reanimação Cardiopulmonar) tradicionalmente envolve a combinação de conteúdos teóricos, demonstrações práticas e treinamento em manequins. Esse método costuma ter boa eficácia, pois permite ao aluno praticar e receber *feedback* imediato de instrutores especializados. No entanto, abordagens baseadas exclusivamente em vídeo ou em módulos de autoaprendizagem podem ser menos eficazes, pois carecem de interatividade ou de correção imediata dos movimentos (Ghanbaryan; Saeidi, 2023). De acordo com esse estudo, mesmo quando um módulo de autoaprendizagem em vídeo é disponibilizado, há limitações, como a ausência de supervisão em tempo real e a dificuldade de avaliar o desempenho prático de cada aluno, gerando possível receio de executar o procedimento em público, medo de assumir responsabilidades em uma situação crítica e insegurança.

O presente trabalho propõe uma ferramenta computacional que emprega bibliotecas de detecção de pose em vídeos e uso de redes neurais, permitindo a análise dos principais parâmetros da RCP a partir de gravações comuns, feitas com câmeras de fácil acesso. Seguindo as etapas descritas nas próximas seções, busca-se fornecer um método complementar que possa oferecer respostas mais objetivas e detalhadas sobre a qualidade das compressões, sendo elas divididas em duas modalidades, o *feedback* de avaliação, e o *feedback* de assistência.

#### 3.1 Materiais

Os materiais utilizados neste projeto compreendem ferramentas, tecnologias e ambientes de desenvolvimento necessários para a criação e implementação da ferramenta de análise de RCP. A seguir, são descritos os principais materiais empregados.

##### 3.1.1 Base de Dados

Os dados fornecidos vem de um projeto de colaboração PIPE-FAPESP nº 2021-15114-8 com a empresa Hipocampus, e para captura dos vídeos das execuções de RCP, foi utilizada uma câmera fotográfica Canon EOS Rebel T7.

A base de dados consiste em 16 vídeos, 12 vídeos demonstrando uma execução correta, e quatro vídeos para as execuções erradas, com resolução de 1920 por 1080 pixels e taxa de quadros de 30 *Frames por segundo* (FPS). As execuções foram realizadas e acompanhadas por profissionais de saúde, e os vídeos foram rotulados manualmente para distinguir entre execuções corretas e incorretas, a Figura 3 exemplifica como foram feitas a captura dos vídeos. As faces dos participantes do estudo foram anonimizados afim de garantir a privacidade dos participantes.

**Figura 3 – Frame de exemplo dos vídeos capturados**



**Fonte: Hipocampus (2024).**

### 3.1.2 Linguagens de Programação e Bibliotecas

A linguagem principal escolhida para o desenvolvimento foi Python, devido a sua flexibilidade e compatibilidade com diversas bibliotecas de aprendizado de máquina e visão computacional. As bibliotecas utilizadas incluem:

- OpenCV: Para pré-processamento e padronização dos vídeos (Team, 2024).
- MediaPipe: Para detecção de poses (Team, 2024).
- YOLO: Adaptado para detecção de poses (Ultralytics; Contributors, 2024).
- PyTorch: Para implementação e treinamento do modelo LSTM (Team, 2023).
- Optuna: Biblioteca para otimização de hiperparâmetros dos modelos (Team, 2023) .
- SciPy e NumPy: Para operações matemáticas e implementações de transformadas de Fourier, para análise dos sinais (Virtanen *et al.*, 2020; Harris *et al.*, 2020).
- Matplotlib: Para visualização de dados e resultados (Hunter, 2007).

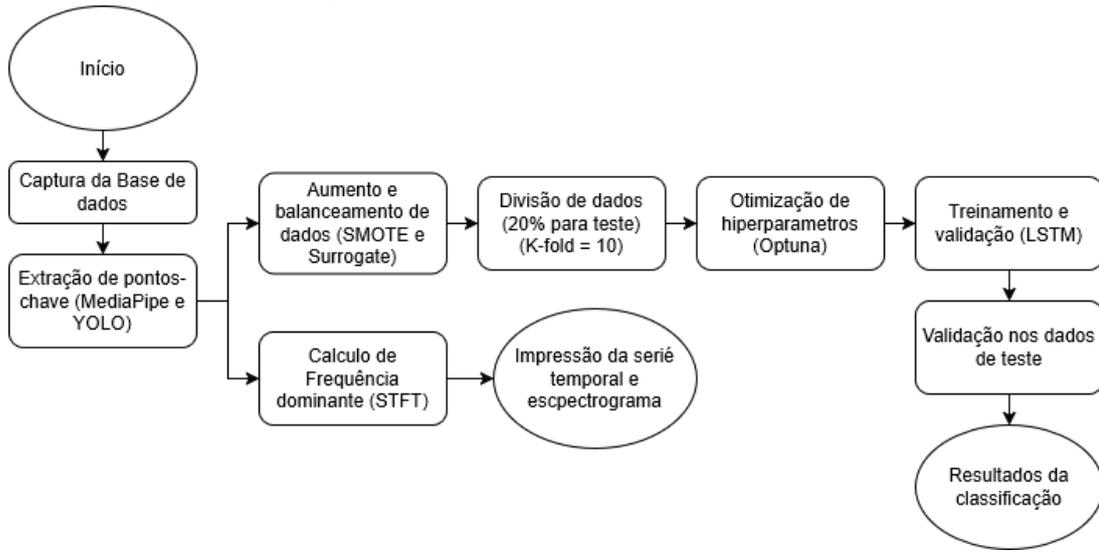
### 3.1.3 Ambiente de Desenvolvimento

O desenvolvimento foi realizado na plataforma *Google Colab*. Devido a plataforma não requerer instalações locais, qualquer dispositivo com acesso à internet e um navegador compatível pode ser utilizado, agilizando a mobilidade no desenvolvimento. As bibliotecas utilizadas nesse projeto podem ser configuradas com poucos comandos, otimizando o tempo de preparação do ambiente.

## 3.2 Métodos

Os métodos adotados neste projeto definem o plano geral para o desenvolvimento da ferramenta de análise de RCP, englobando desde a coleta e preparação dos dados até a implementação dos métodos e validação do modelo treinado. As principais atividades a serem realizadas estão detalhadas no fluxograma da Figura 4 a seguir.

**Figura 4 – Fluxograma do Sistema de Avaliação e Assistência de RCP**



### 3.2.1 Preparação dos Dados

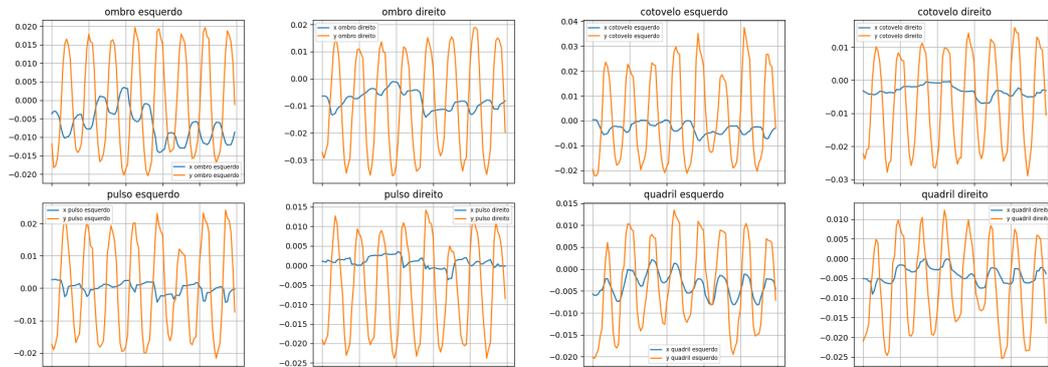
- **Aquisição dos Vídeos:** Os vídeos foram capturados durante sessões de treinamento internas em uma instituição de saúde parceira, todas as gravações foram realizadas com câmera em ângulo fixo, selecionado para facilitar a captura de pontos importantes no corpo a serem analisados posteriormente pelos extratores, realizados em ambiente com iluminação controlada.
- **Pré-processamento dos Vídeos:** O OpenCV foi utilizado para padronizar os vídeos em uma resolução menor (640x360 pixels), e fixá-los em uma duração de 3300 quadros, que representa aproximadamente 2 minutos de duração para cada vídeo. Mantendo toda a base de dados com um tamanho padrão e diminuindo o custo e tempo computacional na extração de pontos-chave pelos algoritmos MediaPipe e YOLO.

### 3.2.2 Extração de Características

As ferramentas utilizadas para a extração de pontos-chave foram o MediaPipe e o YOLO (yolov8n-pose). Em cada quadro do vídeo, foram obtidas os pontos do corpo (ombros, quadris, cotovelos e pulsos) essas ferramentas retornam um vetor de coordenadas desses pontos no

*frame*, e a sua variação no tempo gera séries temporais que representam o movimento de cada ponto do corpo ao longo do tempo, como mostra a Figura 5.

**Figura 5 – Exemplo de série temporal gerada a partir do vetor de coordenadas extraído de um vídeo de RCP**



O MediaPipe utiliza uma rede neural, que fornece diretamente as posições com boa robustez e não requer filtragem adicional de confiança. Em contraste, o YOLOv8-Pose emprega uma arquitetura de detecção de objetos seguida de regressão de pose, o que tende a ser mais pesado e suscetível a falsos positivos. Por este motivo foi implementado um filtro de confiança para garantir a qualidade no sinal, que descarta todo quadro cuja média de confiança dos pontos-chave fique abaixo de 0,50. Nesse caso, a extração via MediaPipe mostrou-se mais direta e estável, enquanto via YOLO precisou-se de um módulo de validação extra para descartar detecções errôneas.

Além do tratamento inicial de detecção dos pontos-chave, foi aplicado um passo de *detrend*, que busca zerar o deslocamento estático de cada coordenada ao longo do tempo nas séries temporais, subtraindo a média dos valores de coordenadas. O *detrend* remove não apenas a média, mas também qualquer mudança que possa surgir de movimentos ou de posicionamento de câmera, por isso é essencial fazer esse passo para não enviesar o modelo em direção a uma das classes.

Para uma eventual escolha de qual técnica é mais adequada para o projeto, foram realizados testes com efeitos de comparação, com a base dos elementos comparativos da Tabela 2. A partir das métricas de avaliação de aprendizado de máquina, buscou-se identificar qual método apresentou melhor desempenho numérico na hora do treinamento do modelo.

### 3.2.3 Aumento de Dados

Devido à quantidade limitada de vídeos disponíveis (16 vídeos no total), foi empregada uma técnica de aumento de dados chamada Surrogate, essa técnica serviu para enriquecimento dos sinais de coordenadas corporais extraídos dos vídeos de RCP. Em vez de simplesmente duplicar ou adicionar ruído branco, utilizamos o método AAFT (*Amplitude Adjusted Fourier Transform*) para gerar séries sintéticas que preservam a distribuição de amplitudes e as componentes espectrais do sinal original. Isso significa que cada sequência Surrogate mantém o ritmo de compressões característico da RCP, mas com variações sutis de fase e padrão.

Além disso, foi utilizada a técnica SMOTE (Chawla *et al.*, 2002) para lidar com o desbalanceamento entre as classes de execução correta e incorreta de RCP. O SMOTE gera amostras sintéticas da classe minoritária ajudando a equilibrar a distribuição e melhorando a capacidade do modelo de aprender padrões de ambas as classes.

Para criar essas amostras, as séries temporais são achatadas, cada vídeo, originalmente em um vetor de dimensão  $(n\_frames, feature)$ , é convertido em um vetor 1-D. Em seguida, para cada amostra da classe minoritária (incorreta), o algoritmo seleciona  $k = 3$  vizinhos mais próximos, cujo valor  $k$  foi selecionado devido ao baixo número de amostras, e interpola pontos entre a amostra e cada vizinho, gerando novas instâncias realísticas.

A combinação das técnicas *Surrogate* e SMOTE permitiu um aumento significativo na diversidade e no volume dos dados de treinamento, saindo de 12 vídeos corretos e 4 incorretos, para uma base de dados com 24 amostras de cada classe, contribuindo para a melhoria da performance e da generalização do modelo de classificação de execuções de RCP. Este procedimento assegura que o modelo LSTM receba uma representação equilibrada e variada das execuções de RCP, facilitando a aprendizagem de padrões relevantes para a classificação precisa das execuções como corretas ou incorretas.

### 3.2.4 Análise de Periodicidade

Para quantificar o ritmo e a regularidade das compressões em cada vídeo de RCP, aplicamos a Transformada de Fourier de Tempo Curto (STFT) diretamente nas séries temporais de coordenadas extraídas pelo YOLO, buscando encontrar valores de compressões por minuto, a fim trazer informações importantes para uma análise de movimento de RCP. O procedimento foi o seguinte:

- Parâmetros da STFT: Foram usadas janelas de tamanho  $t = 256$  quadros, com *overlap* = 128, e frequência de amostragem equivalente ao fps do vídeo. Esses valores equilibram resolução de frequência e de tempo, suficiente para capturar variações de curto prazo (80-150 cpm), e facilitam a visualização de eventos pelo espectrograma.

- **Espectrograma:** Para cada *keypoint* selecionado, o espectrograma mostra a de frequência distribuída ao longo do tempo. Isso permite visualizar possíveis flutuações de ritmo, por exemplo, acelerações ou desacelerações e identificar possíveis interrupções na cadência.
- **Média da frequência:** Após computar a STFT em cada janela de tempo  $t$ , extraímos a frequência de pico (o índice de maior magnitude). Este valor, originalmente em Hz, é convertido para compressões por minuto, que serve como nossa estimativa global de ritmo de compressões. Além disso, computamos o desvio-padrão para quantificar a variabilidade do ritmo ao longo do movimento. Dessa forma, se identifica não só indica se o aluno está mantendo o intervalo ideal de 100–120 CPM, mas também revela instabilidades (acelerações súbitas ou pausas) que podem comprometer a eficácia da RCP.

### 3.2.5 Criação e Treinamento do Modelo

Nesta etapa, buscamos transformar as sequências temporais de coordenadas, já pré-processadas, balanceadas e enriquecidas, em classificações de execuções RCP. Para isso, é feita a escolha da rede neural capaz de capturar padrões dinâmicos nas séries temporais, a divisão inicial dos dados de treinamento e teste, a validação cruzada, um estudo para otimização dos hiperparâmetros de treinamento, e por fim, a avaliação em um conjunto de teste independente.

- **Separação do conjunto de teste:** Para os testes finais do modelo, o *dataset* foi dividido inicialmente, de modo que 80% das amostras fossem utilizadas para o treinamento e 20% fossem utilizadas para testes finais.
- **Validação Cruzada:** foi utilizada a técnica de *Cross-Validation* do tipo *K-Fold* com 5 conjuntos para avaliar a performance do modelo (Nti; Nyarko-Boateng; Aning, 2021), dividindo de forma que 9 dos 44 vídeos (após aumento de dados) foram utilizados em cada etapa para validação. Isso ajuda a garantir que o modelo generalize bem para dados não vistos.
- **Arquitetura do Modelo:** Nosso classificador é baseado em uma rede LSTM de múltiplas camadas, com hiperparâmetros definidos pelo estudo do Optuna, permitindo que a rede capture dependências de curto e longo prazo nas sequências de compressões.
- **Treino Final e Teste:** após encontrar os melhores hiperparâmetros, re-treinamos o modelo em 80% dos dados e avaliamos os 20% finais retidos como teste inédito, reportando matriz de confusão e métricas de avaliação.

### 3.2.6 Validação e Testes

Para assegurar a confiabilidade do sistema, todas as variantes propostas são submetidas ao mesmo protocolo de validação e avaliação, de forma a permitir uma comparação justa e direta de suas capacidades preditivas e de eficiência computacional.

- **Avaliação por *K-Fold*:** Ao utilizar *K-Fold* com  $k = 10$  folds, em cada iteração, o modelo é treinado em  $k - 1$  partições e validado no *fold* restante, gerando para cada *fold* métricas de acurácia, precisão, *recall* e F1-Score. Ao final, calculamos a média e o desvio-padrão dessas métricas sobre todos os *folds*, bem como somamos as matrizes de confusão individuais para uma visão consolidada do comportamento do classificador.
- **Otimização com Optuna:** Cada tentativa do Optuna envolve a seleção de um conjunto de candidatos para cada hiperparâmetro (taxa de aprendizado, número de camadas LSTM, número de neurônios, *dropout*, *batch size*). O Optuna emprega busca Bayesiana para avaliar cada conjunto e pruning automático para descartar rapidamente tentativas fracas. Ao término das tentativas, o Optuna reporta o melhor conjunto de hiperparâmetros e re-treina o modelo final sobre todo o conjunto de treino.
- **Teste Final Independente:** Após validação cruzada e busca de hiperparâmetros, reservamos 10% dos vídeos para avaliação final. Nesse teste, é informado a matriz de confusão e a avaliação completa do treinamento por cada uma das métricas (F1-score, precisão, *recall*).

Esta metodologia combina técnicas avançadas de visão computacional e aprendizado de máquina para desenvolver uma ferramenta eficaz na análise da qualidade da RCP em vídeos. Permitindo uma avaliação precisa e objetiva dos movimentos realizados durante a RCP. A validação com testes controlados e rotulagem manual assegurará a confiabilidade e a robustez do modelo desenvolvido, contribuindo para a melhoria das práticas de RCP e para o aprimoramento do treinamento de profissionais de saúde.

## 4 RESULTADOS

Este capítulo apresenta os resultados obtidos a partir da implementação e testes da ferramenta de análise da qualidade de RCPs em vídeos. Inicialmente, será detalhado o escopo do sistema, seguido pela modelagem do sistema e implementação.

### 4.1 Escopo do sistema

A ferramenta desenvolvida tem como principal objetivo avaliar automaticamente a qualidade das execuções de RCP realizadas em vídeos, diferenciando entre desempenhos corretos e incorretos, bem como auxiliar a avaliação manual a partir de informações de métricas e gráficos gerados a partir dos movimentos. As funcionalidades essenciais do sistema incluem a detecção e rastreamento de poses utilizando modelos avançados de visão computacional e cálculo da periodicidade das compressões torácicas. A ferramenta busca ser uma ajuda valiosa no treinamento de profissionais de saúde, oferecendo informações para facilitar a avaliação da técnica de RCP.

### 4.2 Extração de Características

A extração de poses foi implementada utilizando as bibliotecas MediaPipe e YOLO. Estas bibliotecas permitiram a detecção e rastreamento de pontos-chave dos ombros, quadril, cotovelos e pulsos, em cada quadro do vídeo, assim criando os dados de frequência das compressões, gerando séries temporais de formato  $(T, K)$  onde  $T$  representa o número de quadros e  $K$  o número de pontos de interesse. A função apresentada no trecho de Código 4.1 exemplifica o processo de extração de pontos-chave no corpo humano com a biblioteca MediaPipe, configurada para detectar 33 pontos-chave do corpo humano,  $PTS$  contém os índices dos pontos de interesse (cotovelos, punhos, ombros e quadris) segundo a enumeração oficial do MediaPipe. Assim, apenas os  $(x, y)$  desses pontos-chave são empilhados no vetor de *frames*. Ao término do laço, a função devolve um vetor  $(n_{frames}, 2 \times |PTS|)$ , que serve de entrada para a etapa de análise de ritmo das compressões torácicas.

#### Código 4.1 – Exemplo de função para extração de pontos-chave com o MediaPipe.

```

1 POSE = mp.solutions.pose.Pose(static_image_mode=False)
2 PTS = [11, 12, 13, 14, 15, 16, 23, 24] # Indices de pontos-chave
3 while True:
4     ok, frame = cap.read()
5     if not ok:
6         break

```

```

7     # redimensiona e converte para RGB
8     img = cv2.resize(frame, (RESIZE_W, RESIZE_H)) #640x360
9     rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
10    # detecta pose
11    res = POSE.process(rgb)
12    if res.pose_landmarks:
13        lm = res.pose_landmarks.landmark
14        coords = []
15        # extrai (x,y) em cada ponto
16        for p in PTS:
17            coords.extend([lm[p].x, lm[p].y])
18        frames.append(coords)
19 cap.release()
20 return np.array(frames, dtype=np.float32)

```

Na extração em YOLOv8-Pose, devido a detectar o manequim de treinamento como um corpo humano, só é aceita uma detecção quando:

- Confiança mínima por *keypoint*: cada ponto deve ter score de confiança maior que 0,5 para ser considerado válido, evitando detecções errôneas que possuem baixo valor de confiança.
- Filtro de salto: a variação absoluta em qualquer coordenada entre o local atual e o último *frame* aceito não pode exceder um limiar de 50 pixels, evitando detecções erráticas pois o padrão do movimento é contínuo e com baixa variação.

Após a extração, cada série pode apresentar pequenas variações de câmera que influenciam nas coordenadas. Para correção deste problema é aplicado um *detrend* em cada coluna da matriz, que subtrai a média dos valores das coordenadas, buscando deixar esses dados representando apenas a variação do movimento, como é exemplificado no Código 4.2.

#### Código 4.2 – Função para correção dos dados com *detrend*.

```

1 for fname in filenames:
2     df = pd.read_csv(src_path)           # Carrega o CSV original
3     seq = df.values.astype(np.float32)  # shape = (seq_len, n_feats)
4     # subtrai a média de cada coluna (feature)
5     mean_cols = seq.mean(axis=0, keepdims=True)
6     seq_processed = seq - mean_cols

```

### 4.3 Análise de Frequência

A análise de frequência foi implementada utilizando a STFT, permitindo identificar as frequências dominantes das compressões torácicas ao longo do tempo. Essa técnica permite representar sinais não estacionários (como as compressões em RCP) no domínio tempo-frequência.

A sequência temporal de dados foi obtida a partir dos pontos-chave extraídos com o modelo YOLO, focando, na variação vertical dos ombros, cotovelos, pulsos e quadris durante a RCP. Esses sinais foram então processados pela STFT:

#### Código 4.3 – Aplicação da STFT sobre a série temporal

```
1 f, t, Zxx = stft(seq, fs = 30.0, nperseg=128, noverlap=64)
```

Onde: `seq` representa a série temporal a ser analisada (como a posição vertical dos ombros), `fs` é a taxa de amostragem, que no nosso caso equivale aos quadros por segundo do vídeo, `nperseg` define o número de quadros por janela da STFT, `noverlap` define a sobreposição entre janelas consecutivas, a janela de 4,3 segundos (128 quadros) contém pelo menos cinco compressões, as quais oferecem resolução suficiente para identificar o ritmo. Os vetores `f` obtêm as frequências analisadas e `t` contém os tempos centrais de cada janela, enquanto `Zxx` representa a matriz complexa dos componentes espectrais.

Após a aplicação da STFT, a magnitude dos componentes espectrais é utilizada para identificar a frequência dominante em uma determinada serie temporal:

#### Código 4.4 – Cálculo das frequências dominantes em cada janela da série temporal.

```
1 idx_peaks_hz = np.argmax(mag, axis=0) # Obtém o índice da frequência dominante
2 f_peaks_hz = f_bins[idx_peaks_hz] # Mapeia os índices obtidos
3 return t_bins, f_peaks_hz, f_bins, mag
```

A frequência dominante (`f_peaks_hz`) é então convertida em compressões por minuto (CPM), permitindo também o cálculo de estatísticas como a média e o desvio padrão, conforme apresentado no seguinte trecho:

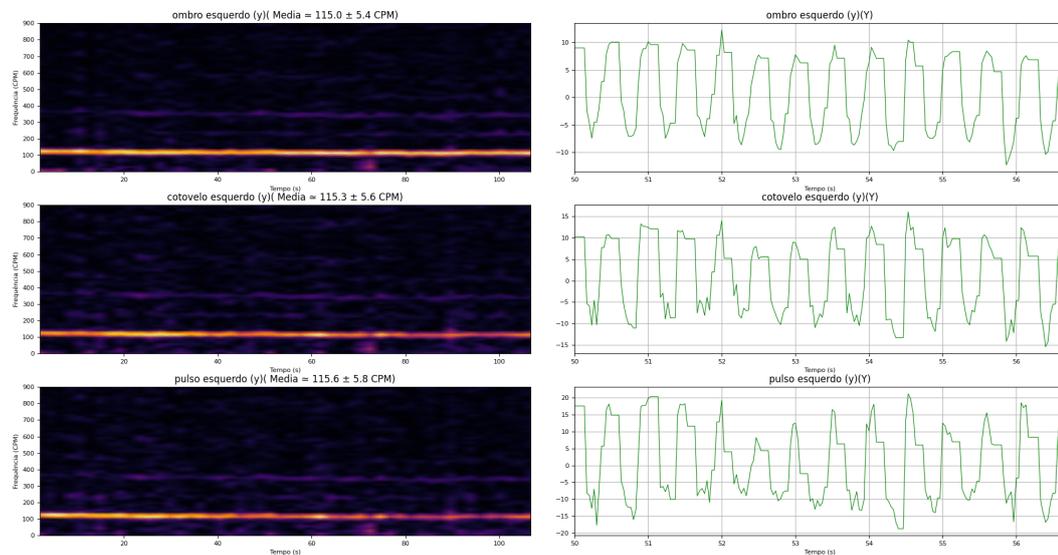
#### Código 4.5 – Conversão para CPM, cálculo de média e desvio padrão da frequência dominante.

```
1 t_bins, f_peaks_hz, f_bins, mag = freq_mean(seq) # Aplica stft
2 f_peaks_cpm = f_peaks_hz * 60.0 # Converte para CPM
3 mean_peak = np.mean(f_peaks_cpm) # Frequencia media
4 std_peak = np.std(f_peaks_cpm) # Desvio-padrão
```

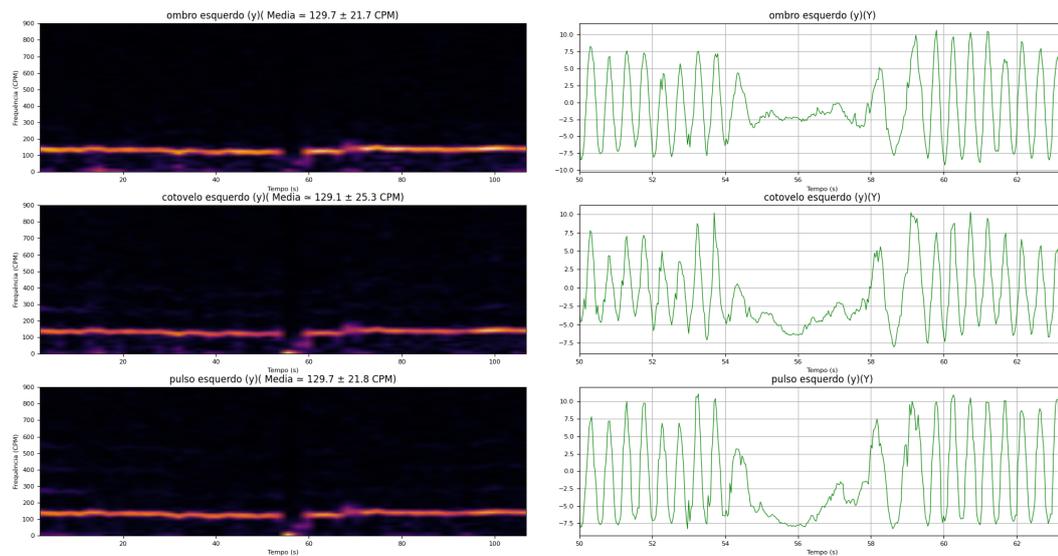
As Figuras 6 e 7 representam as duas classes de execução RCP (correto e incorreto) e cada linha corresponde a um *keypoint* (ombro esquerdo, cotovelo esquerdo, pulso esquerdo) e está dividida em dois gráfico, que permitem a diferenciação de uma execução correta e incorreta:

- Espectrograma : mostra, ao longo do tempo (eixo x), a distribuição da energia em diferentes frequências (eixo y, em CPM). Esse gráfico evidencia qual faixa de frequência predomina durante toda a compressão, além de trazer em forma de texto a frequência media do movimento e desvio-padrão.
- Série temporal: Plota o valor da coordenada (y), em função do tempo (segundos). Aqui visualizamos diretamente o movimento de subida e descida, em uma janela de tempo menor, o que facilita a inspeção de amplitudes, picos e eventuais irregularidades no gesto.

**Figura 6 – Espectrograma do movimento completo e Série temporal (Execução Correta)**



**Figura 7 – Espectrograma do movimento completo e Séries temporal (Execução Incorreta)**



#### 4.4 Aumento e Balanceamento de dados

A partir dos dados capturados o próximo passo foi aplicar as técnicas de balanceamento de classes e aumento de dados por Surrogate, a princípio o vetor inicial possuía a forma (16,3300,16) que significa 16 vídeos, de tamanho 3300 quadros e com 16 pontos-chave diferentes capturados, após a aplicação das técnicas os dados ficaram com o formato (48,3300,16) pois 8 amostras foram criadas por SMOTE e em seguida Surrogate criou um dado novo para cada amostra, duplicando o tamanho da base, agora como ultimo passo, para trabalhar com séries temporais mais curtas e ao mesmo tempo aumentar o conjunto, foi efetuado o janelamento em três diferentes espaços de tempo de 128, 256 e 512 quadros, que representam respectivamente 4.26, 8.53 e 17,06 segundos, dividindo cada serie completa de 3300 quadros em diferentes tamanhos de conjunto.

#### 4.5 Treino Final e Avaliação no Conjunto de Teste

Com as séries temporais pré-processadas, balanceadas e enriquecidas, passamos para a etapa de construção do classificador. Nesta seção será descrito, a configuração do modelo, a estratégia de busca de hiperparâmetros, os resultados de validação cruzada e teste final.

Para selecionar automaticamente os melhores parâmetros a serem usados na rede neural, foi executado um estudo de 250 tentativas via Optuna. O Algoritmo a seguir mostra, de forma simplificada, quais valores de hiperparametros foram testados.

##### Código 4.6 – Exemplo de hiperparametros testados pelo Optunalabel

```

1 hidden = trial.suggest_categorical("hidden_size",[32,64,128,256])
2 layers = trial.suggest_int("num_layers",1,3)
3 drop   = trial.suggest_float("dropout",0.0,0.5,step=0.1)
4 lr     = trial.suggest_float("lr",1e-5,1e-2,log=True)
5 bs     = trial.suggest_categorical("batch_size",[16,32,64])

```

Esse estudo retornou os valores dos hiperparâmetros que obtiveram as melhores avaliações F1-score (taxa de aprendizado, *dropout*, número de camadas, *hidden size* e lote), e com esses hiperparâmetros definidos re-treinamos a LSTM em 90% dos dados e avaliamos nos 10% reservados. Os resultados deste estudo e o relatório do treinamento final estão descritos na Tabela 3, que traz os valores dos resultados finais de F1-score, Precisão e Recall para cada valor de janela adotado e das duas diferentes técnicas de extração de características abordadas neste projeto.

A Tabela 3 e a Figura 8 permitem comparar, como o tamanho da janela afeta a acurácia dos dois métodos de extração de pose. Observa-se que, com janelas de 128 e 256 quadros, MediaPipe e YOLO atingem F1-Scores semelhantes (0.91 vs. 0.95 e 0.90 vs. 0.85). Contudo,

**Tabela 3 – Parâmetros obtidos pelo Optuna e métricas dos testes LSTM para diferentes tamanhos de janela**

Parâmetro	Janela 128	Janela 256	Janela 512
Taxa de aprendizado		$1.8 \times 10^{-4}$	
<i>Dropout</i>		0.1	
Número de camadas		3	
<i>Hidden size</i>		128	
Tamanho do lote		16	
<i>K-Fold</i>		10	
Épocas		60	
Tamanho do conjunto	1100	528	264
Precisão (MediaPipe)	0.92	0.91	0.85
Precisão (YOLO)	0.95	0.87	0.74
Recall (MediaPipe)	0.89	0.88	0.85
Recall (YOLO)	0.94	0.85	0.74
F1-score (MediaPipe)	0.91	0.90	0.85
F1-score (YOLO)	0.95	0.85	0.74

**Figura 8 – Matrizes de confusão para MediaPipe e YOLO em diferentes janelas de entrada (128, 256 e 512 quadros)**

	Janela 128	Janela 256	Janela 512
<b>YOLO</b>	$\begin{pmatrix} 51 & 4 \\ 6 & 49 \end{pmatrix}$	$\begin{pmatrix} 25 & 2 \\ 3 & 23 \end{pmatrix}$	$\begin{pmatrix} 12 & 2 \\ 2 & 11 \end{pmatrix}$
<b>MediaPipe</b>	$\begin{pmatrix} 48 & 7 \\ 9 & 45 \end{pmatrix}$	$\begin{pmatrix} 22 & 4 \\ 5 & 19 \end{pmatrix}$	$\begin{pmatrix} 10 & 4 \\ 3 & 10 \end{pmatrix}$

na janela de 512 quadros que possui o menor número de amostras (264), o desempenho de YOLO cai para 0.74, indicando maior sensibilidade ao tamanho do conjunto de treino/validação. Ou seja, janelas de 128 ou 256 quadros oferecem o melhor compromisso entre robustez temporal e número de amostras para ambos os testes, porém para cenários com poucos dados, recomenda-se o MediaPipe, já que YOLO só mantém vantagem de precisão quando o conjunto é maior e as janelas cobrem intervalos menores de tempo.

## 5 CONCLUSÃO

Este trabalho apresentou uma ferramenta para o ensino e avaliação de RCP, combinando técnicas de visão computacional e aprendizado de máquina. Foi mostrado a relevância da RCP nas estatísticas de mortalidade por doenças cardiovasculares, destacando a necessidade de novos métodos que tornem o treinamento mais acessível, objetivo e eficiente. Abordou redes neurais recorrentes (LSTM) para análise de sequências, técnicas de detecção de poses (MediaPipe e YOLO) para extrair coordenadas corporais a partir de vídeos, e procedimentos de processamento de sinais (STFT) para a identificação das frequências das compressões torácicas, além do uso de técnicas de Aumento de dados e de balanceamento de classes buscando a diversidade e o equilíbrio do conjunto de dados, enquanto a otimização de hiperparâmetros por Optuna propiciou uma otimização na busca por configurações eficazes para o modelo.

Quanto ao método adotado, houve a divisão do trabalho em duas abordagens MediaPipe e YOLO para extração dos pontos-chave nos vídeos, a análise de periodicidade foi feita com STFT e em seguida foi treinada uma rede neural recorrente LSTM para classificação dos movimentos de compressão do RCP.

A ferramenta desenvolvida demonstrou a capacidade de avaliar a execução de RCP de forma automatizada, contribuindo para um treinamento potencialmente mais objetivo e padronizado, conforme demonstrado pelos resultados de classificação obtidos. A combinação de redes neurais recorrentes para análise temporal e técnicas de visão computacional para detecção de poses confirmaram-se como bons elementos para ensino e avaliação de RCPs.

No que se refere às limitações, destaca-se a quantidade restrita de vídeos disponíveis para o treinamento. Para trabalhos futuros, sugere-se ampliar ainda mais o conjunto de dados, explorando vídeos em diferentes cenários, novo método para verificação de profundidade de movimento das compressões ou local em que as mãos estão localizadas no corpo, bem como investigar arquiteturas alternativas para a análise temporal, visando maior precisão e eficiência na detecção de compressões torácicas. Acredita-se que a continuidade deste trabalho possa reforçar ainda mais a aplicabilidade prática do sistema, contribuindo para a formação de profissionais e leigos mais preparados para agir em situações de emergência cardiovascular.

## REFERÊNCIAS

- ALLEN, D. The relationship between variable selection and data augmentation and a method for prediction. **Technometrics**, v. 16, p. 125–127, 04 2012.
- BISHOP, C. M. **Pattern Recognition and Machine Learning**. [S.l.]: Springer, 2006.
- BOASHASH, B. **Time-Frequency Signal Analysis and Processing: A Comprehensive Reference**. [S.l.]: Elsevier Science, 2003.
- CAO, Z. *et al.* Realtime multi-person 2d pose estimation using part affinity fields. *In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2017.
- CHAWLA, N. V. *et al.* Smote: synthetic minority over-sampling technique. **Journal of artificial intelligence research**, AI Access Foundation, v. 16, p. 321–357, 2002.
- DEBALAXMI, D.; VISHWAKARMA, D. K.; RANGA, V. Analyzing yoga pose recognition: A comparison of mediapipe and yolo keypoint detection with ensemble techniques. *In: 2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*. [S.l.: s.n.], 2024. p. 1011–1017.
- Devopedia. **Confusion Matrix**. 2019. Version 6, August 20. Accessed 2024-06-25. Disponível em: <https://devopedia.org/confusion-matrix>.
- FENELEY, M. P. *et al.* Influence of compression rate on initial success of resuscitation and 24 hour survival after prolonged manual cardiopulmonary resuscitation in dogs. **Circulation**, v. 77, n. 1, p. 240–250, 1988. Disponível em: <https://www.ahajournals.org/doi/abs/10.1161/01.CIR.77.1.240>.
- GHANBARYAN, M.; SAEIDI, M. Self-training video module versus instructor taught pediatric basic life support according to peyton model: Report of an experience. **Advanced Biomedical Research**, Medknow, v. 12, n. 1, p. 112, 2023.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge: MIT Press, 2016.
- HARRIS, C. R. *et al.* Array programming with NumPy. **Nature**, v. 585, p. 357–362, 2020.
- HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural Computation**, v. 9, n. 8, p. 1735–1780, 1997.
- HUANG, L.-W. *et al.* Implementation of a smart teaching and assessment system for high-quality cardiopulmonary resuscitation. **Diagnostics**, v. 14, n. 10, 2024. ISSN 2075-4418. Disponível em: <https://www.mdpi.com/2075-4418/14/10/995>.
- HUNTER, J. D. Matplotlib: A 2d graphics environment. **Computing in Science & Engineering**, v. 9, n. 3, p. 90–95, 2007.
- KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **International Conference on Learning Representations (ICLR)**, 2015. Disponível em: <https://arxiv.org/abs/1412.6980>.
- KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. *In: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995. (IJCAI'95), p. 1137–1143. ISBN 1558603638.

- KOYAMA, T. A. and Shotaro Sano and Toshihiko Yanase and Takeru Ohta and Masanori. Optuna: A next-generation hyperparameter optimization framework. **CoRR**, abs/1907.10902, 2019. Disponível em: <http://arxiv.org/abs/1907.10902>.
- LI, Y. *et al.* A deep-learning-based cpr action standardization method. **Sensors (Basel, Switzerland)**, v. 24, n. 15, p. 4813, 2024.
- LOH, W.-Y. Classification and regression trees. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, v. 1, p. 14 – 23, 01 2011.
- LUGARESI, C. *et al.* **MediaPipe: A Framework for Building Perception Pipelines**. 2019. Disponível em: <https://arxiv.org/abs/1906.08172>.
- LUO, Y. *et al.* Lstm pose machines. *In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2018.
- MAJI, D. *et al.* Yolo-pose: Enhancing yolo for multi person pose estimation using object keypoint similarity loss. *In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. [S.l.: s.n.], 2022. p. 2637–2646.
- MARTIN, S. S. *et al.* 2024 heart disease and stroke statistics: A report of us and global data from the american heart association. **Circulation**, v. 149, n. 8, p. e347–e913, 2024. Disponível em: <https://www.ahajournals.org/doi/abs/10.1161/CIR.0000000000001209>.
- MERCHANT, R. M. *et al.* Part 1: Executive summary: 2020 american heart association guidelines for cardiopulmonary resuscitation and emergency cardiovascular care. **Circulation**, v. 142, n. 16\_suppl\_2, p. S337–S357, 2020. Disponível em: <https://www.ahajournals.org/doi/abs/10.1161/CIR.0000000000000918>.
- MURPHY, K. P. **Machine Learning: A Probabilistic Perspective**. [S.l.]: MIT Press, 2012.
- NOH, S.-H. Analysis of gradient vanishing of rnns and performance comparison. **Information**, v. 12, n. 11, 2021. ISSN 2078-2489. Disponível em: <https://www.mdpi.com/2078-2489/12/11/442>.
- NTI, I. K.; NYARKO-BOATENG, O.; ANING, J. Performance of machine learning algorithms with different k values in k-fold cross-validation. **I.J. Information Technology and Computer Science**, MECS, v. 6, p. 61–71, 2021.
- OBDRŽÁLEK, *et al.* Accuracy and robustness of kinect pose estimation in the context of coaching of elderly population. *In: 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. [S.l.: s.n.], 2012. p. 1188–1193.
- OLIVEIRA, G. M. M. d. *et al.* Estatística cardiovascular – brasil 2023. **Arq. Bras. Cardiol.**, v. 121, n. 2, p. e20240079, mar. 2024.
- OWENS, F.; MURPHY, M. A short-time fourier transform. **Signal Processing**, v. 14, n. 1, p. 3–10, 1988. ISSN 0165-1684. Disponível em: <https://www.sciencedirect.com/science/article/pii/0165168488900400>.
- POWERS, D. M. W. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. **CoRR**, abs/2010.16061, 2020. Disponível em: <https://arxiv.org/abs/2010.16061>.
- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, v. 65, p. 386–408, 1958. Disponível em: <https://api.semanticscholar.org/CorpusID:12781225>.

- RUBACK, L.; AVILA, S.; CANTERO, L. Vieses no aprendizado de máquina e suas implicações sociais: Um estudo de caso no reconhecimento facial. *In: Anais do II Workshop sobre as Implicações da Computação na Sociedade*. Porto Alegre, RS, Brasil: SBC, 2021. p. 90–101. ISSN 2763-8707. Disponível em: <https://sol.sbc.org.br/index.php/wics/article/view/15967>.
- SASAKI, Y. **The Truth of the F-measure**. [S.l.], 2007.
- SCHREIBER, T.; SCHMITZ, A. Surrogate time series. **Physica D: Nonlinear Phenomena**, v. 142, n. 3, p. 346–382, 2000. ISSN 0167-2789. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0167278900000439>.
- SRIVASTAVA, N. *et al.* Dropout: A simple way to prevent neural networks from overfitting. **Journal of Machine Learning Research**, v. 15, n. 1, p. 1929–1958, 2014.
- STEPHAN, F. *et al.* Teaching paediatric basic life support in medical schools using peer teaching or video demonstration: A prospective randomised trial. **Journal of Paediatrics and Child Health**, v. 54, n. 9, p. 981–986, 2018. <https://doi.org/10.1111/jpc.13937>.
- SUTSKEVER, I. **Training recurrent neural networks**. [S.l.]: University of Toronto Toronto, ON, Canada, 2013.
- TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to Data Mining**. 2. ed. Boston: Pearson, 2019.
- TATBUL, N. *et al.* Precision and recall for time series. *In: BENGIO, S. et al.* (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2018. v. 31. Disponível em: [https://proceedings.neurips.cc/paper\\_files/paper/2018/file/8f468c873a32bb0619eaeb2050ba45d1-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2018/file/8f468c873a32bb0619eaeb2050ba45d1-Paper.pdf).
- TEAM, M. **MediaPipe Framework**. 2024. Disponível em: <https://github.com/google/mediapipe>. Acesso em: 13 nov. 2024.
- TEAM, O. **Optuna: A hyperparameter optimization framework**. 2023. Disponível em: <https://optuna.org/>. Acesso em: 6 dez. 2024.
- TEAM, O. **OpenCV Library**. 2024. Disponível em: <https://opencv.org/>. Acesso em: 13 nov. 2024.
- TEAM, P. **PyTorch Documentation**. 2023. Disponível em: <https://pytorch.org/>. Acesso em: 6 dez. 2024.
- THEILER, J. *et al.* Testing for nonlinearity in time series: the method of surrogate data. **Physica D: Nonlinear Phenomena**, v. 58, n. 1, p. 77–94, 1992. ISSN 0167-2789. Disponível em: <https://www.sciencedirect.com/science/article/pii/016727899290102S>.
- TSAO, C. W. *et al.* Heart disease and stroke statistics—2022 update: A report from the american heart association. **Circulation**, v. 145, n. 8, p. e153–e639, 2022. Disponível em: <https://www.ahajournals.org/doi/abs/10.1161/CIR.0000000000001052>.
- ULTRALYTICS; CONTRIBUTORS. **YOLO (You Only Look Once)**. 2024. Disponível em: <https://github.com/ultralytics/yolov5>. Acesso em: 13 nov. 2024.
- VAONA A, B. R. K. K. R. G. C. D. P. V. T. I.; MOJA, L. E-learning for health professionals. **Cochrane Database of Systematic Reviews**, John Wiley Sons, Ltd, n. 1, 2018. ISSN 1465-1858. Disponível em: <https://doi.org/10.1002/14651858.CD011736.pub2>.
- VIRTANEN, P. *et al.* SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. **Nature Methods**, v. 17, p. 261–272, 2020.

WANG, S. *et al.* Cpr-coach: Recognizing composite error actions based on single-class training. *In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2024. p. 18782–18792.

XIE, H. *et al.* A novel algorithm of fast cpr quality evaluation based on kinect. **Journal of Algorithms & Computational Technology**, v. 14, p. 1–11, 2020.