UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

# Aplicação de Visão Computacional para Detecção de Quedas em Ambientes Domésticos Utilizando Redes Neurais YOLO e LSTM

**Kelson da Silva Batista**

Monografia - MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

**Kelson da Silva Batista**

# Aplicação de Visão Computacional para Detecção de Quedas em Ambientes Domésticos Utilizando Redes Neurais YOLO e LSTM

Monografia apresentada ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Prof. Dr. João do Espírito Santo Batista Neto

**Versão original**

**São Carlos**

**2025**

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

Batista, Kelson da Silva
B333a Aplicação de Visão Computacional para Detecção de Quedas em Ambientes Domésticos Utilizando Redes Neurais YOLO e LSTM / Kelson da Silva Batista; orientador João do Espírito Santo Batista Neto. -- São Carlos, 2025.
62 p.

Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2025.

1. quedas. 2. acidentes domésticos. 3. emergência. 4. visão computacional. 5. redes neurais. I. Batista Neto, João do Espírito Santo, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

**Kelson da Silva Batista**

# Aplicação de Visão Computacional para Detecção de Quedas em Ambientes Domésticos Utilizando Redes Neurais YOLO e LSTM

Monograph presented to the Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, as part of the requirements for obtaining the title of Specialist in Artificial Intelligence and Big Data.

Concentration area: Artificial Intelligence

Advisor: Prof. Dr. João do Espírito Santo Batista Neto

**Original version**

**São Carlos**

**2025**

*Dedico este trabalho a todas as pessoas que acreditam na inteligência artificial como ferramenta de transformação social.*
*Que este estudo possa contribuir ainda mais para o desenvolvimento de soluções tecnológicas capazes de cuidar e proteger vidas.*

# AGRADECIMENTOS

Agradeço primeiramente aos meus pais, por todo amor, dedicação e pelos valores que me transmitiram ao longo da vida, me apoiando em tudo, incondicionalmente.

Aos professores e tutores do curso, que compartilharam seus conhecimentos e experiências, contribuindo imensamente para a minha evolução e para a construção de meu futuro profissional.

Ao meu orientador João Batista, por todo o apoio, disponibilidade e, sobretudo, pela orientação precisa e enriquecedora, fundamentais para o desenvolvimento desse trabalho com qualidade e profundidade.

À professora e coordenadora do curso, Solange Rezende, por sua presença atenta e atuante, sempre pronta a esclarecer dúvidas, acolher sugestões e incentivar os alunos em cada etapa dessa trajetória.

Ao Instituto de Ciências Matemáticas e de Computação (ICMC), pela concessão de uma bolsa de estudos parcial, sem a qual minha participação no curso e acesso a este conhecimento inestimável não teriam sido possíveis.

Ao apoio das ferramentas de inteligência artificial, em especial o ChatGPT, que auxiliou nas correções gramaticais e no desenvolvimento de parte do código e imagens, sem prejuízo na geração do conteúdo autoral deste trabalho.

A todos, o meu mais sincero e profundo agradecimento. Cada um foi essencial na minha formação e realização deste trabalho.

*"O futuro não é sobre a IA substituir os humanos, mas sim sobre a IA libertar os humanos, compreender melhor os humanos e servir à humanidade."*

*Jack Ma (Co-fundador do Alibaba)*

# RESUMO

BATISTA, K. S. **Aplicação de Visão Computacional para Detecção de Quedas em Ambientes Domésticos Utilizando Redes Neurais YOLO e LSTM**. 2025. 62 p. Monografia (MBA em Inteligência Artificial e Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2025.

O aumento significativo no número de pessoas que moram sozinhas no Brasil, que saltou de 7 milhões em 2010 para 13,7 milhões em 2022, reflete mudanças sociais como o aumento da longevidade, a busca por independência e o trabalho remoto. No entanto, essa realidade traz desafios relacionados à segurança, especialmente em casos de quedas, que podem ser causadas por desmaios, distúrbios cardíacos, epilepsias, derrames ou acidentes domésticos. Para quem se encontra sozinho, a falta de supervisão pode agravar essas situações, levando a complicações graves ou até mesmo ao óbito. Apesar de medidas preventivas, como adaptações no ambiente doméstico, nem sempre é possível evitar acidentes, destacando a necessidade de soluções tecnológicas que garantam respostas rápidas. Este trabalho propõe o desenvolvimento de um modelo de detecção de quedas, através de reconhecimento de atividades humanas (HAR) utilizando técnicas de aprendizado profundo. O modelo proposto será o YOLO e LSTM, que combina Redes Neurais Convolucionais (CNN) com Redes Neurais Recorrentes (RNN) para séries temporais. Essa arquitetura é capaz de realizar classificação de atividades humanas complexas e classificação binária para detecção de quedas (queda e não queda). Os resultados esperados incluem a criação de um sistema funcional capaz de detectar quedas com precisão, utilizando algoritmos de visão computacional. Como direções futuras, planeja-se integrar funcionalidades para o envio automático de mensagens de emergência a familiares, cuidadores ou serviços médicos, além de expandir o sistema para a detecção das várias nuances de desfalecimento (ex. embriaguez, overdose), comportamento anormais (ex. surtos) e de outros tipos de acidentes domésticos, como incêndios ou vazamentos de gás. Conclui-se que a união entre tecnologia e cuidado humano pode salvar vidas e transformar a forma como lidamos com a segurança doméstica, oferecendo soluções inovadoras e acessíveis para um problema cada vez mais comum na sociedade moderna.

**Palavras-chave**: Quedas, Acidentes Domésticos, Emergência, Visão Computacional, Redes Neurais

## ABSTRACT

BATISTA, K. S. **Application of Computer Vision for Fall Detection in Domestic Environments Using Neural Networks YOLO and LSTM**. 2025. 62 p. Monograph (MBA in Artificial Intelligence and Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2025.

The significant increase in the number of people living alone in Brazil, which jumped from 7 million in 2010 to 13.7 million in 2022, reflects social changes such as increased life expectancy, the pursuit of independence, and remote work. However, this reality brings challenges related to safety, especially in cases of falls, which can be caused by fainting, cardiac disorders, epilepsy, strokes, or domestic accidents. For those who are alone, the lack of supervision can worsen these situations, leading to serious complications or even death. Despite preventive measures, such as home adaptations, it is not always possible to avoid accidents, highlighting the need for technological solutions that ensure quick responses. This work proposes the development of a comprehensive fall detection and human activity recognition (HAR) model using deep learning techniques. The proposed model will be YOLO and LSTM, which combines Convolutional Neural Networks (CNN) with Recurrent Neural Networks (RNN) for time series. This architecture is capable of performing complex human activity classification and binary classification for fall detection (fall and no fall). The expected results include the creation of a functional system capable of accurately detecting falls using cameras and computer vision algorithms. As future directions, plans include integrating functionalities for automatically sending emergency messages to family members, caregivers, or medical services, as well as expanding the system to detect various nuances of fainting (e.g., intoxication, overdose), abnormal behaviors (e.g., outbursts), and other types of domestic accidents, such as fires or gas leaks. It is concluded that the union between technology and human care can save lives and transform how we deal with domestic safety, offering innovative and accessible solutions to an increasingly common problem in modern society.



**Keywords**: Falls, Domestic Accidents, Emergency, Computer Vision, Neural Networks.

## LISTA DE FIGURAS

## LISTA DE TABELAS

## LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| AUC | Area Under Curve |
| CNN | Convolutional Neural Networks |
| C3K2 | Cross Stage Partial with Kernel Size 2 |
| IA | Inteligência Artificial |
| DATASUS | Departamento de Informática do Sistema Único de Saúde |
| FPR | False Positive Rate |
| GRU | Gated Recurrent Unit |
| HAR | Human Activity Recognition |
| IBGE | Instituto Brasileiro de Geografia e Estatística |
| IOU | Intersection over Union |
| LSTM | Long Short-Term Memory |
| mAP | Mean Average Precision |
| OBB | Oriented Object Detection |
| RNN | Recurrent Neural Networks |
| ROC | Receiver Operating Characteristics |
| TPR | True Positive Rate |
| SUS | Sistema Único de Saúde |
| USP | Universidade de São Paulo |
| USPSC | Campus USP de São Carlos |
| YOLO | You Only Look Once |

## SUMÁRIO

# 1 INTRODUÇÃO

Nos últimos anos, o número de pessoas que moram sozinhas no Brasil tem crescido de forma expressiva. Segundo o Censo do Instituto Brasileiro de Geografia e Estatística (IBGE) de 2022, cerca de 13,7 milhões de domicílios são ocupados por apenas uma pessoa, o que representa 19% do total de residências no país. Esse número quase dobrou em relação a 2010, quando aproximadamente 7 milhões de pessoas viviam sozinhas, representando 13% dos lares. Esse cenário reflete mudanças no perfil das famílias brasileiras, impulsionadas por fatores como o aumento da longevidade e também mudanças nos padrões sociais como o trabalho remoto e a busca por independência ou privacidade. No entanto, essa realidade também traz preocupações importantes relacionadas à segurança e ao bem-estar desses indivíduos, especialmente em casos de quedas e outros acidentes domésticos.

Embora os idosos sejam frequentemente associados aos riscos de quedas, pessoas de todas as idades que vivem sozinhas ou se encontram sem supervisão estão sujeitas a situações perigosas, como desmaios ou desfalecimento (por pressão baixa, overdose, dores, exaustão, estresse, jejum prolongado, etc.), distúrbios cardíacos, epilepsias, derrames ou até mesmo acidentes mais simples, como tropeços ou escorregões. Em muitos casos, a pessoa fica impossibilitada de se levantar ou pedir ajuda, podendo até perder a consciência. Para quem se encontra sozinho, a falta de supervisão pode agravar essas situações, já que a demora no socorro pode levar a complicações graves, sequelas permanentes ou até mesmo ao óbito. Diante disso, surge a necessidade de soluções tecnológicas que garantam respostas rápidas em situações de emergência.

Este trabalho tem como objetivo explorar o uso da inteligência artificial para detectar quedas em tempo real em ambientes domésticos. A proposta é desenvolver um sistema que, por meio de câmeras e algoritmos de visão computacional, identifique quedas e futuramente integre funcionalidades para emitir alertas automáticos para familiares, cuidadores ou serviços de emergência. Para isso, serão utilizados os modelos YOLO e LSTM, que combina Redes Neurais Convolucionais (CNN) com Redes Neurais Recorrentes (RNN) para séries temporais. Essa arquitetura é capaz de realizar classificação de atividades humanas complexas e classificação binária para detecção de quedas (queda e não queda).

A motivação para esse estudo, além de explorar as tecnologias relacionadas à esta aplicação, visa contribuir para a segurança e a qualidade de vida de milhões de pessoas que vivem ou se encontram sozinhas, independentemente da idade. A importância dessa solução é ainda mais evidente quando consideramos que, embora medidas preventivas, como adaptações no ambiente doméstico, sejam recomendadas, elas nem sempre eliminam completamente os riscos, já que acidentes podem ocorrer por fatores inesperados.

Ao unir tecnologia e cuidado humano, este trabalho propõe uma solução eficiente para um problema real e urgente. A detecção automática de quedas não apenas pode salvar vidas, mas também oferecer maior tranquilidade para indivíduos e famílias, especialmente em uma sociedade que valoriza a autonomia individual, mas precisa lidar com os riscos associados a esse estilo de vida. Com o desenvolvimento dessa solução, espera-se reduzir os impactos negativos de acidentes domésticos e promover um ambiente mais seguro para quem vive ou se encontra sozinho.

# 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é apresentada a fundamentação teórica necessária para o desenvolvimento do trabalho. Os tópicos incluem desde tecnologias assistivas para detecção de queda, com alguns dados estatísticos relativos a óbitos em ambiente domiciliar no Brasil, até os modelos computacionais inteligentes para a detecção de quedas a partir de vídeos.

## 2.1 Emergências em Ambientes Domésticos

Os ambientes domésticos, considerados lugares de conforto e segurança, também podem ser palco de diversas emergências que colocam em risco a saúde e a vida das pessoas. Essas situações são mais comuns do que se imagina e, muitas vezes, acontecem de forma inesperada. Por isso, representam uma preocupação de saúde pública, pois podem indicar a ocorrência de condições médicas graves.

As emergências podem afetar pessoas de qualquer faixa etária e ter diversas causas, como, por exemplo, quedas que, no contexto deste trabalho, são decorrentes de tropeços, escorregões ou eventos súbitos, como desmaios, crises epilépticas, problemas cardiovasculares, intoxicações, entre outros. Se a ocorrência não for atendida a tempo, pode agravar a enfermidade, prolongar hospitalizações e, em casos mais graves, levar ao óbito, principalmente quando a pessoa está sozinha e perde a capacidade de pedir ajuda.

Tabela 1 – Óbitos registrados em domicílios no Brasil no ano de 2022

| **Categoria** | **Causa** | **2022** | **%** |
|---|---|---|---|
| Transtornos mentais e comportamentais | Transtornos mentais e comportamentais por uso de substâncias psicoativas (incluindo álcool) | 6345 | 6.10% |
| | Outros transtornos mentais e comportamentais | 2738 | 2.63% |
| Doenças do sistema nervoso | Epilepsia | 1700 | 1.63% |
| Doenças do aparelho circulatório | Doenças isquêmicas do coração (incluindo infarto) | 42683 | 41.00% |
| | Outras doenças cardíacas | 19283 | 18.52% |
| | Doenças cerebrovasculares | 18493 | 17.77% |
| Causas externas | Quedas | 1804 | 1.73% |
| | Afogamento e submersões acidentais | 364 | 0.35% |
| | Exposição à fumaça, ao fogo e às chamas | 180 | 0.17% |
| | Envenenamento, intoxicação por ou exposição a substâncias nocivas | 185 | 0.18% |
| | Lesões autoprovocadas voluntariamente | 10318 | 9.91% |
| **Total** | | **104093** | **100.00%** |

Fonte: DATASUS (2022)

Segundo o Instituto Brasileiro de Geografia e Estatística (IBGE) (2022), o número de pessoas que moram sozinhas praticamente dobrou em relação ao censo de 2010, passando

de 7 milhões para 13,7 milhões em 2022. Esse crescimento reflete mudanças sociais como maior longevidade, o isolamento, a busca por independência e o trabalho remoto. Esses indivíduos, muitas vezes independentes, não se preparam para o declínio da saúde e, em momentos de crise, podem se encontrar desamparados (Parekh; Adams; Schuman, 2022).

De acordo com os dados do Sistema Único de Saúde (DATASUS, 2022) referentes ao ano de 2022, foram registrados aproximadamente 326 mil óbitos em domicílios, sendo 104 mil relacionados a eventos súbitos, incluindo transtornos mentais e comportamentais, doenças do sistema nervoso, doenças do aparelho circulatório e causas externas diversas, conforme apresentado na Tabela 1. Observa-se que somente as doenças do aparelho circulatório (incluindo infarto e acidente vascular cerebral) foram responsáveis por 80.459 mortes em domicílio, o que representa cerca de 77% dos casos.

Outro aspecto crítico está nas causas de intoxicações por substâncias diversas não relacionadas ao trabalho, como mostrado na Tabela 2. Os dados indicam que, em 2022, 92.859 casos de intoxicação estavam associados a medicamentos, enquanto 18.492 estavam relacionados ao uso de drogas ilícitas, sendo que, juntos, representam aproximadamente 80% dos casos das intoxicações.

Tabela 2 – Intoxicação Exógena não relacionadas a trabalho registrada no Brasil

| **Causa** | **2022** | **%** |
|---|---|---|
| Medicamento | 92859 | 65.97% |
| Agrotóxico agrícola | 1974 | 1.40% |
| Agrotóxico doméstico | 1549 | 1.10% |
| Agrotóxico saúde pública | 94 | 0.07% |
| Raticida | 4376 | 3.11% |
| Produto veterinário | 1011 | 0.72% |
| Produto uso domiciliar | 6521 | 4.63% |
| Cosmético | 1286 | 0.91% |
| Produto químico | 2033 | 1.44% |
| Metal | 197 | 0.14% |
| Drogas de abuso | 18492 | 13.14% |
| Planta tóxica | 607 | 0.43% |
| Alimento e bebida | 6403 | 4.55% |
| Outro | 3351 | 2.38% |
| **Total** | **140753** | **100.00%** |

Fonte: DATASUS (2022)

O tempo de resposta ao acidente pode ser um fator determinante na sobrevida da vítima. Diversos estudos demonstram que a redução no tempo de resposta está diretamente associada a melhores desfechos clínicos e a uma maior chance de sobrevivência em pacientes que sofreram quedas com trauma grave (Bichofe *et al.*, 2024). Nesse tipo de acidente, cada minuto de atraso no atendimento aumenta em 2% a chance de mortalidade (Nasser *et al.*, 2020).

Em casos de origem cardíaca, um estudo apontou que, para um resultado neurológico favorável e maiores chances de sobrevivência, o tempo entre a chamada de emergência e a

chegada da equipe de socorro deve ser inferior a 7,5 minutos (DW Moon HJ, 2019).

No caso de isquemias cerebrais, o paciente deve realizar uma tomografia em, no máximo, 20 minutos após a chegada ao hospital e receber o medicamento trombolítico em até uma hora (Empresa Brasileira de Serviços Hospitalares (EBSERH), 2025). Outro estudo revela que um atraso de 15 minutos no atendimento de um AVC reduz um mês de vida saudável. Além disso, o tempo de decisão para procurar um serviço de saúde foi significativamente maior entre pessoas que moram sozinhas e são solteiras, o que pode estar relacionado à ausência de alguém por perto para presenciar o evento e buscar ajuda (LS Moraes MA, 2023).

As emergências, quando agravadas, geram alto custo para os sistemas de saúde e para as famílias, exigindo internações prolongadas, reabilitação e, muitas vezes, adaptações estruturais nas residências. Diante desses dados, fica evidente a necessidade de sistemas de detecção de quedas em tempo real, essencial para a prevenção de complicações. O uso de inteligência artificial, câmeras e/ou sensores pode viabilizar uma resposta rápida, reduzindo a mortalidade e os impactos sociais desses incidentes.

## 2.2 Tecnologias Assistivas e o Desafio da Detecção de Quedas

A tecnologia assistiva corresponde a recursos desenvolvidos com o objetivo de promover a autonomia, inclusão e melhoria da qualidade de vida das pessoas. Esses recursos variam desde dispositivos simples até soluções tecnológicas avançadas, como a visão computacional aplicada à saúde. Os sistemas de detecção de quedas, por exemplo, são um tipo de tecnologia assistida voltada à segurança de indivíduos, permitindo a resposta rápida em situações de emergência.

Nesse contexto, a detecção de quedas apresenta um desafio crítico na segurança e saúde dos indivíduos, e diversas técnicas têm sido implementadas há anos para alertar essas emergências e fornecer o devido suporte em tempo hábil. Sistemas para esse fim são cada vez mais necessário devido ao envelhecimento da população, indivíduos com necessidades especiais e obesos (Nizam; Jamil, 2020). Basicamente, existem três tipos de métodos disponíveis no mercado: a detecção baseada em sensores vestíveis, sensores de ambientes e vídeo.

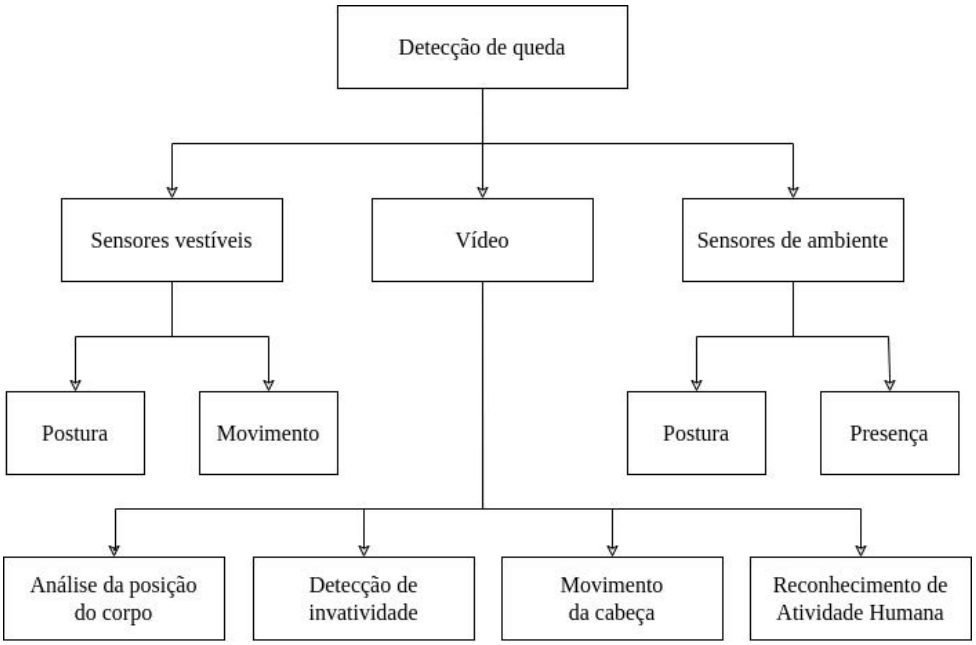
Os sistemas baseados em sensores vestíveis, por exemplo, utilizam acelerômetros e giroscópios que, embora capazes de reconhecer movimentações típicas de quedas, sofrem com falsos positivos gerados por outras atividades, como movimentos abruptos que simulam a mesma desaceleração de uma queda. Além disso, o usuário precisa utilizar o dispositivo constantemente. Porém, por vezes, as pessoas deixam de carregá-los ou de usá-los, seja pelo esquecimento ou pelo incômodo (Nizam; Jamil, 2020; Jiang *et al.*, 2023).

Existem também os dispositivos acionados por botão, mas se a queda for acompa-

nhada de perda de consciência, a pessoa não conseguirá acionar manualmente o alarme. Alguns possuem cobertura limitada e podem não detectar quedas que ocorram fora da área monitorada. Em um protótipo desenvolvido por Marques, Campos e Nascimento (2022), por exemplo, a transmissão do acionamento do botão começou a falhar em distâncias maiores que 15 metros, sem mencionar presença de paredes no teste. Já os sensores de ambientes, como pressão ou vibração no solo e elétricos em geral como infravermelho, além de também lidar com áreas específicas, podem confundir as ações de uma pessoa com algum outro objeto (Adhikari, 2019).

Por outro lado, os sistemas baseados em vídeo, embora muito promissor em diversos aspectos, possui alguns desafios relacionados ao próprio ambiente doméstico. Desenvolver um sistema capaz de monitorar quedas em tempo real dentro de residências é uma tarefa complexa (Adhikari, 2019).

Figura 1 – Abordagens dos métodos de detecção de quedas.




Fonte: Adhikari (2019) - Adaptado pelo autor.

Diferentemente de ambientes hospitalares ou clínicos, que são controlados e padronizados, as casas são espaços isolados e apresentam uma ampla diversidade de cenários e condições ambientais (Ikechukwu; Wang, 2024). Um dos principais obstáculos está nas variações de iluminação, pois durante o dia, a luz natural pode ser intensa, enquanto à noite a iluminação pode ser fraca ou inexistente. Isso afeta diretamente o desempenho das câmeras domésticas, que geralmente têm dificuldades em ambientes com baixa luminosidade, afetando a precisão da detecção (Zheng *et al.*, 2024; Zhang *et al.*, 2025).

Outro fator importante é a diversidade de ângulos e a presença de obstáculos. Quedas podem ocorrer em qualquer cômodo, em diferentes posições, e câmeras fixas muitas

vezes deixam pontos cegos. São comuns, por exemplo, obstrução da visão causadas por móveis, paredes, outras pessoas ou até animais, dificultando o reconhecimento preciso da situação (Ikechukwu; Wang, 2024) (Khalili; Mohammadzade; Ahmadi, 2022). Soma-se a isso a variação nos tipos de quedas, que podem ocorrer para frente, para trás, para os lados ou até mesmo a partir da posição sentada. A forma como essas quedas acontecem também varia, sendo algumas são rápidas e abruptas, como em escorregões e outras mais lentas, como em casos de desmaio ou quando a pessoa tenta se proteger.

Além disso, a diversidade de indivíduos também é um desafio para os sistemas de visão computacional. Diferenças na idade, porte físico e no próprio padrão de movimento podem dificultar a criação de modelos generalistas que funcionem igualmente bem para todos os usuários. Treinamentos com grandes bases de dados diversificadas são essenciais para que os algoritmos consigam identificar quedas de maneira eficaz, independentemente das diferenças individuais. Por fim, o sistema ainda precisa distinguir entre quedas reais e atividades comuns do dia a dia que se assemelham a uma queda. A velocidade vertical do tronco corporal, por exemplo, pode ser um dos fatores mais importantes para distinguir entre um evento de queda e as atividades cotidianas (Wang; Deng, 2024).

Todas essas dificuldades são susceptíveis a gerar falsos alarmes, especialmente falsos positivos, que ocorrem quando o alarme é acionado sem que ocorra uma queda real. Por exemplo, se a pessoa simplesmente decide deitar-se, já que "estar deitado" nem sempre significa uma "queda". Assim, em um cenário ideal, a detecção deve considerar a sequência do movimento, pois normalmente uma queda envolve uma transição anômala da posição em pé para o chão, diferindo de uma pessoa que se deita voluntariamente (Adhikari, 2019). Uma grande quantidade de alarmes falsos também pode gerar estresse desnecessário para o usuário e seus familiares, levando à "fadiga de alarme", situação em que os envolvidos passam a ignorar ou desacreditar as notificações, aumentando o risco de uma queda verdadeira não receber atenção adequada.

Por outro lado, um falso negativo é uma situação ainda mais crítica: quando a queda acontece, mas não é detectada. Isso significa que o sistema falhou, seja por um sensor defeituoso, uma imagem de má qualidade ou não treinada, obstrução ou qualquer outro motivo, resultando possivelmente em um longo período sem que a pessoa receba a ajuda necessária após o acidente, com consequências potencialmente fatais (Ariunbold; Brito; Leong, 2020).

Em suma, a detecção de quedas é um problema multifacetado e, diante desses desafios, é indispensável a adoção de um sistema inteligente capaz de monitorar e analisar imagens de forma ininterrupta e eficiente. A visão computacional pode superar as limitações das abordagens tradicionais ao capturar informações sobre a cena e distinguir situações de emergência de outras ações por meio da leitura das sequências de eventos (Adhikari, 2019). Além disso, pode interpretar imagens em baixa resolução, abstrair obstáculos e dispensar

o uso de dispositivos vestíveis ou botões de acionamento, proporcionando maior segurança e autonomia às pessoas em situação de risco.

## 2.3 Modelos Aplicados para Reconhecimento de Atividades Humanas

Identificar e compreender ações humanas, também conhecida por Reconhecimento de Atividade Humana (do inglês, HAR - *Human Activity Recognition*), é essencial para diversas aplicações (Kumar; Chauhan; Awasthi, 2024). A capacidade em identificar e interpretar as ações humanas é um tema central em áreas como visão computacional e aprendizado de máquina (Ravuri, 2024). O HAR vem sendo amplamente utilizado em áreas como bem-estar, esportes, saúde, segurança, desempenho esportivo, entre outras (V7Labs, 2022).

Nesse trabalho, o HAR será a base conceitual para a detecção automática de quedas em ambientes domésticos. O objetivo principal do HAR é prever o rótulo da ação de uma pessoa a partir de uma imagem ou vídeo, e um dos maiores desafios é considerar os diversos fatores humanos da cena como atributos físicos, direção e a pose do indivíduo (V7Labs, 2022). Por exemplo, considerando a imagem abaixo de forma isolada, o sistema poderia interpretar equivocadamente que ocorreu uma queda devido à posição deitada, quando, na realidade, a pessoa está apenas descansando.

Figura 2 – Pessoa deitada no sofá de sua residência.

Fonte: Imagem gerada por IA.

Portanto, para uma resposta assertiva, é importante entender o contexto da imagem analisando a sequência de eventos anteriores. Uma queda só pode ser confirmada caso tenha ocorrido uma situação anômala, como um tropeço ou desmaio, resultando em uma aceleração repentina ou ainda em uma posição incomum do corpo.

Estudos com algoritmos de aprendizagem profunda têm se destacado pela eficiência no reconhecimento de ações humanas. De forma geral, para atingir o objetivo proposto por esse trabalho, será necessário detectar a pessoa na cena, analisar a sequência de imagens ao longo do tempo e por fim, classificar se houve ou não uma queda. Diversos algoritmos de aprendizagem profunda pré-treinados se destacam pela alta precisão na análise de movimentos humanos e pela capacidade de adaptação a aplicações específicas.

Para desenvolver um modelo rápido e eficiente no monitoramento de emergências, será utilizado o modelo YOLO (*You Only Look Once*) para a detecção espacial de pessoas, e a técnica LSTM (*Long Short-Term Memory*) para a análise temporal. O YOLO se destaca pela sua velocidade e desempenho em tempo real, enquanto o LSTM é altamente eficaz na análise de sequências temporais, como a detecção de quedas.

### 2.3.1 Detecção de Objetos com YOLO

O YOLO (*You Only Look Once*) é um algorítimo criado em 2015 para tarefas de visão computacional em geral, incluindo detecção de objetos, segmentação de instâncias, estimativa de pose e detecção objetos orientados (OBB).

Neste trabalho, o foco está na detecção de objetos, técnica fundamental da visão computacional utilizada para identificar e localizar objetos dentro de imagens ou vídeos por meio de caixas delimitadoras (*bounding boxes*). Essas caixas correspondem às marcações retangulares que envolvem cada objeto identificado. Esse processo é frequentemente confundido com classificação de imagens ou reconhecimento de objetos, mas a diferença é que, enquanto a classificação prediz a categoria de uma imagem inteira ou de um objeto específico, a detecção de objetos combina localização e classificação simultaneamente, informando o que é o objeto e onde ele está na imagem (Figura 3).

Figura 3 – Detecção de objeto.


Fonte: DataCamp (2023)

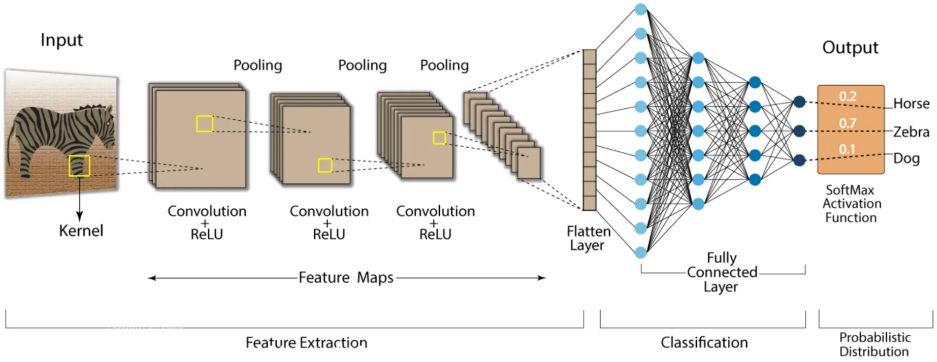
Essa característica de localizar objetos rapidamente, deu origem ao nome "*You Only Look Once*", pois, diferentemente dos métodos tradicionais de detecção que analisam várias partes da imagem em etapas separadas, o YOLO "olha apenas uma vez"para a imagem inteira e realiza as previsões com precisão (Redmon *et al.*, 2016).

O YOLO se destacou em estudos comparativos de performance entre modelos de detecção de objetos como um dos mais rápidos no processamento das imagens e inferência, conforme demonstrado em Hui (2018), Joshi (2025), Fatima *et al.* (2024) e KeyLabs (2023), comprovando sua eficácia para aplicações em tempo real. Além do excelente desempenho, fatores como a facilidade de uso e suporte contínuo da comunidade consolidaram o YOLO como uma das abordagens mais populares em visão computacional, o que foi decisivo para sua escolha neste trabalho.

O algorítimo se beneficia ainda da transferência de aprendizado (*transfer learning*), permitindo ser ajustado (*fine-tunning*) para tarefas específicas e com um conjunto de dados menores, o que o torna altamente adaptável a diferentes tipos de aplicações. Além disso, sua arquitetura leve permite uma inferência rápida e com baixo consumo de energia. O YOLO também se destaca pelo desempenho frente a desafios do mundo real, como variações de luminosidade, oclusões e movimentos rápidos de objetos, características que se adequam perfeitamente à proposta deste trabalho.

Originalmente baseado na arquitetura de Redes Neurais Convolucionais (Figura 4), o YOLO se tornou popular por ser um modelo de código aberto, de fácil acesso e adaptação. Desde então, vem evoluindo continuamente, com novas versões lançadas periodicamente, trazendo melhorias de desempenho e incorporando técnicas mais avançadas de detecção de objetos.

Figura 4 – Rede Neural Convolucional (CNN)

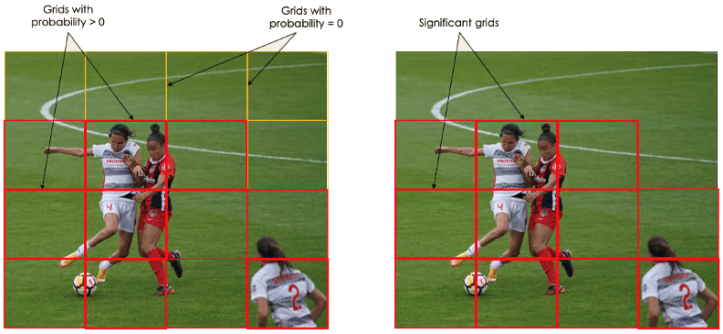



Fonte: Shahriar (2021)

O princípio de funcionamento do modelo para detecção de objetos (YOLO) é baseado em quatro pilares principais: blocos residuais, caixas delimitadoras, Interseção Sobre Uniões (IOU) e supressão não-máxima.

Inicialmente, a imagem de entrada é redimensionada e processada por uma rede neural convolucional (CNN). Nessa rede, os blocos residuais funcionam como "atalhos"que facilitam o fluxo de informações entre camadas, permitindo que a rede aprenda os padrões visuais e acelere o treinamento. A saída da rede é um mapa de características interpretado como uma grade (*grid*) que divide a imagem em células de tamanhos iguais. Cada célula fica responsável por detectar a existência de algum objeto dentro de sua área, prever sua classe e calcular a probabilidade de confiança. As células com as probabilidades maiores que zero são mantidas e as demais descartadas (Figura 5).

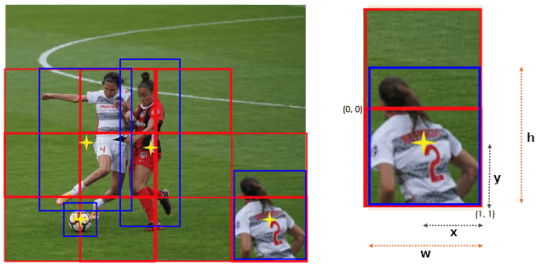
Figura 5 – Divisão da imagem em grade e definição das células válidas




Fonte: DataCamp (2023). Adaptado pelo autor.

Para cada objeto detectado, o YOLO gera uma caixa delimitadora (*bounding box*) (Figura 6) formada pelo nível de confiança, duas coordenadas representando o centro da caixa em relação aos limites da célula da grade (x e y), duas coordenadas com altura e largura da caixa em relação às dimensões da imagem (w e h) e as classes correspondentes.

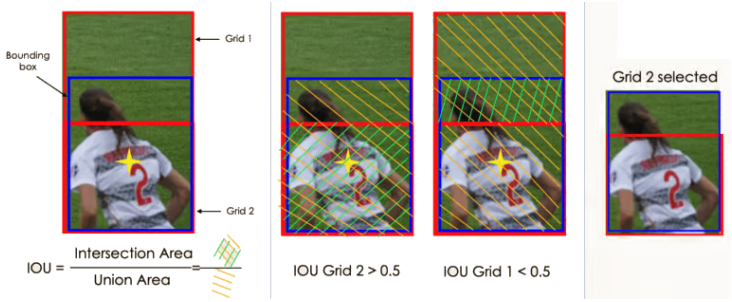
Figura 6 – Caixas delimitadoras (bounding boxes)




Fonte: DataCamp (2023). Adaptado pelo autor.

Em seguida, é aplicado um filtro baseado na Interseção Sobre a União (IOU) para eliminar caixas redundantes que detectam o mesmo objeto e, por fim, realiza uma Supressão Não-Máxima para manter somente a caixa com a maior probabilidade de confiança (Figura 7), garantindo detecções mais precisas.

Figura 7 – Interseção Sobre a União (IOU) e Supressão Não-Máxima

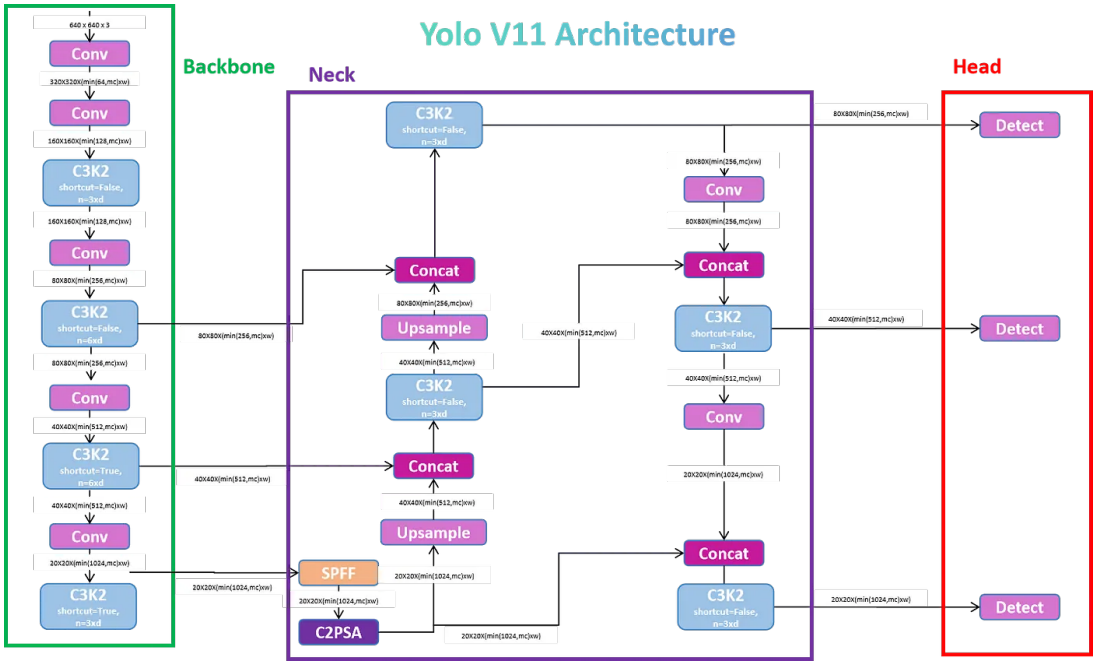



Fonte: DataCamp (2023). Adaptado pelo autor.

Para a detecção de objetos proposta aqui, será utilizado o modelo YOLO v11, que apresenta uma arquitetura híbrida avançada (Figura 8), indo além das tradicionais CNNs das versões mais antigas e de outros modelos do mesmo tipo.

O *backbone* atua como o cérebro do modelo, capturando detalhes importantes das imagens e extraindo características. O *neck*, por sua vez, é a ligação do cérebro (*backbone*) ao resto do sistema. É onde se encontra os mecanismos de atenção para focar em objetos relevantes e o agrupamento inteligente para detectar objetos de tamanhos diferentes, seja pequeno como uma placa de trânsito ou grande como um ônibus. O *head* é a visão, onde são realizadas as previsões. Nesta fase é onde ele informa qual objeto está presente e onde ele se encontra. Esses recursos avançados tornam esse modelo ideal para detecção de objetos em tempo real (Reddy, 2024).

Figura 8 – Arquitetura do YOLO v11.



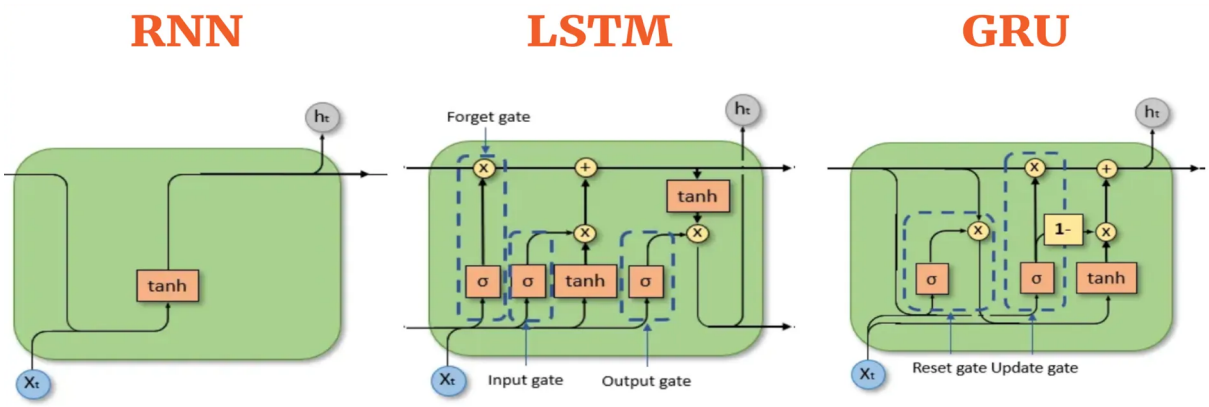

Fonte: Rao (2024)

### 2.3.2 Análise Temporal com Redes LSTM

Desenvolver um modelo para prever eventos futuros sempre foi um desafio, seja em áudio, vídeo ou texto. Por exemplo, na detecção de quedas em vídeos, é necessário entender o que aconteceu ao longo do tempo. Ao observar uma imagem estática com uma pessoa deitada, não é possível afirmar se ela caiu ou está apenas descansando. Para lidar com essas sequências temporais, utilizam-se modelos de redes recorrentes, como *Recurrent Neural Networks* (RNN), *Long Short-Term Memory* (LSTM) e *Gated Recurrent Unit* (GRU) (Figura 9).

A RNN é a forma mais simples, porém possui o problema de desaparecimento de gradientes (*vanishing gradients*) (Sherstinsky, 2020). Ocorre no processo de *back-progragation* quando os gradientes calculados durante o treinamento de redes neurais ficam tão pequenos que dificultam o ajuste dos pesos, impedindo assim o aprendizado, especialmente em sequências mais longas. A LSTM foi criada em 1997, como uma variação da RNN, justamente para resolver esse problema, permitindo a rede memorizar informações relevantes por mais tempo. Já a GRU foi introduzida por Cho *et al.* (2014) e é uma versão modernizada de rede neural recorrente, baseada na LSTM, porém mais leve e rápida, com desempenho comparável, conforme demonstrado no artigo de GeeksForGeeks (2025).

Figura 9 – Arquiteturas de RNN, LSTM e GRU



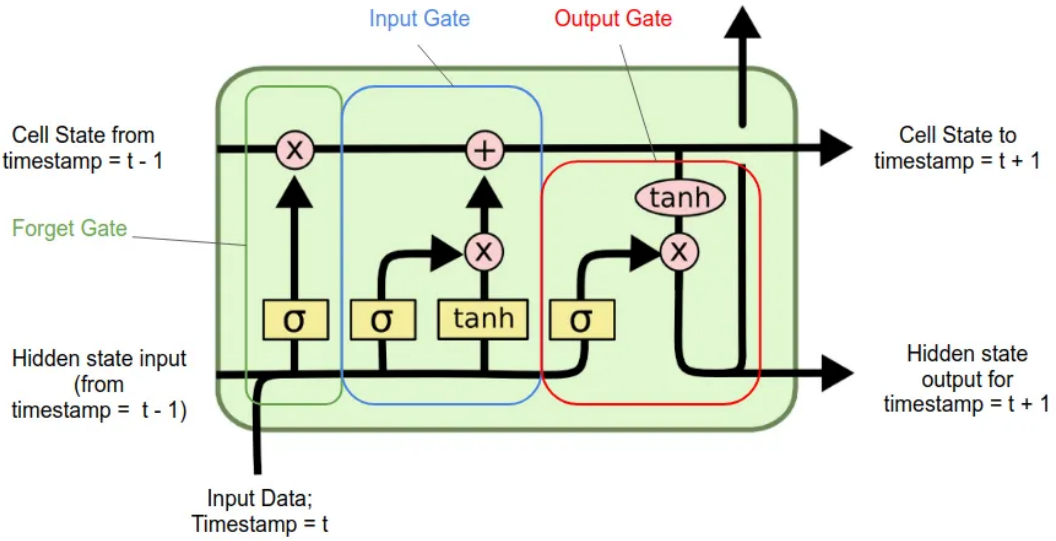

Fonte: Dabass (2024)

Para este trabalho será usada a rede LSTM, pois, em comparação com a rede GRU, aquela lida melhor com conjunto de dados maiores, vídeos mais longos e complexos, e situações em que o contexto anterior é crucial (Idrees, 2025), já que a queda pode acontecer de forma lenta ou rápida. Nos estudos comparativos de Wria e Mohammed (2022) entre LSTM e GRU para Reconhecimento de Atividade Humana (HAR), objeto deste estudo, a LSTM obteve resultados melhores, provando sua robustez para esse tipo de aplicação.

A principal ideia por trás da LSTM é então, permitir que a rede mantenha as informações importantes por longos períodos, e ignore as que não são relevantes. Para

isso, cada célula LSTM possui uma memória de longo prazo (*cell state*) e uma memória de curto prazo (*hiddens state*). A memória de longo prazo lembra o que é importante, enquanto a de curto prazo serve como entrada para a próxima célula ou como saída da rede, por isso o nome "*Long Short-Term Memory*". Essas memórias utilizam um sistema de portas (*gates*) que controlam o que entra (*input gate*), o que sai (*output gate*) e o que deve ser esquecido (*forget gate*) (Vennerød; Kjærran; Bugge, 2021). A estrutura da célula de uma LSTM está detalhada na Figura 10.

Figura 10 – Célula de uma LSTM




Fonte: Ryan (2020)

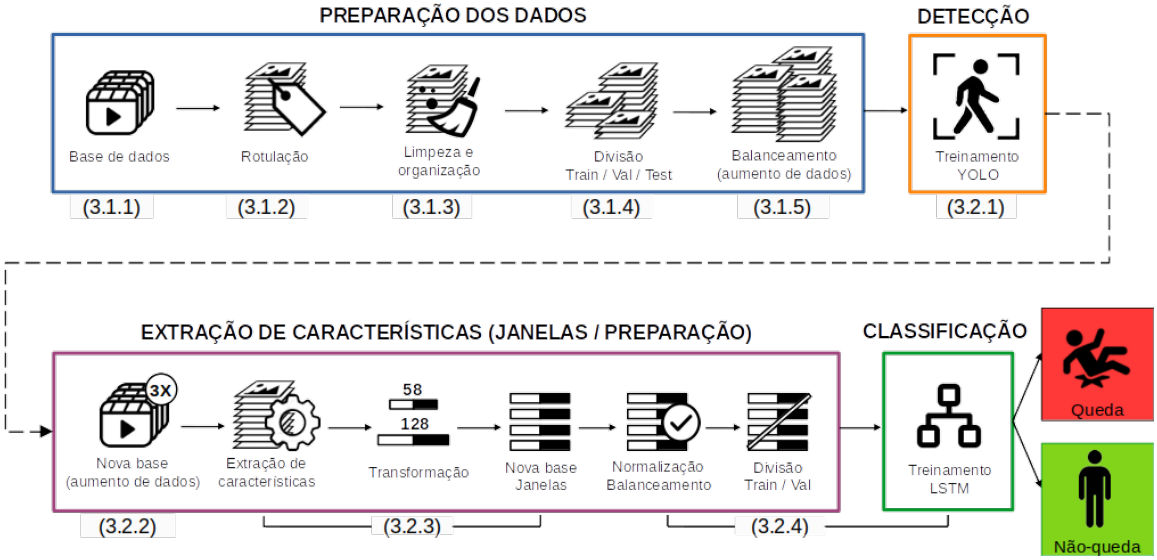
Durante esse processo são realizados cálculos onde a LSTM utiliza funções de ativação sigmóide e tangente hiperbólica (tanh). A função sigmóide, basicamente transforma qualquer valor de entrada (x) em um resultado (y) entre 0 e 1, atuando como um filtro, definindo o que deve ser mantido ou descartado. A função tanh, por sua vez, retorna valores entre -1 e 1, ajudando a regular os valores internos da célula, evitando explosões ou desaparecimento de gradientes, já que os valores são limitados.

Na detecção de quedas, objeto deste trabalho, o LSTM entra em ação logo após a pessoa ser identificada na cena pelo YOLO. Nesse momento o algorítimo analisa a sequência dos movimentos ao longo do tempo, essencial para entender se a pessoa realmente caiu ou apenas se abaixou ou deitou. Como essa análise é feita quadro a quadro, o LSTM é promissor para esse tipo de variação temporal e identificar padrões complexos que ocorrem dentro de alguns segundos.

# 3 METODOLOGIA

Para uma melhor compreensão do funcionamento da proposta, é apresentado o *pipeline* da metodologia adotada para a detecção de quedas (Figura 11). O diagrama mostra a sequência de todas as etapas do processo, desde a obtenção dos vídeos (base de dados) até a classificação final. O *pipeline* ilustra, por exemplo, como o modelo de detecção e estimativa de pose é utilizado para extração de características e, posteriormente, como essas informações são processadas pelo LSTM para identificar o evento de queda. Essa visão geral destaca a integração entre diferentes fases e a lógica da solução proposta. Abaixo de cada bloco do diagrama há o índice da seção que detalha cada um destes elementos.

Figura 11 – Diagrama da solução proposta




Fonte: De autoria própria.

## 3.1 Preparação e rotulação da base de dados

### 3.1.1 Base de dados e organização dos arquivos

A base de dados utilizada neste trabalho foi a GMDCSA-24 (Ekram *et al.*, 2024), pública, composta por 160 vídeos no total, sendo 79 contendo eventos de queda (*fall*) e 81 sem queda (*no_fall*). Para atender às necessidades específicas deste estudo, a base original precisou ser adaptada: os vídeos foram reorganizados em um único diretório e numerados de 1 a 160, sendo de 1 a 79 com eventos de queda e de 80 a 160 com eventos sem queda e, em seguida, rotulados para o modelo YOLO, conforme detalhado na próxima seção.

### 3.1.2 Rotulação dos dados

Os vídeos foram processados na ferramenta de anotação de dados *Computer Vision Annotation Tool* - CVAT (2017), de forma gratuita em ambiente local. Essa plataforma, interativa, de código aberto e baseada na web, foi projetada para otimizar o fluxo de trabalho de rotulação de dados para tarefas de visão computacional e suporta vários tipos de anotação, incluindo caixas delimitadoras (*bounding boxes*) para detecção de objetos, polígonos para segmentação semântica e pontos-chave (*keypoints*) para estimativa de pose, como a utilizada nesse trabalho. Nesse processo foram necessárias duas rotulações distintas:

1. **Rotulação para detecção da ação:** Cada frame foi anotado com uma caixa delimitadora (*bounding box*) e com as classes de ação: *no_fall* (classe 0), *fall* (classe 1) e *attention* (classe 2). A classe *attention* foi utilizada para designar ações que antecedem uma queda e que podem ser consideradas emergências como tonturas, desequilíbrio, desmaios em posição sentada, assim como pode também ser confundidas com quedas, como agachar-se ou sentar-se no chão.

2. **Rotulação para estimativa de pose:** A anotação de pose não permite determinar diversas classes e, por isso, cada frame foi rotulado apenas com a classe *person*. Além da classe, essa rotulagem também era composta por 18 pontos-chave (*keypoints*) que descrevem a articulação do corpo humano e uma caixa delimitadora determinada automaticamente por esses pontos.

### 3.1.3 Padronização das imagens e rótulos

A base de dados de referência para o treinamento foi a com a rotulagem de pose, sendo realizados os seguintes procedimentos:

1. Padronização das imagens para a entrada da rede neural com o tamanho de 640px de largura e altura proporcional.

2. Verificação de integridade para garantir a paridade entre a quantidade de imagens e rótulos entre ambas anotações.

3. Padronização dos rótulos convertendo as classes e valores de visibilidade, previamente no tipo *float*, para inteiro. Além disso, valores negativos foram truncados para zero e valores maiores que um (exceto a visibilidade) truncados para um.

4. As classes da rotulagem de pose (*person*) foram substituídas pelas classes equivalentes da rotulagem de ação (*no_fall, fall ou attention*).

### 3.1.4 Divisão da base de dados

A base de dados, originalmente com vídeos de quatro pessoas distintas, foi organizada de forma que todos fizessem parte do treinamento para maior generalização e robustez do modelo. Ao realizar o treinamento com as características de movimento de indivíduos diferentes, evita-se que o modelo adquira vícios de um tipo de corpo ou movimento, para garantir uma performance mais consistente em dados não vistos de pessoas diferentes. A divisão da base de dados foi realizada da seguinte forma: os 160 vídeos foram separados em 70% (112) para treinamento com YOLO e 30% (48) para testes. Os 112 vídeos de treinamento com YOLO, foram divididos em 70% (80) para treinamento (*train*) e 30% (32) para validação (*val*). A distribuição final ficou de acordo com a Tabela 3.

Tabela 3 – Distribuição dos arquivos da base de dados

| **Pessoa** | **Distribuição** | | **Classe** | **Qtde** | **Total** | **Vídeos** |
|---|---|---|---|---|---|---|
| **1** | YOLO 70% | Images/Labels Train 70% | Fall | 8 | 22 | 001, 003, 005, 007, 009, 011, 013, 015 |
| | | | No Fall | 8 | | 080, 082, 084, 086, 088, 090, 092, 094 |
| | | Images/Labels Val 30% | Fall | 3 | | 002, 004, 006 |
| | | | No Fall | 3 | | 081, 083, 085 |
| | Teste 30% | Vídeos | Fall | 5 | 10 | 008, 010, 012, 014, 016 |
| | | | No Fall | 5 | | 087, 089, 091, 093, 095 |
| **2** | YOLO 70% | Images/Labels Train 70% | Fall | 13 | 34 | 017, 019, 021, 023, 025, 027, 029, 031, 033, 035, 037, 039, 041 |
| | | | No Fall | 11 | | 096, 098, 100, 102, 104, 106, 108, 110, 112, 114, 116 |
| | | Images/Labels Val 30% | Fall | 5 | | 018, 020, 022, 024, 026 |
| | | | No Fall | 5 | | 097, 099, 101, 103, 105 |
| | Teste 30% | Vídeos | Fall | 7 | 14 | 028, 030, 032, 034, 036, 038, 040 |
| | | | No Fall | 7 | | 107, 109, 111, 113, 115, 117, 118 |
| **3** | YOLO 70% | Images/Labels Train 70% | Fall | 11 | 30 | 042, 044, 046, 048, 050, 052, 054, 056, 058, 060, 062 |
| | | | No Fall | 11 | | 119, 121, 123, 125, 127, 129, 131, 133, 135, 137, 139 |
| | | Images/Labels Val 30% | Fall | 4 | | 043, 045, 047, 049 |
| | | | No Fall | 4 | | 120, 122, 124, 126 |
| | Teste 30% | Vídeos | Fall | 6 | 13 | 051, 053, 055, 057, 059, 061 |
| | | | No Fall | 7 | | 128, 130, 132, 134, 136, 138, 140 |
| **4** | YOLO 70% | Images/Labels Train 70% | Fall | 8 | 26 | 063, 065, 067, 069, 071, 073, 075, 077 |
| | | | No Fall | 10 | | 141, 143, 145, 147, 149, 151, 153, 155, 157, 159 |
| | | Images/Labels Val 30% | Fall | 4 | | 064, 068, 072, 079 |
| | | | No Fall | 4 | | 142, 144, 146, 148 |
| | Teste 30% | Vídeos | Fall | 5 | 11 | 066, 070, 074, 076, 078 |
| | | | No Fall | 6 | | 150, 152, 154, 156, 158, 160 |

Fonte: De autoria própria.

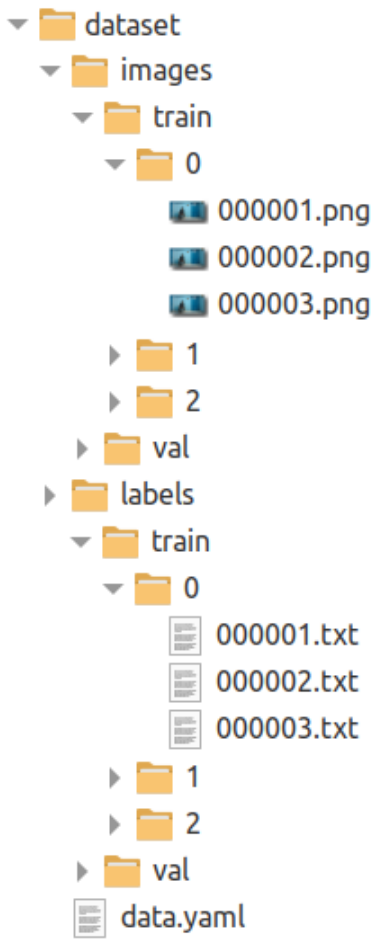
### 3.1.5 Distribuição e consolidação da base de dados

Inicialmente a base de dados a ser utilizada no treinamento apresentava um desbalanceamento de classes significativo, com a classe *no_fall* sendo majoritária com

71.2%, a classe *fall* com 23.6% e a classe *attention* com 5.2%. Para mitigar essa discrepância e evitar um modelo tendencioso, foi aplicada uma técnica de aumento de dados sintéticos (*data augmentation*), através da biblioteca Albumentations, nas classes minoritárias (*fall e attention*) do conjunto de treino. Esse recurso expande a base de dados existente, criando artificialmente novas imagens e/ou vídeos para treinamento, expondo o modelo a uma maior variedade de cenários, e consequentemente, melhorando a capacidade de generalização e reduzindo o risco de *overfitting*.

Nesse processo foram geradas novas amostras da mesma base, porém com efeitos visuais aplicados, resultando em uma distribuição de classes mais equilibrada no conjunto de treinamento final, com a classe *no_fall* representando 41.8%, a classe *fall* com 27.7% e a classe *attention* com 30.5%. Após o balanceamento da base de dados, os arquivos que estavam separados por vídeos, foram redistribuídos por classes, de acordo com a Figura 12 abaixo:

Figura 12 – Estrutura de pastas da base de dados




Fonte: De autoria própria.

Todo esse tratamento resultou em uma única base de dados capaz de, simultaneamente, detectar uma pessoa, estimar sua pose e classificar a ação no frame.

## 3.2 Treinamento do modelo

### 3.2.1 Detecção e estimativa de pose

Na primeira fase, o treinamento para detecção da pessoa em cena e estimativa da pose foi realizado com o YOLO, no modelo pré-treinado yolov11n.pt. Essa versão é uma variante que estende a funcionalidade principal do YOLO v11 (Jocher; Qiu; Chaurasia, 2023) para realizar estimativa de pose humana, baseada nas coordenadas das articulações esqueléticas em uma única inferência. O modelo do tipo n (nano) é o menor, mais leve e mais rápido de todos. Excelente para prototipagem rápida e para aplicações em tempo real com recursos limitados, mas por outro lado, é menos preciso que as outras versões mais robustas.

O treino foi configurado com 100 épocas, 16 batch, 4 workers, amp *True* e cache *True*. Desse processo, foi gerado um novo modelo treinado com a base de dados deste trabalho e, com o melhor desempenho, chamado de best.pt. Esse modelo foi utilizado nas fases seguintes para fazer inferências em cada frame dos vídeos.

### 3.2.2 Geração de nova base com aumento dos dados

Em seguida, devido à necessidade de novos dados para a análise temporal, foi realizado um novo aumento dos dados. Para essa base, foi criada três variações de todos os 160 vídeos, todos invertidos horizontalmente, fornecendo assim uma característica única, totalizando 480 vídeos.

### 3.2.3 Extração de características e geração de janelas deslizantes

Para cada pessoa detectada nos vídeos, foi extraído um vetor de características, originalmente com 58 valores composto por: 4 valores de caixa delimitadora (*bounding box*) e 54 valores de *keypoints* (18 coordenadas x, y normalizadas e confiança). Esses valores foram então transformados para maior robustez do modelo:

1. **Características estáticas:** Os 4 pontos foram transformados em 10, derivados da caixa delimitadora (x, y, x, y, centro x/y, largura, altura, área, *aspect ratio*), normalizadas pelas dimensões do *frame* que, juntamente com os 54 pontos das articulações do esqueleto, totalizou 64 pontos-chave.

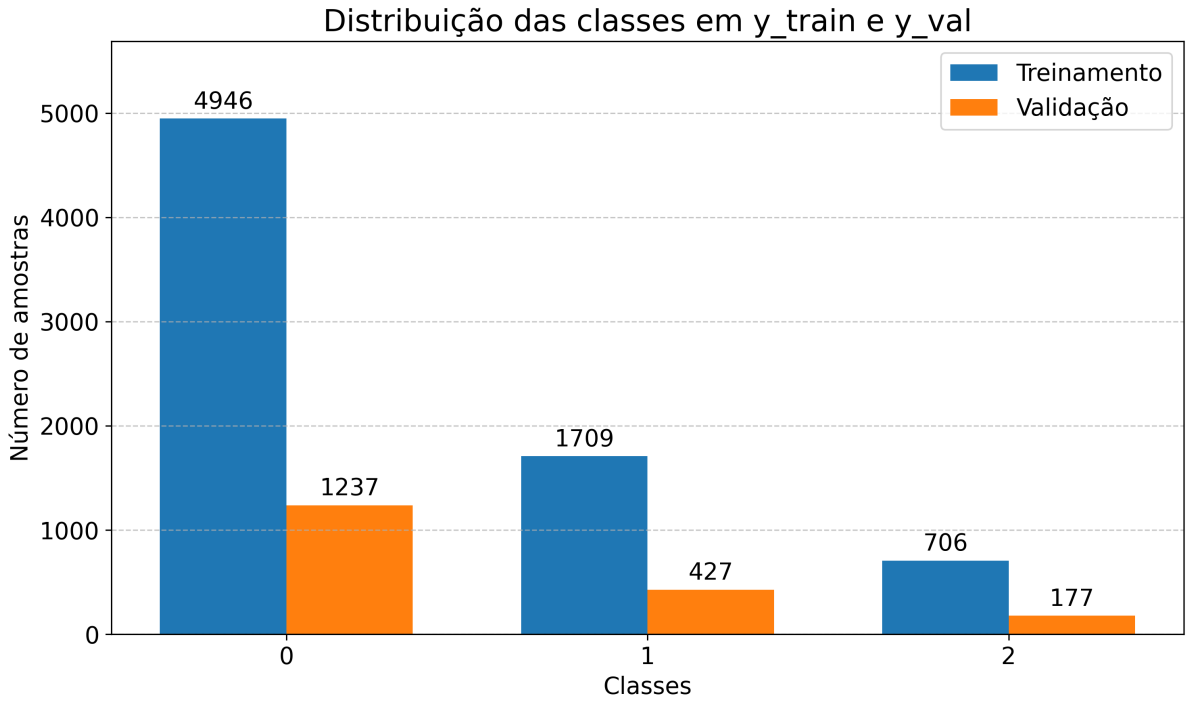
2. **Características dinâmicas:** Para capturar o movimento, foi calculada a diferença vetorial entre as características estáticas do *frame* atual com as do *frame* anterior, representando assim a velocidade, totalizando 64 pontos-chave.

Dessa forma, o vetor final de cada *frame* foi composto por 128 dimensões.

Esses vetores foram agrupados em janelas deslizantes de 30 frames de duração para este experimento (equivalente à 1 segundo dos vídeos da base deste trabalho). Para otimizar a geração dos dados, foi utilizado um passo (*stride*) de 10 *frames*, ou seja, uma nova janela era criada a cada 10 *frames*. O rótulo final de cada janela foi determinado por uma regra de prioridade: uma janela era rotulada com a classe *fall* se esse evento ocorresse em 8 *frames* consecutivos ou com a classe *attention* se repetisse em 5 *frames* seguidos.

Esse processo resultou em uma nova base de dados de 9202 janelas, obtidas da base com 480 vídeos, com as classes distribuídas de acordo com a Figura 13. Essa divisão corresponde a 80% dos dados usados no treinamento e 20% na validação da classificação temporal.

Figura 13 – Distribuição das classes



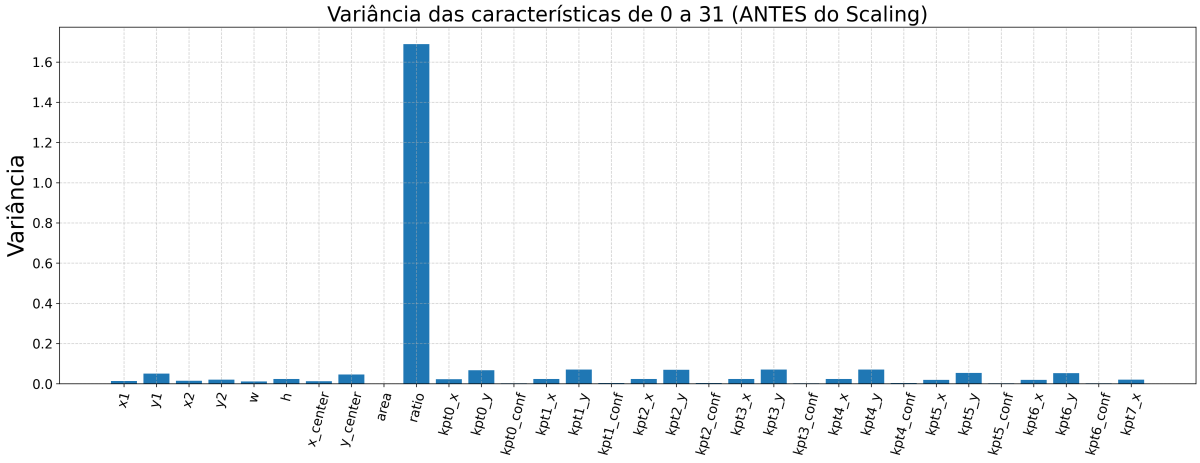

Fonte: De autoria própria.

Nessa fase foi utilizada a biblioteca Numpy para a gravação, carregamento e leitura dos vetores extraídos dos frames e formação das janelas. Foram criados arquivos no formato .npy, para armazenar os arrays de forma binária e eficiente, incluindo não apenas os valores numéricos, mas também informações sobre o tipo de dados e as dimensões da estrutura. Esse formato é muito mais rápido que arquivos de texto tradicionais como .csv ou .txt, pois ele mantém a organização do array em memória. O artigo de Sarkar (2018) detalha outras características positivas desse formato.

### 3.2.4 Treinamento do modelo de classificação temporal

Antes do treinamento, foi identificado que a variância da característica *aspect_ratio* era significativamente maior que as outras, conforme visto na Figura 14.

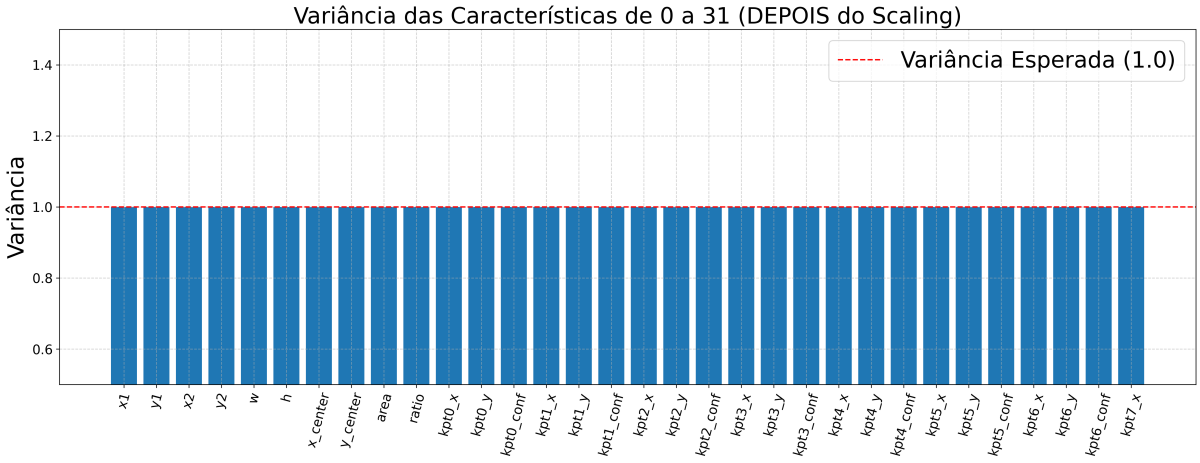
Figura 14 – Amostra da variância antes da padronização




Fonte: De autoria própria.

Para garantir que todas as característica contribuíssem de forma equilibrada, toda a base de dados de janelas foi padronizada utilizando o *StandardScaler* da biblioteca Scikit-learn, ajustando cada característica pra ter uma média 0 e variância 1, conforme mostrada na Figura 15.

Figura 15 – Amostra da variância depois da padronização




Fonte: De autoria própria.

Para lidar com o desbalanceamento de classes na base de dados de janelas, Figura 13, foram calculados os pesos correspondentes para cada classe usada no treinamento,

atribuindo uma penalidade maior a erros nas classes minoritárias. Os pesos atribuídos foram: 0.50 para a classe 0 (*no_fall*), 1.44 para a classe 1 (*fall*) e 3.47 para a classe 2 (*attention*).

Por fim, foi realizado o treinamento LSTM, recurso este proveniente da biblioteca TesorFlow/Keras, projetada para prototipação rápida e experimentação com redes neurais, caracterizada por ser simples e modular. Nessa etapa foi utilizada uma arquitetura de rede neural recorrente do tipo LSTM Empilhada (*Stacked LSTM*), composta por duas camadas de 64 e 32 unidades, com regularização utilizando um *dropout* e *recurrent_dropout* de 0.3. Esse modelo também utilizou uma camada de saída do tipo *Dense* com ativação *softmax*, otimização *Adam* e função de perda *categorical_crossentropy*. O treinamento foi realizado por até 200 épocas, com um *callback* de *early stopping* monitorando a perda de validação com uma paciência de 10 épocas para evitar *overfitting* e salvar o modelo de melhor performance.

# 4 ANÁLISE E RESULTADOS

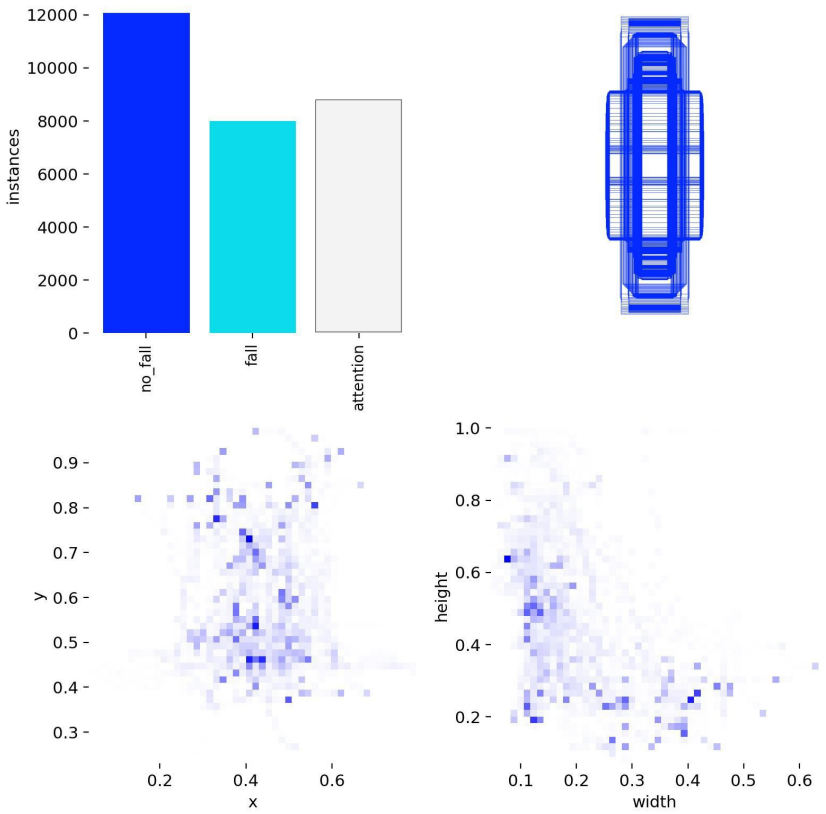
Após a conclusão da fase de treinamento, tanto no modelo de Detecção de Objetos com YOLO quanto no de Análise Temporal com Redes LSTM, o modelo com o melhor desempenho no conjunto de validação foi submetido à uma avaliação detalhada. A análise dos resultados, em ambos os modelos, foi dividida basicamente em três partes: a avaliação do comportamento do modelo durante o processo de treinamento, análise quantitativa do seu desempenho de classificação no conjunto de dados de validação e a análise qualitativa dos modelos finais de inferência.

## 4.1 Modelo de detecção e estimativa de pose

### 4.1.1 Análise exploratória dos dados

A distribuição final das classes da base de dados já aumentada para o treinamento com o YOLO, está ilustrada na Figura 16. Nesse gráfico observa-se que as classes estão bem equilibradas, com aproximadamente 12 mil instâncias na classe *no_fall*, 8 mil na classe *fall* e 9 mil na classe *attention*.

Figura 16 – Anotações da base de dados utilizada no YOLO

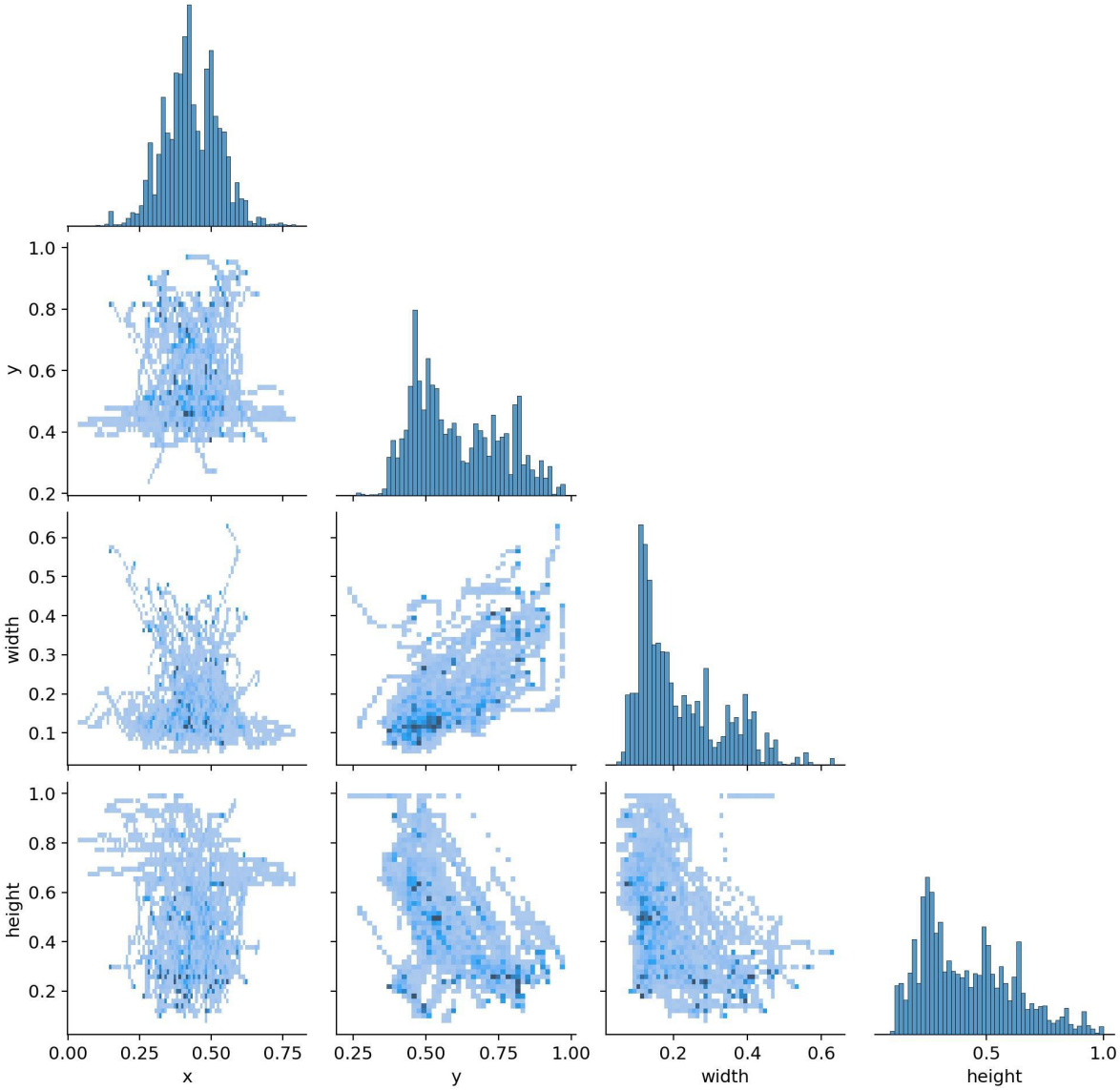



Fonte: De autoria própria.

Os gráficos na parte inferior dessa mesma figura, do lado esquerdo, indicam que as pessoas nas imagens estavam localizadas de forma adequada e, do lado direito, indicam uma correlação visual entre as classes e as dimensões das caixas delimitadoras (*bounding boxes*), com um pouco mais de concentração na detecção da posição em pé (*no_fall*) do que deitada (*fall*), podendo ser predominado pela maior quantidade de instâncias *no_fall*.

O gráfico de correlações entre as variáveis (correlograma) na Figura 17 reforça essas observações, mostrando as distribuições e relações entre as coordenadas x e y de posicionamento, e as dimensões de largura (*width*) e altura (*height*) das caixas delimitadoras.

Figura 17 – Correlação entre as variáveis da base de dados



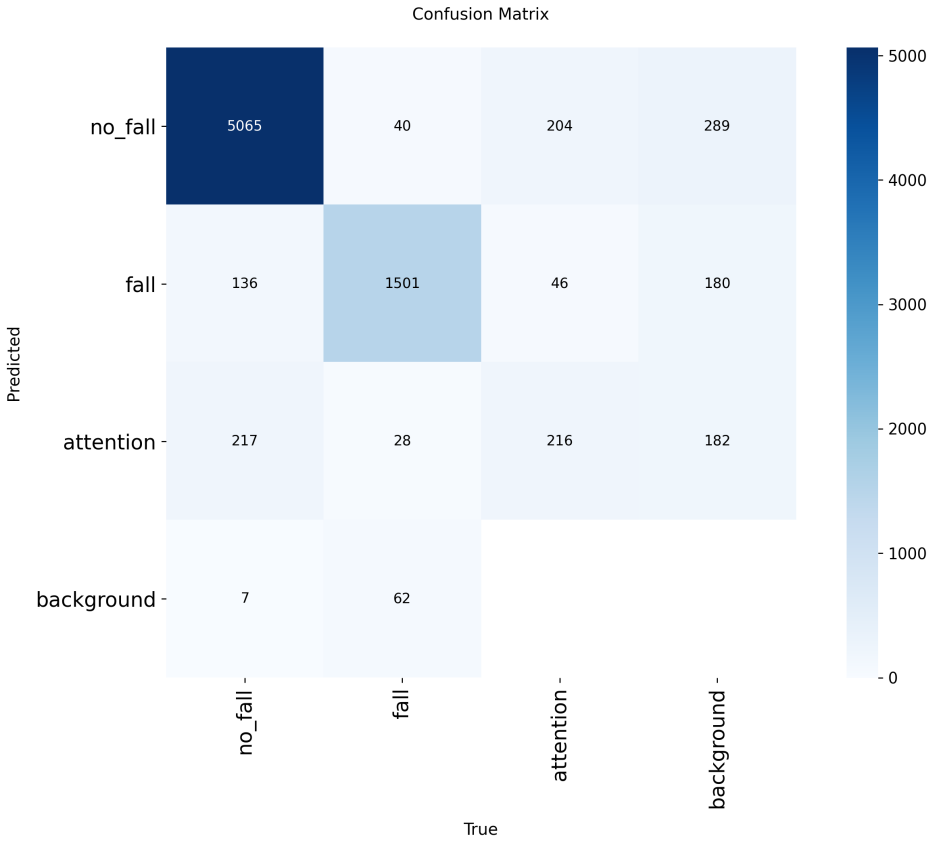

Fonte: De autoria própria.

### 4.1.2 Análise da acurácia do modelo

A matriz de confusão fornece um resumo detalhado dos erros e acertos na classificação do modelo treinado com o YOLO (Figura 18). De forma geral, o modelo demonstra uma alta capacidade de distinguir corretamente as classes *no_fall* com 93% e *fall* com 92%. Porém, a classe *attention*, teve uma taxa de acerto de apenas 46% , sendo incorretamente classificada como *no_fall* em 44% das instâncias, constantemente confundida por ser uma fase ambígua e transitória. A baixa performance nessa classe impacta diretamente a acurácia geral do modelo, que ficou em 81%. No entanto, a alta sensibilidde para as classes *no_fall* e *fall* confirma a robustez do modelo na distinção desses eventos.

O erro mais crítico, que é classificar uma queda (*fall*) como uma não queda (*no_fall*), ou seja, um falso negativo, ocorreu em 2% dos casos, e classificar uma não queda (*no_fall*) como queda (*fall*), um falso positivo, ocorreu em 3% das instâncias. Dada a criticidade do evento e às configurações adotadas neste trabalho, esse resultado foi considerado promissor para esse modelo.

Figura 18 – Matriz de confusão




Fonte: De autoria própria.

Cálculos das acurácias do modelo baseados na matriz de confusão:

$$\text{Acurácia (no\_fall) \%} = \frac{\text{Verdadeiros Positivos (no\_fall)}}{\text{Total Real (no\_fall)}} = \frac{5065}{5425} \approx 93.36\%$$

$$\text{Acurácia (fall) \%} = \frac{\text{Verdadeiros Positivos (fall)}}{\text{Total Real (fall)}} = \frac{1501}{1631} \approx 92.03\%$$

$$\text{Acurácia (attention) \%} = \frac{\text{Verdadeiros Positivos (attention)}}{\text{Total Real (attention)}} = \frac{216}{466} \approx 46.35\%$$

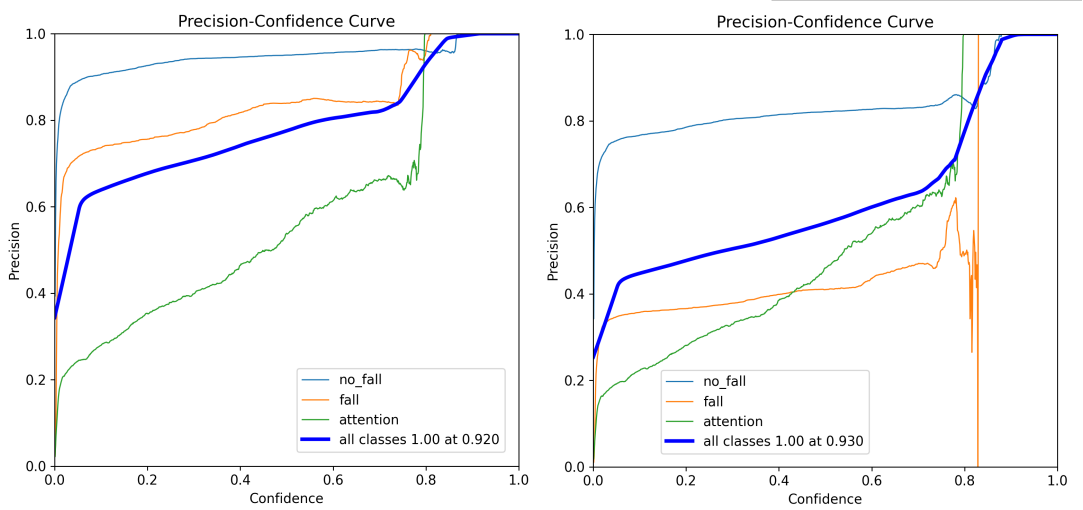
$$\text{Acurácia Geral (\%)} = \frac{\text{Total de Acertos}}{\text{Total de Amostras}} = \frac{5065 + 1501 + 216}{8355} = \frac{6782}{8355} \approx 81.17\%$$

### 4.1.3 Performance geral na detecção e estimativa da pose

As curvas de precisão, *recall* e *F1-Score* em função do limiar de confiança, detalham o comportamento do modelo, conforme podemos ver nos gráficos a seguir, sempre com a caixa delimitadora à esquerda e pose à direita.

No gráfico de precisão *versus* confiança (Figura 19), a precisão para as classes *no_fall* e *fall* se mantém constantemente alta (acima de 50%), mesmo em limiares de confiança mais baixos. A classe *attention* demonstra uma precisão menor, indicando que predições para esta classe são menos confiáveis.

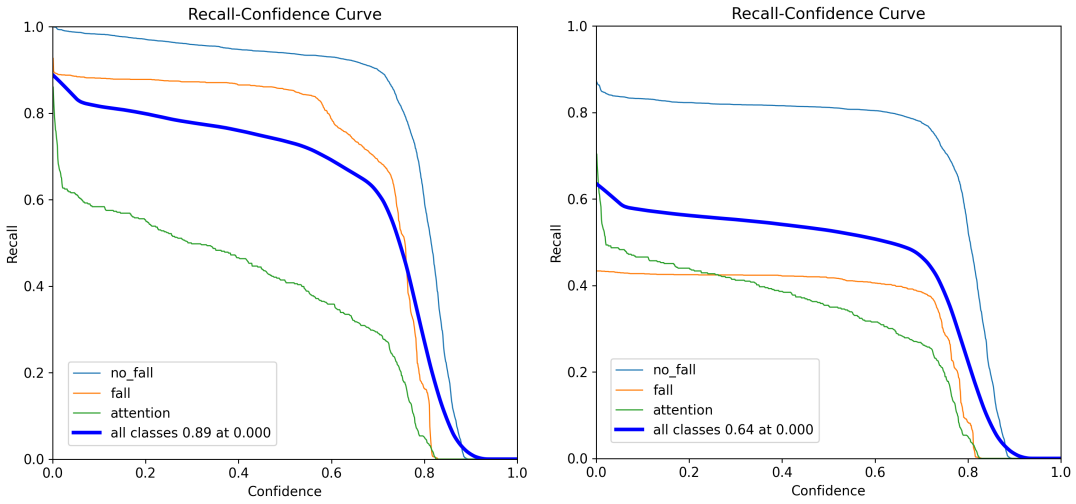
Figura 19 – Precisão *versus* Confiança




Fonte: De autoria própria.

A curva de *recall versus* confiança mostra que, à medida que o limiar de confiança aumenta, o modelo se torna mais seletivo e, consequentemente, ignora mais detecções, diminuindo o *recall*. Para garantir um alto *recall* (detectar a maioria das quedas), um limiar de confiança mais baixo seria necessário, ao custo de uma precisão menor (mais falsos positivos).

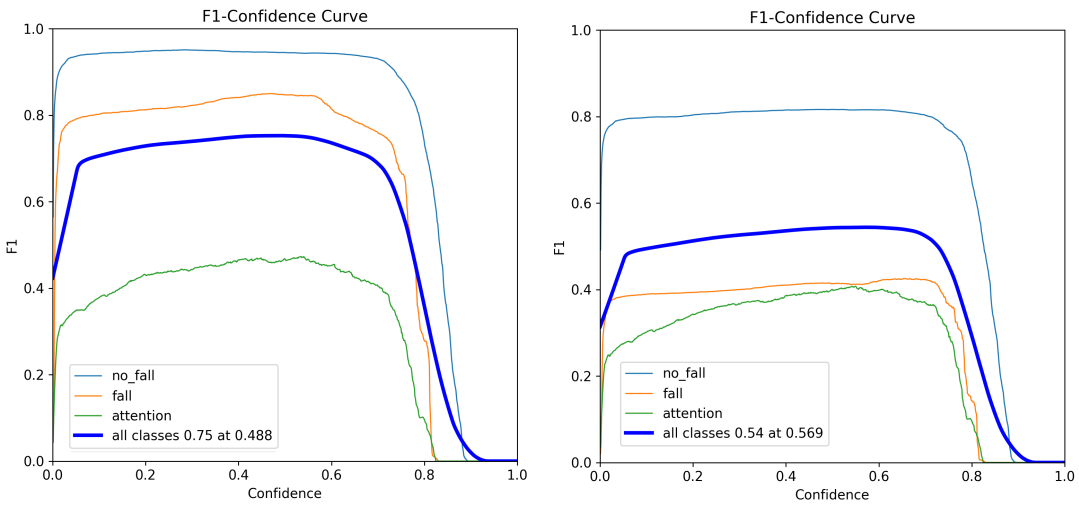
Figura 20 – *Recall versus* Confiança




Fonte: De autoria própria.

O *F1-Score*, que representa o equilíbrio entre precisão e *recall*, atinge seu valor máximo para todas as classes da detecção da caixa em um limiar de confiança de aproximadamente 50%. Isso sugere que o ponto ótimo de operação do modelo, balanceando falsos positivos e falsos negativos, se encontra em um limiar de confiança intermediário.

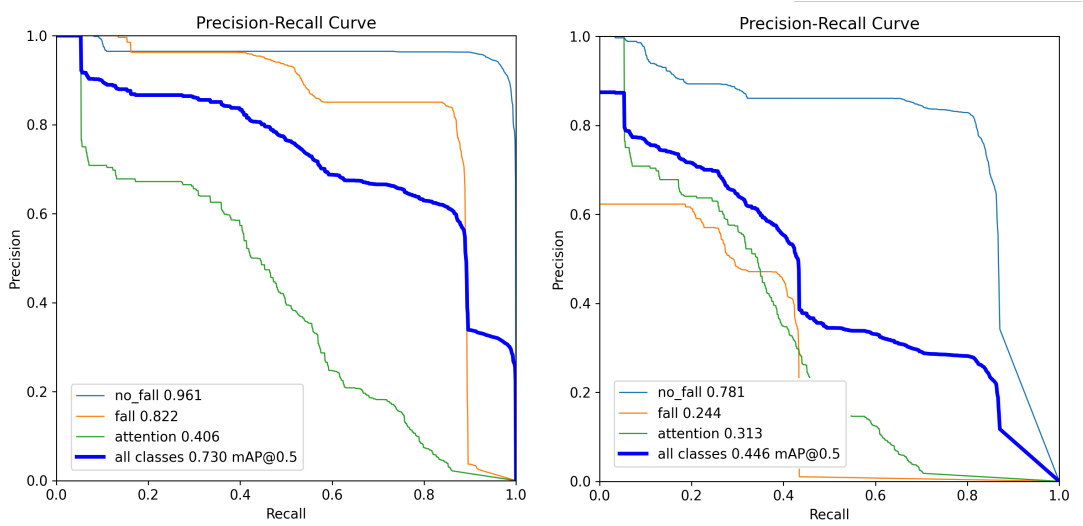
Figura 21 – *F1-Score versus* Confiança




Fonte: De autoria própria.

O gráfico de precisão *versus recall* resume a performance do modelo YOLO. O mAP@0.5 (*Mean Average Precision com IoU de 0.5*) para todas as classes foi de 0.730. Individualmente, a classe *no_fall* obteve o melhor resultado com 0.961, seguida por *fall* com 0.822 e *attention* com 0.406, confirmando as observações da matriz de confusão.

Figura 22 – Precisão *versus Recall*




Fonte: De autoria própria.

Analisando as métricas obtidas pela avaliação do melhor modelo do treinamento (médias de performance), observa-se que a performance geral do modelo para detecção da pessoa por caixas delimitadoras (*bounding boxes)* é sólida, apresentando uma boa capacidade de acertos na localização e definição da classe do objeto, como pode ser visto em mAP50 0.73 e um mAP50-95 de 0.503 (mais rigorosa). Já na estimativa de pose, a capacidade de acertos ficou relativamente baixa com um mAP50 de 0.44 e mAP50-95 de 0.15, o que é esperado, pois a localização de 18 pontos-chave com precisão é uma tarefa muito mais complexa do que uma caixa.

Tabela 4 – Validação do modelo de detecção e estimativa de pose

| **Classe** | **Imagens** | **Instâncias** | **Box** | | | | **Pose** | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | **P** | **R** | **mAP50** | **mAP50-95** | **P** | **R** | **mAP50** | **mAP50-95** |
| all | 7522 | 7522 | 0.770 | 0.739 | 0.730 | 0.503 | 0.588 | 0.514 | 0.446 | 0.155 |
| no_fall | 5425 | 5425 | 0.950 | 0.940 | 0.961 | 0.736 | 0.825 | 0.806 | 0.781 | 0.285 |
| fall | 1631 | 1631 | 0.839 | 0.857 | 0.822 | 0.450 | 0.417 | 0.410 | 0.244 | 0.042 |
| attention | 466 | 466 | 0.522 | 0.418 | 0.407 | 0.322 | 0.521 | 0.324 | 0.314 | 0.140 |

Fonte: De autoria própria.

Na análise por classe, a *no_fall* apresentou um desempenho excelente, se destacando pela alta precisão de 0.95, *recall* de 0.94 e mAP50 de 0.961, mostrando ser extremamente confiável para identificar atividades normais, com poucos alarmes falsos. A classe *fall*

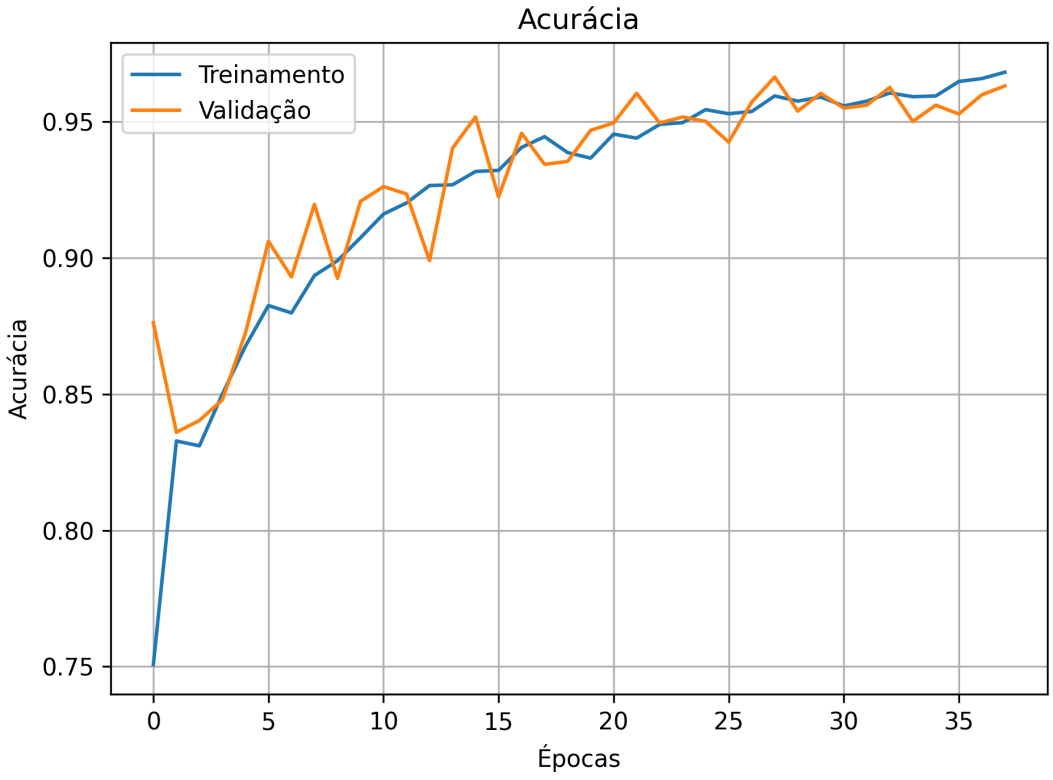
desempenhou-se bem, com uma precisão de 0.839, um *recall* de 0.857 e um mAP50 de 0.822, correto na maioria das vezes. A classe *attention*, por sua vez, obteve um desempenho fraco com precisão de 0.522, *recall* de 0.418 e mAP50 de 0.407, claramente a classe mais desafiadora, com dificuldade em identificar e classificar corretamente essa ação.

## 4.2 Modelo de classificação temporal

### 4.2.1 Análise da acurácia e perda (*loss*)

Os gráficos de acurácia (Figura 23) e perda (Figura 25) ao longo das épocas de treinamento da classificação temporal fornecem informações importantes sobre o processo de aprendizagem do modelo. Na acurácia, tanto no treinamento quanto na validação, apresentaram um crescimento acentuado nas épocas iniciais, estabilizando em valores elevados. Além disso, as duas curvas se mantiveram muito próximas durante todo o processo, frequentemente se cruzando, indicando assim uma excelente capacidade de generalização do modelo.

Figura 23 – Acurácia do modelo de classificação temporal

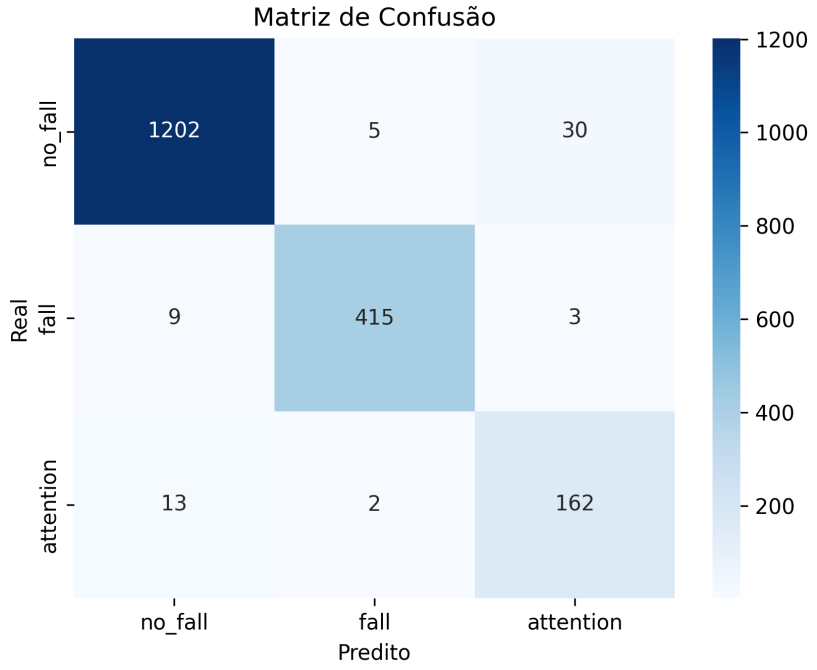



Fonte: De autoria própria.

A matriz de confusão (Figura 24), detalha erros e acertos do modelo para cada uma das classes, sendo que os acertos foram majoritários nas predições realizadas, resultando em uma acurácia geral de 96.6% para as 1841 amostras de validação.

As classes *no_fall* e *fall* demonstraram um desempenho excepcional nas tarefas mais críticas, com baixos índices de falso negativos e positivos. A classe *attention*, embora tenha apresentado o maior erro na predição, especialmente com a classe *no_fall*, obteve um desempenho satisfatório com 162 acertos e 33 erros.

Figura 24 – Matriz de confusão do modelo de classificação temporal




Fonte: De autoria própria.

Cálculos das acurácias do modelo baseados na matriz de confusão:

$$\text{Acurácia (no\_fall) \%} = \frac{\text{Verdadeiros Positivos (no\_fall)}}{\text{Total Real (no\_fall)}} = \frac{1202}{1237} \approx 97.17\%$$

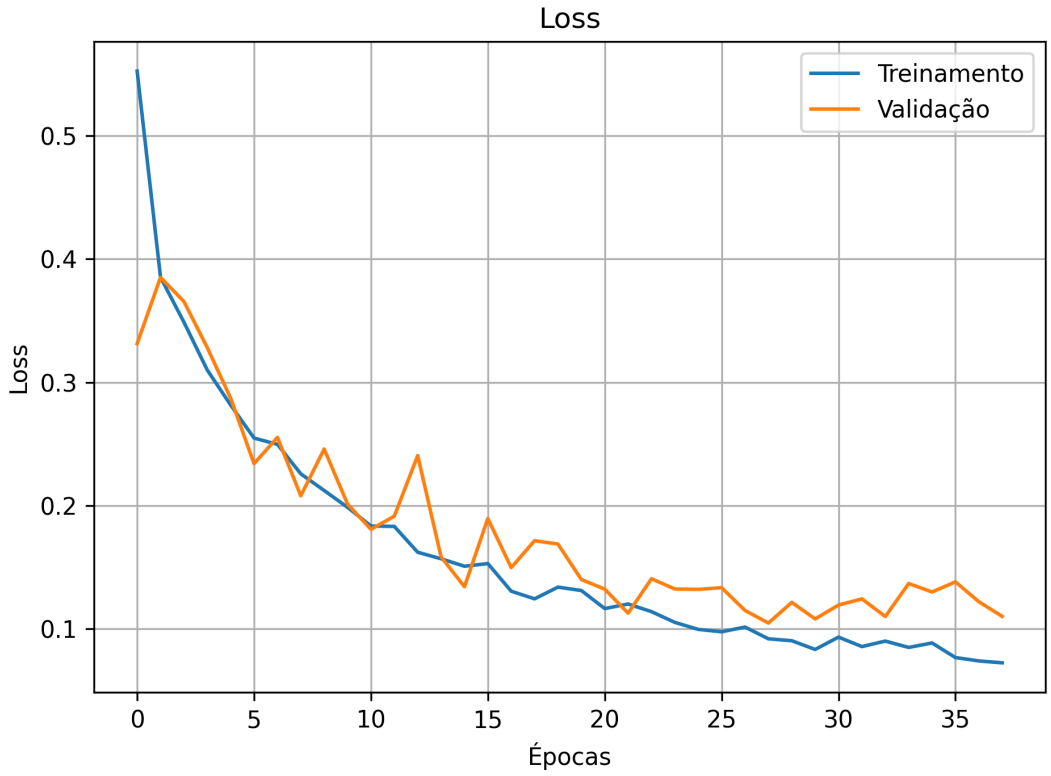
$$\text{Acurácia (fall) \%} = \frac{\text{Verdadeiros Positivos (fall)}}{\text{Total Real (fall)}} = \frac{415}{427} \approx 97.19\%$$

$$\text{Acurácia (attention) \%} = \frac{\text{Verdadeiros Positivos (attention)}}{\text{Total Real (attention)}} = \frac{162}{177} \approx 91.53\%$$

$$\text{Acurácia Geral (\%)} = \frac{\text{Total de Acertos}}{\text{Total de Amostras}} = \frac{1202 + 415 + 162}{1841} = \frac{1779}{1841} \approx 96.63\%$$

Uma observação similar pode ser vista no gráfico de perda (*loss*). A perda de treinamento e de validação diminuíram de forma consistente e convergiram para valores baixos, sem apresentar uma divergência significativa. A falta de um espaçamento crescente entre as curvas de treinamento e validação demonstra que as técnicas de regularização empregadas, como o *dropout*, foram essenciais para prevenir o *overfitting*, resultando em um modelo robusto. O treinamento foi interrompido por volta da época 37, indicando que o modelo atingiu sua performance ótima sem a necessidade de um número excessivo de iterações e consumo de recursos.

Figura 25 – Perda (*Loss*) do modelo de classificação temporal



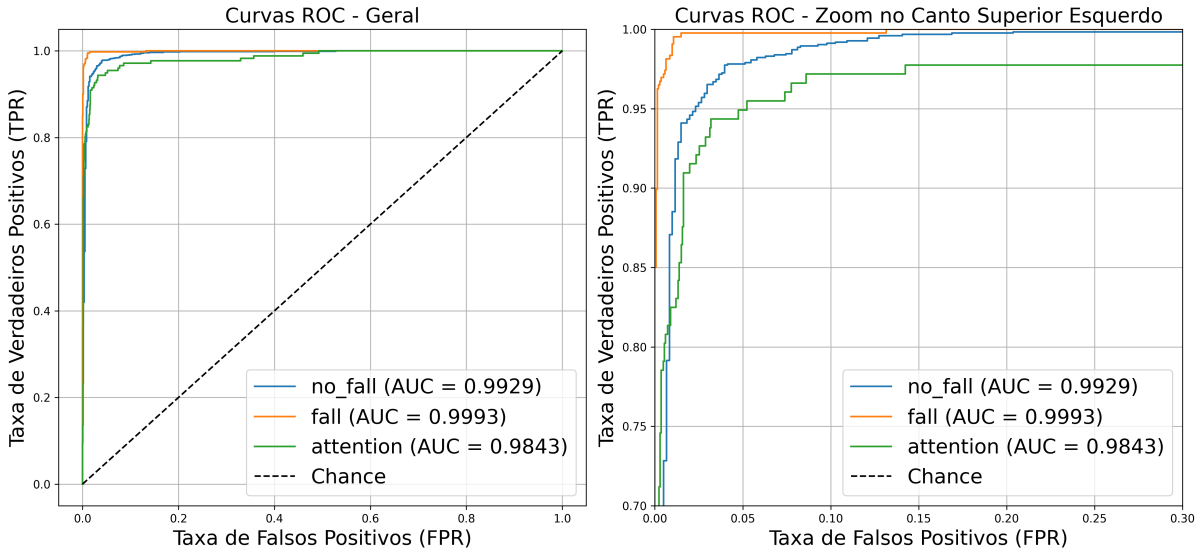

Fonte: De autoria própria.

### 4.2.2 Performance da classificação temporal

A performance do modelo de classificação temporal, nos dados de validação, é avaliada através da análise da matriz de confusão, conforme já detalhada na seção anterior, e da curva ROC (*Receiver Operating Characteristics*).

A análise da curva ROC (*Receiver Operating Characteristics*) indica a alta capacidade do modelo de distinguir entre as classes. A Área Sob a Curva (AUC), que quantifica essa capacidade de distinção, apresentou valores excelentes: classe *no_fall* com 0.9929, *fall* com 0.9993 e *attention* com 0.9843.

Figura 26 – Curva ROC do modelo de classificação temporal




Fonte: De autoria própria.

O resultado da classe *fall* indica um desempenho praticamente perfeito (AUC = 0,9993) na identificação de quedas em relação às demais atividades. O gráfico também mostra que o modelo sustenta uma alta Taxa de Verdadeiros Positivos (TPR) mesmo com uma Taxa de Falsos Positivos (FPR) mínima. Essa característica é ideal para uma aplicação prática, pois minimiza a ocorrência de alarmes falsos sem sacrificar a detecção de eventos de quedas reais.

## 4.3 Análise qualitativa em vídeos de teste

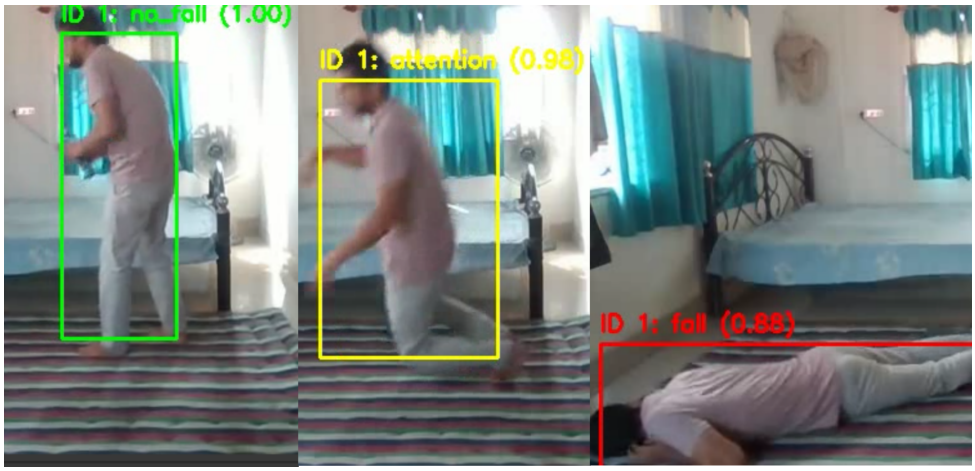
Uma análise qualitativa foi realizada para avaliar a performance do pipeline implementado neste estudo (YOLO e LSTM) em vídeos de teste que não foram utilizados em nenhuma etapa do treinamento (os 30% para testes separados no início do estudo). O objetivo é demonstrar a capacidade do modelo em interpretar o contexto temporal das ações do mundo real. Dois cenários de teste representativos são apresentados a seguir:

### 4.3.1 Estudo de caso 1: Sequência completa de uma queda

O primeiro cenário, Figura 27, avalia a resposta para um evento de queda completo. O vídeo exibe uma pessoa inicialmente em pé, realizando uma atividade normal e classificado como *no_fall*. Em seguida, a pessoa se desequilibra e inicia o movimento de queda, momento em que o modelo corretamente classifica como *attention*. Por fim, quando a pessoa atinge o chão, a classificação é definida como *fall*, e o alerta é mantido enquanto ela permanece na posição de queda.

A identificação correta das classes durante os movimentos, principalmente na ação transitória *attention*, demonstra que o modelo aprendeu os padrões temporais característicos de uma queda não intencional, confirmando sua eficácia nessa análise.

Figura 27 – Teste de vídeo (1) do modelo final

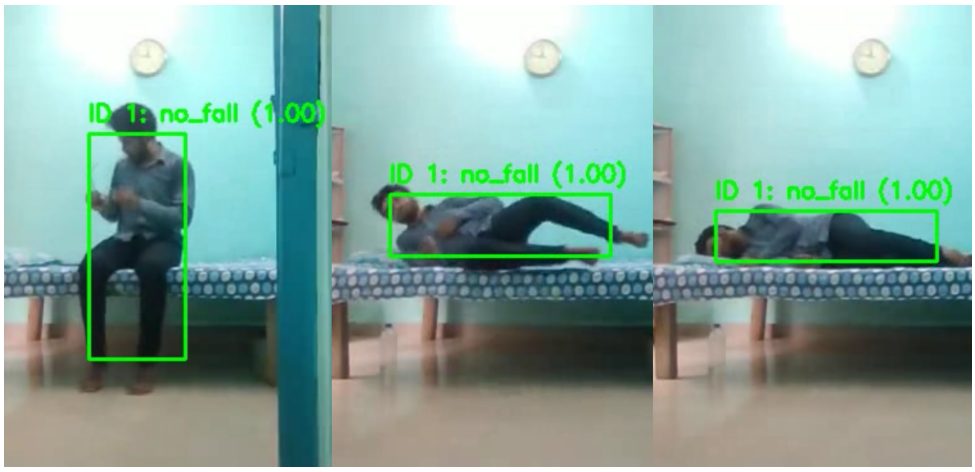



Fonte: De autoria própria.

### 4.3.2 Estudo de caso 2: Sequência ao deitar-se em uma cama

O segundo cenário, Figura 28, avalia a robustez do modelo contra falsos positivos, com uma ação cotidiana como deitar-se para dormir, que possui características semelhantes a uma queda. No vídeo, a pessoa está sentada na cama e, em seguida, deita-se de forma natural. Apesar da transição de uma posição vertical para horizontal, o modelo classificou corretamente toda a sequência de movimentos como *no_fall*. A análise temporal diferenciou a velocidade e a maneira de deitar-se de uma aceleração abrupta e descontrolada característica de uma queda real.

Figura 28 – Teste de vídeo (2) do modelo final




Fonte: De autoria própria.

# 5 CONCLUSÃO

O objetivo principal deste trabalho foi desenvolver uma solução robusta e automatizada para a detecção de quedas em ambientes domésticos, um desafio de crescente relevância social diante do aumento expressivo no número de pessoas que vivem sozinhas. A hipótese central era utilizar uma abordagem híbrida, combinando a extração de características espaciais com a análise do contexto temporal, para superar as limitações dos sistemas baseados em sensores vestíveis ou sensores ambientais, oferecendo um método de monitoramento não invasivo, ininterrupto, preciso e confiável.

Para validar essa hipótese, foi implementado um pipeline em duas etapas principais. A primeira etapa consistiu no treinamento de um modelo de visão computacional para detecção e estimativa de pose, o yolov11n-pose, sobre uma base de dados de vídeos curtos, rotulados com três classes de ação (*no_fall, fall e attention*), caixas delimitadoras (*bounding boxes*) e 18 pontos-chave de articulações do corpo humano (*keypoints*). Através de técnicas de balanceamento de classes com o aumento de dados sintéticos (*data augmentation*), o modelo YOLO foi utilizado para extrair as características das detecções, convertendo cada *frame* dos vídeos em um vetor de 128 dimensões, englobando tanto o posicionamento estático quanto a dinâmica do movimento (velocidade). A segunda etapa envolveu a utilização desses vetores, em formato de sequências (janelas) de 30 *frames*, para treinar um modelo de classificação temporal, uma Rede Neural Recorrente do tipo LSTM Empilhada, capaz de analisar e classificar a ação contida nessa janela de tempo.

A análise dos resultados confirmou o sucesso da abordagem proposta. O modelo de classificação temporal obteve uma acurácia geral de 96.6% no conjunto de validação, demonstrando uma excelente capacidade de generalização (para cenários similares à base de dados), evidenciada principalmente pela proximidade entre as curvas de acurácia e perda de treinamento e validação. A performance na detecção de uma queda, objeto principal deste estudo, foi excepcional, com uma AUC de 0.9993, indicando um alto poder de distinção entre outras atividades. A análise da matriz de confusão confirma essa eficácia, com taxas de falsos negativos e positivos extremamente baixas. Conclui-se, portanto, que a metodologia empregada não apenas validou a hipótese inicial, mas também se mostrou uma solução de alta performance para o problema.

Apesar do desempenho promissor, algumas limitações devem ser consideradas. A performance do modelo foi validada com uma base de dados pública, com um número restrito de atores e ambientes controlados. O modelo não conseguiu generalizar muito bem cenários do mundo real, possivelmente influenciado pelo uso de dados sintéticos baseados nos originais com planos de fundo semelhantes e fatores como oclusões, ângulo da câmera, proximidade do indivíduo, cenário complexo, variações de iluminação, presença de

múltiplos indivíduos, entre outros. Esta observação baseou-se em experimentos realizados tanto com vídeos gravados por mim, em condições domésticas simuladas, quanto com outras base de dados disponíveis publicamente na internet. Nesses testes adicionais, o desempenho do modelo não foi satisfatório, de acordo com os pontos já mencionados e reforçando a necessidade de estudos aprofundados para maior robustez.

Como trabalhos futuros, sugere-se validar o modelo com uma base de dados mais ampla e diversificada, cobrindo as limitações apresentadas. Integrar uma funcionalidade para o envio de mensagens de emergência automáticas aos familiares ou serviços médicos, para a criação de um produto realmente funcional. Adicionalmente, a arquitetura pode ser melhorada para detectar outras nuances como desmaios em diferentes posições ou ainda um indivíduo parado na mesma posição por muito tempo, assim como comportamentos anormais e outros tipos de acidentes. Explorar outras configurações e arquiteturas temporais como GRU, LSTM Bidirecional ou Transformers, podendo trazer novos ganhos de performance e eficiência. Por fim, otimizar o pipeline para implantação em dispositivos de borda (*edge devices*) com menor poder computacional, um caminho importante para tornar a solução mais acessível e escalável.

Em resumo, este trabalho demonstrou que a junção de redes neurais convolucionais para detecção de pessoas e redes neurais recorrentes para a análise temporal constitui uma abordagem eficaz para a detecção de quedas, reforçando o potencial da inteligência artificial para criar soluções que podem salvar vidas e aumentar a segurança e a autonomia de pessoas que vivem sozinhas, sem a necessidade de utilizar dispositivos ou se expor a sistemas invasivos ou não confiáveis.

# REFERÊNCIAS

ADHIKARI, K. **Computer Vision Based Posture Estimation and Fall Detection**. July 2019. Tese (PhD Thesis) — Bournemouth University, Faculty of Media and Communication, July 2019. Submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy. Disponível em: https://eprints.bournemouth.ac.uk/33227/.

ARIUNBOLD, Y.; BRITO, S.; LEONG, A. Falldetectnet: A computer vision platform for fall detection. *In*: . [*S.l.: s.n.*], 2020. Disponível em: https://api.semanticscholar.org/CorpusID:236474105.

BICHOFE, A. R. *et al.* Impacto do tempo de resposta no serviÇo de trauma agudo pré-hospitalar. **Revista ft**, v. 29, n. 140, nov. 2024. Disponível em: https://revistaft.com.br/impacto-do-tempo-de-resposta-no-servico-de-trauma-agudo-pre-hospitalar/.

CHO, K. *et al.* **Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation**. 2014. Disponível em: https://arxiv.org/abs/1406.1078.

CVAT. **Computer Vision Annotation Tool (CVAT)**. 2017. Accessed: 2025-04-11. Disponível em: https://github.com/cvat-ai/cvat.

DABASS, J. **Introducing GRU, RNN, and LSTM: A Beginner's Guide to Understanding These Revolutionary Deep Learning Models**. 2024. Accessed: 2025-04-11. Disponível em: https://python.plainenglish.io/introducing-gru-rnn-and-lstm-a-beginners-guide-to-understanding-these-revolutionary-deep-35b509a3

DataCamp. **YOLO Object Detection Explained**. 2023. Accessed: 2025-04-11. Disponível em: https://www.datacamp.com/blog/yolo-object-detection-explained.

DATASUS. **DATASUS - Departamento de Informática do SUS**. 2022. Disponível em: https://datasus.saude.gov.br/informacoes-de-saude-tabnet/. Acesso em: 08 abr. 2025.

DW MOON HJ, H. N. K. L. Association between ambulance response time and neurologic outcome in patients with cardiac arrest. **Am J Emerg Med.**, fev. 2019. Disponível em: https://pubmed.ncbi.nlm.nih.gov/30795948/.

EKRAM, A. *et al.* A dataset for human fall detection in videos. **GMDCSA**, Elsevier, 2024.

Empresa Brasileira de Serviços Hospitalares (EBSERH). **Atendimento rápido é fundamental para salvar a paciente com AVC agudo**. 2025. Acesso em: 08 abr. 2025. Disponível em: https://www.gov.br/ebserh/pt-br/hospitais-universitarios/regiao-sudeste/hc-ufmg/comunicacao/noticias/atendimento-rapido-e-fundamental-para-salvar-a-paciente-com-avc-agudo.

FATIMA, Z. *et al.* Performance comparison of object detection models for road sign detection under different conditions. **International Journal of Advanced Computer Science and Applications**, v. 15, 01 2024.

GEEKSFORGEEKS. **RNN vs LSTM vs GRU vs Transformersy**. 2025. Accessed: 2025-09-16. Disponível em: https://www.geeksforgeeks.org/deep-learning/rnn-vs-lstm-vs-gru-vs-transformers/.

HUI, J. **Object Detection Speed and Accuracy Comparison: Faster R-CNN, R-FCN, SSD and YOLO**. 2018. Accessed: 2025-04-11. Disponível em: https://jonathan-hui.medium.com/object-detection-speed-and-accuracy-comparison-faster-r-cnn-r-fcn-ssd-and-yolo-5425656ae359.

IDREES, H. **RNN vs. LSTM vs. GRU: A Comprehensive Guide to Sequential Data Modeling**. 2025. Accessed: 2025-09-16. Disponível em: https://medium.com/@hassaanidrees7/rnn-vs-lstm-vs-gru-a-comprehensive-guide-to-sequential-data-modeling-03aab16647bb.

IKECHUKWU, M. C.; WANG, J. **Tackling Visual Illumination Variations in Fall Detection for Healthcare Applications**. 2024. 1-6 p.

Instituto Brasileiro de Geografia e Estatística (IBGE). **Censo Demográfico 2022: População e Domicílios do Brasil**. 2022. Acesso em: 08 abr. 2025. Disponível em: https://censo2022.ibge.gov.br/.

JIANG, Y. *et al.* Fall detection on embedded platform using infrared array sensor for healthcare applications. **Neural Computing and Applications**, v. 36, p. 1–16, 12 2023.

JOCHER, G.; QIU, J.; CHAURASIA, A. **Ultralytics YOLO**. Ultralytics, 2023. Disponível em: https://github.com/ultralytics/ultralytics.

JOSHI, S. **Top Object Detection Models in 2025**. 2025. Accessed: 2025-09-13. Disponível em: https://www.hitechbpo.com/blog/top-object-detection-models.php.

KEYLABS. **YOLOv8 vs SSD: Choosing the Right Object Detection Model**. 2023. Accessed: 2025-09-13. Disponível em: https://keylabs.ai/blog/yolov8-vs-ssd-choosing-the-right-object-detection-model/.

KHALILI, S.; MOHAMMADZADE, H.; AHMADI, M. M. Elderly fall detection using cctv cameras under partial occlusion of the subjects body. 08 2022.

KUMAR, P.; CHAUHAN, S.; AWASTHI, L. K. Human activity recognition (har) using deep learning: Review, methodologies, progress and future research directions. **Archives of Computational Methods in Engineering**, v. 31, p. 179–219, 2024. Disponível em: https://doi.org/10.1007/s11831-023-09986-x.

LS MORAES MA, S. R. R. L. C. B. J. P. S. E. B. C. T. C. M. F. M. Factors associated with decision time to seek care in the face of ischemic stroke. **Revista da Escola de Enfermagem da USP**, Universidade de São Paulo, v. 57, p. e20230075, 2023. PMID: 37624382. Acesso em: 08 abr. 2025. Disponível em: https://pubmed.ncbi.nlm.nih.gov/37624382/.

MARQUES, M.; CAMPOS, G.; NASCIMENTO, A. Protótipo de aparelho para detecção de quedas em idosos. **Revista Kairós-Gerontologia**, v. 25, p. 227–249, 11 2022.

NASSER, A. A. *et al.* Every minute counts: The impact of pre-hospital response time and scene time on mortality of penetrating trauma patients. **The American Journal of Surgery**, v. 220, n. 1, p. 240–244, 2020. ISSN 0002-9610. Disponível em: https://www.sciencedirect.com/science/article/pii/S0002961019312693.

NIZAM, Y.; JAMIL, M. M. A. Classification of daily life activities for human fall detection: A systematic review of the techniques and approaches. p. 137–179, 01 2020.

PAREKH, R.; ADAMS, K. B.; SCHUMAN, D. Advance directives: Solo agers risk of becoming "unbefriended". **Innovation in Aging**, Oxford University Press on behalf of The Gerontological Society of America, v. 6, n. Suppl 1, p. 527, 2022. ECollection 2022 Nov. PMID: PMC9771146. Acesso em: 08 abr. 2025. Disponível em: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9771146/.

RAO, N. **YOLOv11 Explained: Next-Level Object Detection with Enhanced Speed and Accuracy**. 2024. Accessed: 2025-04-11. Disponível em: https://medium.com/@nikhil-rao-20/yolov11-explained-next-level-object-detection-with-enhanced-speed-and-accuracy-2dbe2d376f71.

RAVURI, A. A systematic literature review on human activity recognition. **Journal of Electrical Systems**, v. 20, p. 1175–1191, 04 2024.

REDDY, N. **Exploring YOLOv11: Faster, Smarter, and More Efficient**. 2024. Accessed: 2025-04-11. Disponível em: https://medium.com/@nandinilreddy/exploring-yolo11-faster-smarter-and-more-efficient-f4243d910d1e.

REDMON, J. *et al.* You only look once: Unified, real-time object detection. 2016. Disponível em: https://arxiv.org/abs/1506.02640.

RYAN, T. J. J. **LSTMs Explained: A Complete, Technically Accurate, Conceptual Guide with Keras**. 2020. Accessed: 2025-04-11. Disponível em: https://medium.com/analytics-vidhya/lstms-explained-a-complete-technically-accurate-conceptual-guide-with-keras-2a650327e8f2.

SARKAR, T. **Why You Should Start Using .npy Files More Often**. 2018. Accessed: 2025-09-14. Disponível em: https://www.kdnuggets.com/2018/04/start-using-npy-files-more-often.html.

SHAHRIAR, N. **What is Convolutional Neural Network (CNN) - Deep Learning**. 2021. Accessed: 2025-04-11. Disponível em: https://nafizshahriar.medium.com/what-is-convolutional-neural-network-cnn-deep-learning-b3921bdd82d5.

SHERSTINSKY, A. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. **Physica D: Nonlinear Phenomena**, Elsevier BV, v. 404, p. 132306, mar. 2020. ISSN 0167-2789. Disponível em: http://dx.doi.org/10.1016/j.physd.2019.132306.

V7LABS. **Human Activity Recognition: The Definitive Guide**. 2022. Acessado em: 11 abr. 2025. Disponível em: https://www.v7labs.com/blog/human-activity-recognition.

VENNERøD, C. B.; KJæRRAN, A.; BUGGE, E. S. **Long Short-term Memory RNN**. 2021. Disponível em: https://arxiv.org/abs/2105.06756.

WANG, Y.; DENG, T. Enhancing elderly care: Efficient and reliable real-time fall detection algorithm. **Digital Health**, v. 10, Jan 2024.

WRIA, A.; MOHAMMED, A. A comparative study using improved lstm /gru for human action recognition. 12 2022.

ZHANG, X. *et al.* Fall-mamba: A multimodal fusion and masked mamba-based approach for fall detection. **IEEE Internet of Things Journal**, v. 12, n. 8, p. 10493–10505, 2025.

ZHENG, X. *et al.* A high-precision human fall detection model based on fasternet and deformable convolution. **Electronics**, v. 13, p. 2798, 07 2024.