Universidade Estadual de Campinas
Instituto de Computação

Guilherme Vieira Leite

Detecção de Quedas de Pessoas em Vídeos Utilizando Redes Neurais Convolucionais com Múltiplos Canais

CAMPINAS
2020

Guilherme Vieira Leite

Detecção de Quedas de Pessoas em Vídeos Utilizando Redes Neurais Convolucionais com Múltiplos Canais

Dissertação apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Hélio Pedrini

Este exemplar corresponde à versão final da Dissertação defendida por Guilherme Vieira Leite e orientada pelo Prof. Dr. Hélio Pedrini.

CAMPINAS
2020

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

L536d

Leite, Guilherme Vieira, 1991-
Detecção de quedas de pessoas em vídeos utilizando redes neurais convolucionais com múltiplos canais / Guilherme Vieira Leite. – Campinas, SP : [s.n.], 2020.

Orientador: Hélio Pedrini.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Aprendizado de máquina. 2. Reconhecimento de padrões. 3. Redes neurais convolucionais. I. Pedrini, Hélio, 1963-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Human fall detection on videos using convolutional neural networks with multiple channels
Palavras-chave em inglês:
Machine learning
Pattern recognition
Convolutional neural networks
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
Hélio Pedrini [Orientador]
Gabriel Martins Dias
Esther Luna Colombini
Data de defesa: 14-02-2020
Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0002-6871-2069
- Currículo Lattes do autor: http://lattes.cnpq.br/2927269217627994

Universidade Estadual de Campinas
Instituto de Computação

Guilherme Vieira Leite

Detecção de Quedas de Pessoas em Vídeos Utilizando Redes Neurais Convolucionais com Múltiplos Canais

Banca Examinadora:

- Prof. Dr. Hélio Pedrini
  IC/UNICAMP
- Prof. Dr. Gabriel Martins Dias
  Semantix Brasil
- Profa. Dra. Esther Luna Colombini
  IC/UNICAMP

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 14 de fevereiro de 2020

Agradecimentos

- Aos meus familiares.
- À minha futura esposa, Jordana.
- Ao meu orientador, Prof. Hélio.
- À Semantix Brasil pela bolsa de mestrado.

Resumo

Baixas taxas de mortalidade infantil, avanços na medicina e mudanças culturais aumentaram a expectativa de vida nos países desenvolvidos para mais de 60 anos. Alguns países esperam que, até 2030, 20% da sua população tenham mais de 65 anos. A qualidade de vida nessa idade avançada é altamente determinada pela saúde do indivíduo, que ditará se o idoso pode se engajar em atividades importantes para o seu bem estar, independência e satisfação pessoal. O envelhecimento é acompanhado por problemas de saúde causados por limitações biológicas e fraqueza muscular. Esse enfraquecimento facilita a ocorrência de quedas, responsáveis pela morte de aproximadamente 646.000 pessoas em todo o mundo e, mesmo quando uma pequena queda ocorre, ela ainda pode fraturar ossos ou danificar tecidos moles, que não cicatrizam completamente. Lesões e danos dessa natureza, por sua vez, podem afetar a autoconfiança do indivíduo, diminuindo sua independência. Neste trabalho, propomos um método capaz de detectar quedas humanas em sequências de vídeo usando redes neurais convolucionais (CNNs) multicanais. Nós desenvolvemos dois métodos para detecção de quedas, o primeiro utilizando uma CNN 2D e o segundo utilizando uma CNN 3D. Nossos métodos utilizam características extraídas previamente de cada quadro do vídeo e as classificam. Após a etapa de classificação, uma máquina de vetores de suporte (SVM) é aplicada para ponderar os canais de entrada e indicar se houve ou não uma queda. Experimentamos quatro tipos de características, a saber: (i) fluxo óptico, (ii) ritmo visual, (iii) estimativa de pose e (iv) mapa de saliência. As bases de dados utilizadas (URFD e FDD) estão disponíveis publicamente e nossos resultados são comparados com os da literatura. As métricas selecionadas para avaliação são acurácia balanceada, acurácia, sensibilidade e especificidade. Nossos métodos apresentaram resultados competitivos com os obtidos pelo estado da arte na base de dados URFD e superam os obtidos na base de dados FDD. Ao conhecimento dos autores, nós somos os primeiros a realizar testes cruzados entre os conjuntos de dados em questão, e a reportar resultados de acurácia balanceada. Os métodos propostos são capazes de detectar quedas nas bases selecionadas. A detecção de quedas, bem como a classificação de atividades em vídeos, está fortemente relacionada à capacidade da rede de interpretar informações temporais e, como esperado, o fluxo óptico é a característica mais relevante para a detecção de quedas.

Abstract

Lower child mortality rates, advances in medicine, and cultural changes have increased life expectancy in developed countries over 60 years old. Some countries expect that, by 2030, 20% of their population will be over 65 years old. The quality of life at this advanced age is highly dictated by the individual's health, which will determine whether the elderly can engage in important activities to their well-being, independence, and personal satisfaction. Old age is accompanied by health problems caused by biological limitations and muscle weakness. This weakening facilitates the occurrence of falls, which are responsible for the deaths of approximately 646,000 people worldwide and, even when a minor fall occurs, it can still cause fractures, break bones or damage soft tissues, which will not heal completely. Injuries and damages of this nature, in turn, will consume the self-confidence of the individual, diminishing their independence. In this work, we propose a method capable of detecting human falls in video sequences using multichannel convolutional neural networks (CNN). We developed two methods for fall detection, the first using a 2D CNN and the second using a 3D CNN. Our method uses features previously extracted from each frame and classifies them with a CNN. After the classification step, a support vector machine (SVM) is applied to weight the input channels and indicate whether or not there was a fall. We experiment with four types of features, namely: (i) optical flow, (ii) visual rhythm, (iii) pose estimation, and (iv) saliency map. The benchmarks used (URFD and FDD) are publicly available and our results are compared to those in the literature. The metrics selected for evaluation are balanced accuracy, accuracy, sensitivity, and specificity. Our results are competitive with those obtained by the state of the art on the URFD data set and surpass those on the FDD data set. To the authors' knowledge, we are the first to perform cross-tests between the datasets in question and to report results for the balanced accuracy metric. The proposed method is able to detect falls in the selected benchmarks. Fall detection, as well as activity classification in videos, is strongly related to the network's ability to interpret temporal information and, as expected, optical flow is the most relevant feature for detecting falls.

Lista de Figuras

1.1 Visão geral do sistema de emergência 13

2.1 Arquitetura VGG-16 17
2.2 Arquitetura Inception 17
2.3 Módulo Inception 18
2.4 Exemplos de fluxo óptico 19
2.5 Construção do ritmo visual 20
2.6 Exemplos de ritmo visual 20
2.7 Exemplo de mapas de saliência 21
2.8 Exemplo de estimação de pose 22
2.9 Hardware de Kukharenko e Romanenko [30] 22
2.10 Hardware de Kumar et al. [31] 23
2.11 Hardware de Vallejo et al. [72] 23
2.12 Hardware de Zhao et al. [83] 24
2.13 Experimento de Zigel et al. [85] 24

3.1 Diagrama geral da metodologia 29
3.2 Janela deslizante 30
3.3 Transferência de aprendizado 31

4.1 Base de dados URFD 34
4.2 Base de dados FDD 35

Lista de Tabelas

4.1 VGG-16 multicanais URFD 37
4.2 VGG-16 multicanais FDD 38
4.3 VGG-16 cruzado URFD FDD 39
4.4 VGG-16 cruzado FDD URFD 39
4.5 Inception 3D multicanais URFD 40
4.6 Inception 3D multicanais FDD 41
4.7 Inception 3D cruzado URFD FDD 42
4.8 Inception 3D cruzado FDD URFD 42
4.9 Método proposto vs literatura - URFD 43
4.10 Método proposto vs literatura - FDD 43

Lista de Abreviações e Siglas

2D	Two-Dimensional
3D	Three-Dimensional
ADL	Assisted Daily Living
AWS	Amazon Web Services
CNN	Convolutional Neural Network
DNN	Deep Neural Network
DTW	Dynamic Time Warping
FDD	Fall Detection Dataset
FPS	Frames per Second
GPU	Graphics Processing Unit
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
KNN	K Nearest Neighbor
LSTM	Long Short-Term Memory
MEWMA	Multivariate Exponentially-Weighted Moving Average
OF	Optical Flow
PE	Pose Estimation
R-CNN	Regions with Convolutional Neural Network
RAM	Random-Access Memory
RGB	Red-Green-Blue
RGB+D	Red-Green-Blue+Depth
SA	Saliency
SVM	Support Vector Machine
TST	Telecommunication Systems Team
TPU	Tensor Processing Unit
URFD	University of Rzeszow Fall Detection Dataset
VGG	Visual Geometry Group
VR	Visual Rhythm

Sumário

1 Introdução **12**
1.1 Motivação . . . 12
1.2 Objetivos . . . 14
1.3 Questões de Pesquisa . . . 14
1.4 Contribuições . . . 14
1.5 Lista de Publicações . . . 15
1.6 Organização do Texto . . . 15

2 Conceitos e Trabalhos Relacionados **16**
2.1 Redes Neurais Profundas . . . 16
2.2 Redes Neurais Convolucionais . . . 16
2.2.1 Transferência de Aprendizado . . . 18
2.3 Definição de Queda . . . 18
2.4 Fluxo Óptico . . . 19
2.5 Ritmo Visual . . . 19
2.6 Mapa de Saliência . . . 20
2.7 Estimação de Pose . . . 21
2.8 Literatura Relacionada . . . 21
2.8.1 Métodos Sem Vídeos . . . 22
2.8.2 Métodos Com Vídeos . . . 24

3 Metodologia **28**
3.1 Pré-Processamento . . . 28
3.1.1 Extração de Características . . . 28
3.1.2 Aumentação de Dados . . . 30
3.2 Treinamento . . . 31
3.3 Teste . . . 32

4 Experimentos e Resultados **33**
4.1 Bases de Dados . . . 33
4.1.1 Métricas de Avaliação . . . 35
4.1.2 Recursos Computacionais . . . 36
4.2 Experimentos e Resultados . . . 36

5 Conclusões e Trabalhos Futuros **44**

Referências Bibliográficas **46**

Capítulo 1

Introdução

A expectativa de vida, em países desenvolvidos, alcançou valores acima de 60 anos [75]. Alguns países da União Europeia e China esperam que 20% de sua população atinjam mais de 65 anos até 2030 [24]. Segundo a Organização Mundial da Saúde [24, 77], esse recente patamar é um efeito colateral de avanços científicos, descobertas médicas, redução da mortalidade infantil e mudanças culturais.

Entretanto, embora os seres humanos estejam vivendo mais, a qualidade deste tempo de vida é definida principalmente pela saúde. Como a saúde dita a independência, a satisfação e a possibilidade de engajar em atividades importantes para o bem-estar do idoso, diversos grupos de pesquisa direcionaram seu foco para tecnologias de manutenção da saúde na terceira idade.

1.1 Motivação

Naturalmente, ao longo do envelhecimento aparecem problemas de saúde, principalmente em decorrência das limitações biológicas do corpo humano e do enfraquecimento muscular. Esse enfraquecimento em idades avançadas aumenta as chances de um individuo sofrer uma queda. Em termos de acidentes domésticos, as quedas são a segunda maior causa de morte ao redor do mundo, com números em torno de 646.000 mortes por ano [24]. Relatórios apontam que entre 28% e 35% da população com mais de 65 anos sofrem pelo menos uma queda ao ano e este percentual sobe para 32% a 42% em idosos com mais de 70 anos.

Em uma tentativa de agrupar os eventos que podem levar a uma queda, Lusardi et al. [45] relataram os fatores de risco, a forma como as quedas ocorrem, quem as sofre e algumas precauções para evitá-las. Mesmo em casos em que ocorre uma pequena queda, ela pode quebrar ou trincar ossos e machucar tecidos moles que, devido à idade avançada, podem não se recuperar completamente. Isso causa uma cadeia de danos fisiológicos e psicológicos, consequentemente diminuindo a auto-confiança e independência do idoso.

Assim como crianças, os idosos necessitam de cuidados e acompanhamento constantes para evitar ferimentos graves, especialmente no que diz respeito ao tempo entre a ocorrência do acidente e o início dos cuidados adequados. Uma das medidas efetivas é a contratação de cuidadores qualificados para acompanharem o idoso em tempo integral.

Porém, em particular nos países desenvolvidos, onde serviços são caros, este custo com profissionais somado a outros orçamentos elevado da vida idosa, como médicos e medicamentos, acaba sendo inviabilizado. Em geral, esses idosos são realocados para a casa de seus familiares próximos, o que diminui sua privacidade e independência.

Com base no cenário descrito anteriormente, pode-se arquitetar uma solução tecnológica na forma de um sistema de emergência que, de forma confiável, acionaria a assistência qualificada automaticamente. Dessa maneira, o tempo de resposta entre o acidente e os cuidados seria reduzido, possibilitando ao idoso habitar em sua própria residência, mantendo sua independência e auto-estima. O diagrama da Figura 1.1 ilustra os componentes de um sistema de assistência ao dia-a-dia (do inglês ADL ou *Assisted Daily Living*).

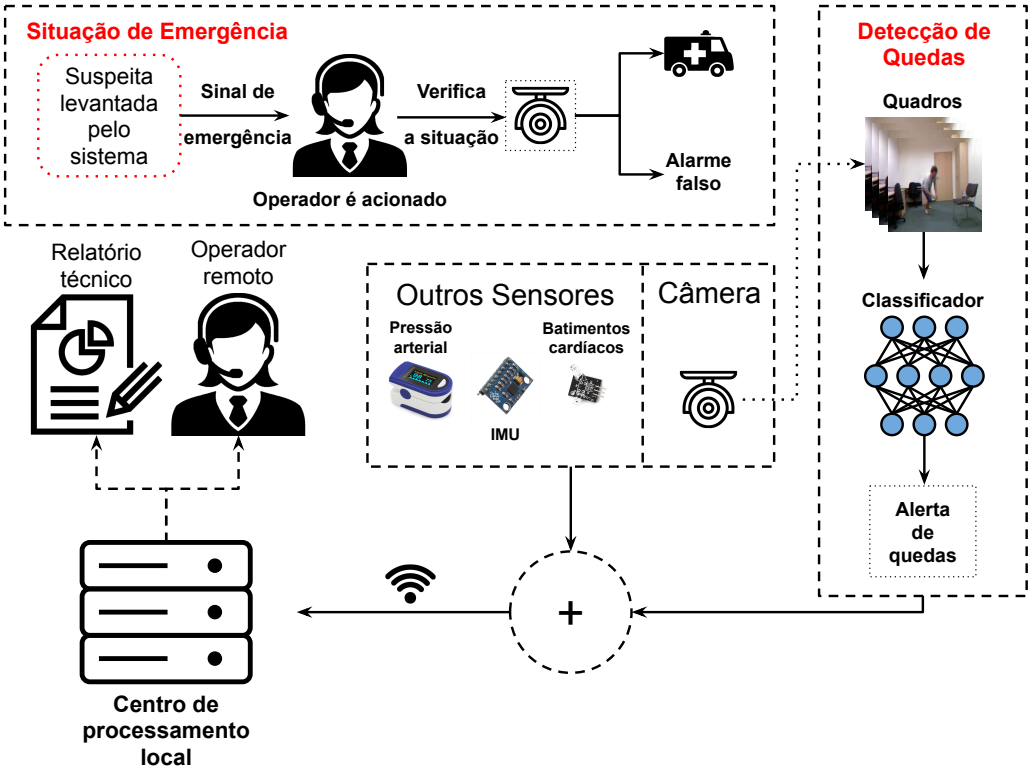


Figura 1.1: Diagrama com os principais componentes do sistema de computação de emergência para monitoramento de acidentes.

O sistema é alimentado por uma gama de sensores, instalados na residência ou aderidos ao corpo do idoso, que monitoram informações como a pressão sanguínea, número de batimentos cardíacos, temperatura, níveis de glicose no sangue, aceleração e postura. Esses dispositivos são conectados a um centro de processamento local que, além de construir continuamente relatórios técnicos da saúde do idoso, também aciona um operador remoto em caso de emergência. Ao receber o alerta de emergência, o operador realiza uma verificação da situação e, ao constatar a necessidade, aciona o socorro médico.

1.2 Objetivos

A partir das motivações acima, este trabalho foca na utilização de uma abordagem multicanal, aliada a técnicas de aprendizado profundo, para a detecção de quedas humanas em sequências de vídeos.

Um algoritmo capaz de realizar essa detecção de forma automática é um módulo importante no sistema de emergência da Figura 1.1 e, para tal, os seguintes objetivos foram definidos:

1. Definição de um patamar base na literatura relacionada.
2. Preparação das bases de dados.
3. Extensão da arquitetura de patamar base.
4. Avaliação dos multicanais de informações.
5. Avaliação do modelo proposto.
6. Publicação dos resultados.

1.3 Questões de Pesquisa

Durante o cumprimento dos objetivos definidos anteriormente, pretendemos responder às seguintes questões de pesquisa:

1. Os multicanais são capazes de manter informação temporal o suficiente para que uma rede aprenda seus padrões?
2. Quais canais contribuem melhor ao problema de detectar quedas?
3. A eficácia do método proposto se mantém para outras bases de dados contendo quedas humanas?
4. As arquiteturas tridimensionais (3D) são mais discriminativas do que as arquiteturas bidimensionais (2D) para este problema?

1.4 Contribuições

As principais contribuições deste trabalho são dois modelos multicanais para a detecção de quedas humanas, testados em duas bases de dados e disponíveis publicamente. Além disso, nós apresentamos uma extensa comparação entre os canais e as arquiteturas empregadas, cujos resultados são comparáveis ao estado da arte, bem como uma discussão sobre a utilização de conjuntos de dados simulados.

1.5 Lista de Publicações

Os seguintes artigos [7, 8, 36] foram publicados durante a realização deste trabalho de pesquisa, cujos resultados estão diretamente relacionados ao tema investigado nesta dissertação:

- G.V. Leite, G.P. Silva, H. Pedrini. *Fall Detection in Video Sequences Based on a Three-Stream Convolutional Neural Network.* 18th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 191–195, Boca Raton-FL, USA, December 16-19, 2019.
- S.A. Carneiro, G.P. Silva, G.V. Leite, R. Moreno, S.J.F. Guimarães, H. Pedrini. *Deep Convolutional Multi-Stream Network Detection System Applied to Fall Identification in Video Sequences.* 15th International Conference on Machine Learning and Data Mining (MLDM), pp. 681-695, New York-NY, USA, July 20-24, 2019.
- S.A. Carneiro, G.P. Silva, G.V. Leite, R. Moreno, S.J.F. Guimarães, H. Pedrini. *Multi-Stream Deep Convolutional Network Using High-Level Features Applied to Fall Detection in Video Sequences.* 26th International Conference on Systems, Signals and Image Processing (IWSSIP), pp. 293-298, Osijek, Croatia, June 05-07, 2019.

1.6 Organização do Texto

O restante deste trabalho está estruturado da seguinte forma. O Capítulo 2 apresenta os principais conceitos que foram utilizados na implementação do trabalho, bem como abordagens existentes na literatura relacionadas ao tema investigado. O Capítulo 3 descreve em detalhes a metodologia proposta e suas etapas. O Capítulo 4 descreve os conjuntos de dados utilizados nos experimentos, as métricas de avaliação selecionadas, os experimentos realizados, seus resultados e discussões. Finalmente, o Capítulo 5 apresenta algumas considerações finais e propostas para trabalhos futuros.

Capítulo 2

Conceitos e Trabalhos Relacionados

Neste capítulo, os conceitos que foram utilizados na metodologia proposta são descritos em detalhes. Além disso, os trabalhos revisados durante a execução da tese são reportados na segunda parte do capítulo.

2.1 Redes Neurais Profundas

As redes neurais profundas (DNNs) são uma classe de algoritmos de aprendizado de máquina, em que várias camadas de processamento são utilizadas para extrair e transformar características dos dados de entrada e o aprendizado dos neurônios da rede ocorre pela aplicação do algoritmo de retropropagação (*backpropagation*). As informações de entrada de cada camada são as mesmas da saída da camada anterior, exceto na primeira camada, onde entram os dados externos, e na última camada, de onde são extraídos os resultados do processamento [21]. Essa estrutura não é necessariamente fixa, algumas camadas podem ter duas outras camadas como entrada ou várias saídas.

Deng e Yu [14] citam alguns motivos pela crescente popularidade das redes profundas nos últimos anos, os quais incluem seus resultados nos problemas de classificação, melhorias nas Unidades de Processamento Gráfico (GPUs), aparecimento de Unidades de Processamento de Tensores (TPUs) e a quantidade de dados disponíveis digitalmente.

As camadas de uma rede profunda podem ser organizadas de diversas maneiras, a depender da necessidade de cada tarefa. A maneira na qual uma rede profunda é organizada é chamada de arquitetura e algumas delas se tornaram bem conhecidas devido aos resultados na classificação de imagens. Algumas delas são: AlexNet [29], LeNet [34], VGGNet [62] e ResNet [23].

2.2 Redes Neurais Convolucionais

As redes neurais convolucionais (CNNs) são um subtipo de redes profundas, em que sua estrutura é semelhante a de uma DNN, tal que a informação flui de uma camada para a próxima. Porém, no caso das CNNs, os dados passam pelas camadas convolucionais, que aplicam várias operações de convolução e redimensionam os dados, antes de serem repassados para a próxima camada.

As operações de convolução permitem que a rede aprenda características de baixo nível nas primeiras camadas, e combine-as nas camadas seguintes para aprender características de alto nível. Apesar de não ser obrigatório, geralmente no final de uma rede convolucional existem algumas camadas totalmente conexas.

No contexto deste trabalho, duas arquiteturas de CNNs são relevantes: (i) VGG-16 [62] e (ii) Inception [66]. A VGG-16 foi a vencedora da competição Desafio de Reconhecimento Visual em Larga Escala - ImageNet 2014 (ILSVRC), com um erro de 7,3% na categoria de localização. Sua característica de utilizar filtros pequenos, convoluções de 3×3, *stride* 1, *padding* 1 e *max pooling* de 2×2 com *stride* 2, permitiu que a rede fosse mais profunda, sem torná-la computacionalmente proibitiva. A VGG-16 possui 16 camadas e 138 milhões de parâmetros, o que é considerado pouco para redes profundas. A maior carga das computações desta rede ocorre nas primeiras camadas, pois, a partir delas, as camadas de *pooling* reduzem consideravelmente a carga das camadas mais profundas. A Figura 2.1 ilustra a arquitetura da VGG-16.

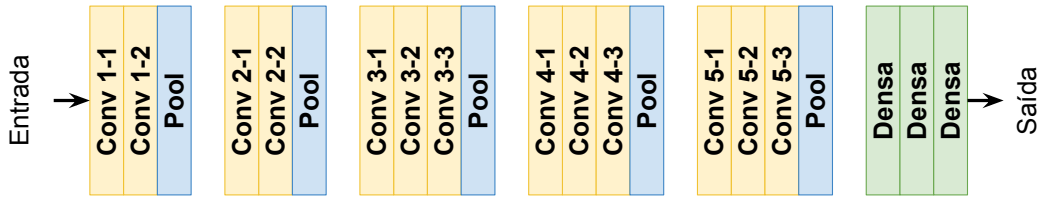


Figura 2.1: Ilustração das camadas da arquitetura VGG-16.

A segunda arquitetura, Inception V1 [66], foi a vencedora da ILSVRC 2014, mesmo ano da VGG-16, porém, venceu na categoria de classificação, com um erro de 6,7%. Esta rede foi desenvolvida para ser mais profunda e, ao mesmo tempo, computacionalmente mais eficiente. Como ilustrada na Figura 2.2, a rede possui 22 camadas e somente 5 milhões de parâmetros. Sua construção consiste no empilhamento de vários módulos, chamados de Inception, ilustrados na Figura 2.3a.

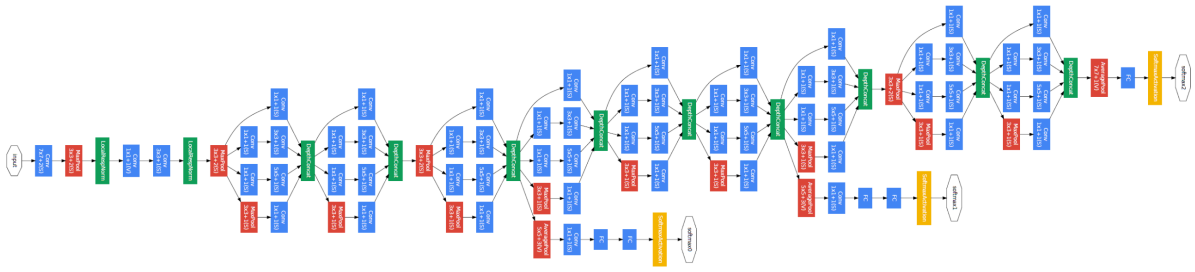
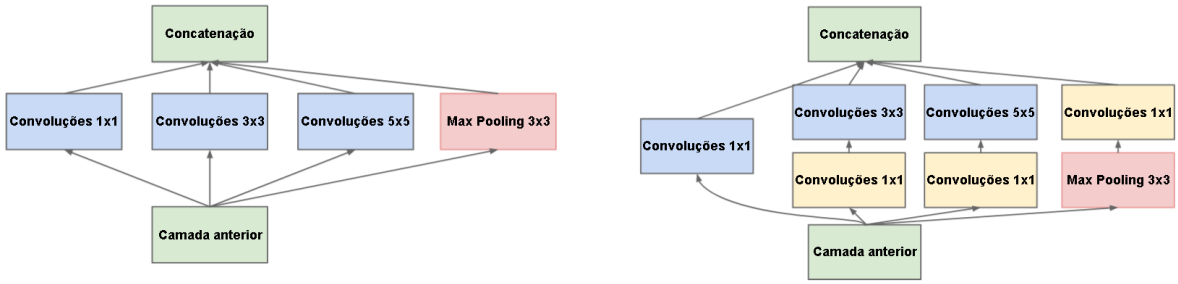


Figura 2.2: Ilustração das camadas da arquitetura Inception. Fonte: Szegedy et al. [66].

Os módulos foram pensados de forma a criar algo como uma rede dentro da rede, em que várias operações de convolução e *max pooling* são executadas paralelamente e, ao final destas operações, as características são concatenadas para ser enviadas ao próximo módulo. Entretanto, se a rede fosse composta por módulos Inception, como ilustrado na Figura 2.3a, ela executaria 850 milhões de operações. Para reduzir este número, os gargalos foram criados. Os gargalos reduzem o número de operações para 358 milhões. Eles são convoluções 1×1 que preservam a dimensão espacial, ao mesmo tempo que diminuem

a profundidade das características. Eles foram alocados antes das convoluções 3×3, 5×5 e após o *max pooling* 3×3, como ilustrado na Figura 2.3b.



(a) (b)

Figura 2.3: Ilustrações do módulo Inception. (a) módulo sem gargalo; (b) módulo com gargalo. Fonte: Szegedy et al. [66].

2.2.1 Transferência de Aprendizado

A transferência de aprendizado consiste no reaproveitamento de uma rede cujos pesos foram treinados em um outro contexto. Além de melhorar os resultados das redes que a utilizam, a técnica também colabora para diminuir o tempo necessário de convergência e suprir casos em que não há dados o suficiente para treinar a rede inteira [84].

Tipicamente em problemas de classificação, as redes são previamente treinadas na base de dados ImageNet [13], que é uma das maiores e mais conhecidas bases de dados. O objetivo é que a rede aprenda padrões complexos e genéricos o suficiente, de forma que sejam úteis a novos problemas.

2.3 Definição de Queda

A literatura não apresenta uma definição universal de queda [28, 44, 67], todavia, alguns órgãos de saúde criaram suas próprias definições, as quais podem ser utilizadas para descrever uma ideia geral de queda.

A instituição americana *Joint Commission* [68] define uma queda como "[...] uma mudança não intencional da posição, que termina ao chão ou a alguma superfície baixa (por exemplo, uma cama, cadeira ou um tapete). [...]". A Organização Mundial da Saúde [76] define como "[...] um evento cujo resultado é a pessoa estar inadvertidamente no chão ou algum nível mais baixo [...]". A definição do Centro Nacional de Assuntos de Veteranos [70] é a "Perda da postura ereta, resultando no descanso ao chão, objeto ou mobília, ou uma repentina, incontrolada, involuntária disposição do corpo na direção do chão, ou objeto próximo, como uma cadeira ou escada [...]". Neste trabalho nós descrevemos uma queda como um movimento involuntário que resulta em um indivíduo indo em direção ao chão.

2.4 Fluxo Óptico

O fluxo óptico é uma técnica que deduz representações de movimentos de pixels causados pelo deslocamento do objeto ou da câmera. O vetor que representa o movimento é o mesmo para uma vizinhança de pixels e é esperado que os pixels não saiam da área do quadro. O fluxo óptico é um método local, o que implica a dificuldade de seu cálculo em regiões uniformes.

O cálculo do fluxo óptico é realizado a partir da comparação de dois quadros consecutivos e sua representação é um vetor de direção e magnitude. Considere I um quadro de vídeo e $I(x, y, t)$ um pixel neste quadro. Um quadro analisado em um tempo futuro dt é descrito na Equação 2.1 em função de um deslocamento (dX, dY) do pixel $I(x, y, t)$. A Equação 2.2 é obtida de uma série de Taylor dividida por dt e apresenta os gradientes f_t, f_x e f_y (Equação 2.3). Os componentes do fluxo óptico são os valores de u e v (Equação 2.4) e podem ser obtidos por diversos métodos como o de Lucas-Kanade [43] ou Gunnar-Farnebäck [18]. A Figura 2.4 ilustra alguns quadros de fluxo óptico.

$$I(x, y, t) = I(x + dX, y + dY, t + dT) \tag{2.1}$$

$$f_x u + f_y v + f_t = 0 \tag{2.2}$$

$$f_x = \frac{\partial f}{\partial x} \qquad f_y = \frac{\partial f}{\partial y} \qquad f_t = \frac{\partial f}{\partial t} \tag{2.3}$$

$$u = \frac{\partial x}{\partial t} \qquad v = \frac{\partial y}{\partial t} \tag{2.4}$$

Figura 2.4: Exemplos de fluxo óptico. Cada fluxo óptico foi extraído entre o quadro ilustrado e o próximo na sequência. A cor do pixel indica a direção do movimento e o seu brilho indica a magnitude.

2.5 Ritmo Visual

Ritmo visual é uma técnica de codificação, com o objetivo de obter a informação temporal de um vídeo, sem perder a informação espacial. Sua representação consiste em uma única imagem correspondendo ao vídeo inteiro, de forma que cada quadro de vídeo contribui para uma coluna da imagem final [50, 69, 71].

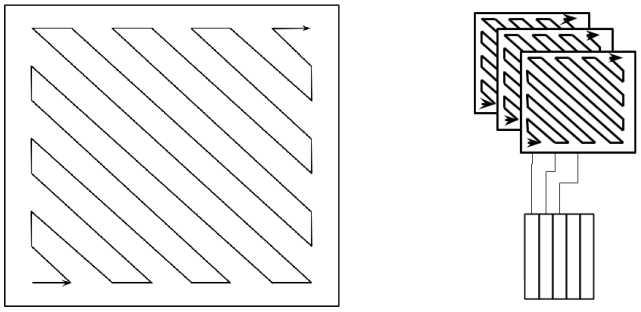


Figura 2.5: Processo de construção do ritmo visual. (a) forma em zigue-zague na qual cada quadro de vídeo é percorrido; (b) Construção do ritmo pela concatenação das colunas, obtidas pelo zigue-zague. Fonte: [71].

Para a construção do ritmo visual, cada quadro de vídeo é percorrido em zigue-zague, da sua diagonal esquerda inferior até sua diagonal direita superior, como ilustrado na Figura 2.5a. Cada quadro processado em zigue-zague gera uma coluna de pixels, a qual é concatenada com as outras colunas para formar o ritmo visual (Figura 2.5b). As dimensões da imagem de ritmo são de $W \times H$, em que a largura W é o número de quadros do vídeo e a altura H é o tamanho do caminho percorrido. A Figura 2.6 ilustra alguns ritmos visuais extraídos.

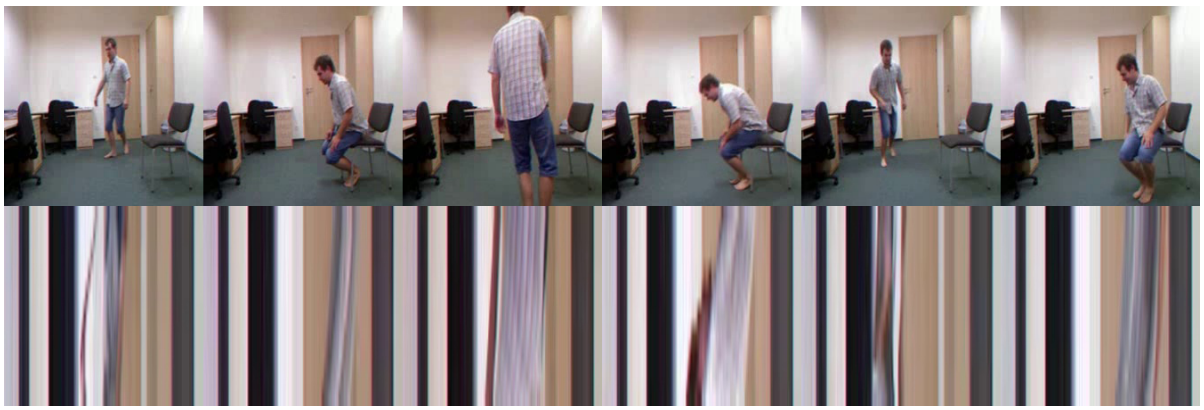


Figura 2.6: Exemplos de ritmo visual. Um quadro de cada vídeo é mostrado na parte superior e o correspondente ritmo visual do vídeo é apresentado na parte inferior da figura.

2.6 Mapa de Saliência

A saliência, no contexto de processamento de imagens, é uma característica da imagem que representa localizações de regiões na imagem. As regiões latentes são geralmente apresentadas em tons de cinza, regiões sem características ativadas são representadas por pixels pretos e a região mais latente pelo pixel branco. No contexto de aprendizado profundo, o mapa de saliência é o mapa da contribuição de cada pixel da imagem no processo de classificação. Inicialmente, a saliência foi extraída como uma forma de se compreender o que as redes profundas estavam de fato aprendendo e ainda é utilizada desta forma, como no trabalho de Li et al. [38].

O mapa de saliência foi utilizado por Zuo et al. [86] como característica para a classificação de ações egocêntricas, em que a fonte de informação é uma câmera que corresponde à visão em primeira pessoa do sujeito. A utilização da saliência na classificação de ações vem da suposição de que o importante de uma ação ocorre à frente dos quadros de vídeo, em vez de ocorrer ao fundo.

O mapa de saliência pode ser obtido de diversas maneiras, tal como mostrado por Smilkov et al. [63], Sundararajan et al. [65] e Simonyan et al. [60]. A Figura 2.7 ilustra a extração do mapa de saliência.

Figura 2.7: Exemplos de mapas de saliência. Os pixels variam do preto ao branco, de acordo com sua importância para a classificação, branco sendo a maior relevância.

2.7 Estimação de Pose

É uma técnica de derivação da postura de um ou mais seres humanos. Diferentes sensores são utilizados como entrada para essa técnica, como os sensores de profundidade de um Microsoft Kinect ou imagens de uma câmera.

O algoritmo proposto por Cao et al. [6], o OpenPose, é notável pela sua eficácia ao estimar a pose de indivíduos em quadros de vídeo. O OpenPose opera com uma rede de dois estágios em busca de 18 juntas do corpo. No primeiro estágio, o método cria mapas de confiança das posições das juntas e o segundo estágio prediz campos de afinidade entre as partes encontradas. A afinidade é representada por um vetor 2D, que codifica a posição e a orientação de cada membro do corpo. A Figura 2.8 exibe a extração da postura de alguns quadros.

2.8 Literatura Relacionada

Nas subseções a seguir, nós elucidamos os trabalhos relacionados à detecção de quedas e seus métodos. Os trabalhos foram divididos em dois grupos, baseados nos sensores utilizados: (i) métodos sem vídeos e (ii) métodos com vídeos.

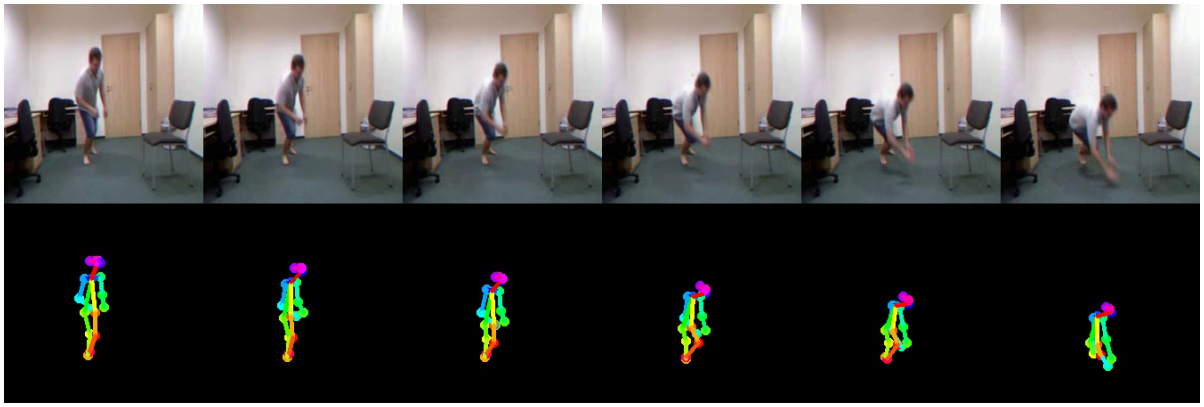


Figura 2.8: Exemplo da extração da estimação de postura. Os círculos representam as juntas encontradas e as arestas os membros. Cada membro é associado a uma cor fixa.

2.8.1 Métodos Sem Vídeos

Neste grupo, encontram-se os trabalhos que utilizam diversos tipos de sensores para obtenção de dados, podendo ser um relógio, acelerômetro, giroscópio, sensor de frequência cardíaca ou um *smartphone*, ou seja, qualquer sensor que não utilize uma câmera.

Khin et al. [27] desenvolveram um sistema de triangulação que utiliza vários sensores de presença instalados no cômodo monitorado de uma casa. A presença de alguém no cômodo causaria uma pertubação nos sensores e uma queda ativaria outro padrão de ativação. Apesar de não reportarem os resultados, os autores afirmaram que testaram essa configuração e que os sensores detectaram padrões diferentes para ações diferentes, indicando que a solução poderia ser utilizada para detectar quedas.

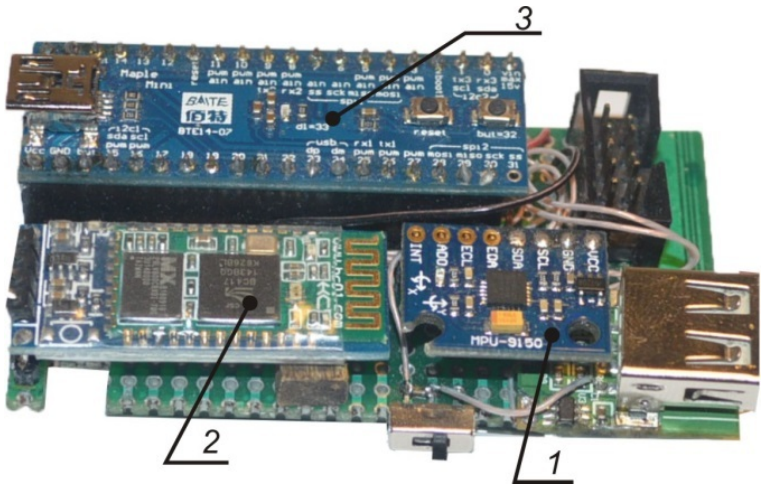


Figura 2.9: Ilustração do dispositivo proposto por Kukharenko e Romanenko [30].

Kukharenko e Romanenko [30] obtiveram seus dados de um dispositivo pulseira, ilustrado na Figura 2.9. Um limiar foi utilizado para detectar impacto ou um estado "sem peso" e, após esta detecção, o algoritmo espera por um segundo e analisa as informações obtidas. O método foi testado em alguns voluntários, que relataram reclamações tais como esquecer de vestir o dispositivo e o desconforto que o mesmo causava.

Kumar et al. [31] colocaram um sensor preso a um cinto na pessoa (Figura 2.10)

e compararam quatro métodos para detectar quedas: limiar, Máquinas de Vetores de Suporte (SVM), K vizinhos mais próximos (KNN) e *Dynamic Time Warping* (DTW). Os autores também comentaram a importância de sensores acoplados ao corpo, uma vez que eles monitorariam o indivíduo constantemente e não sofreriam com pontos cegos das câmeras.

Figura 2.10: Ilustração da simulação de queda e do dispositivo utilizado por Kumar et al. [31].

Vallejo et al. [72] desenvolveram uma rede neural profunda para classificar os dados do sensor. O sensor em questão é um giroscópio vestido na cintura, ilustrado na Figura 2.11, e a rede é composta de três camadas ocultas, com cinco neurônios cada uma. Os autores realizaram experimentos com adultos de 19 a 56 anos.

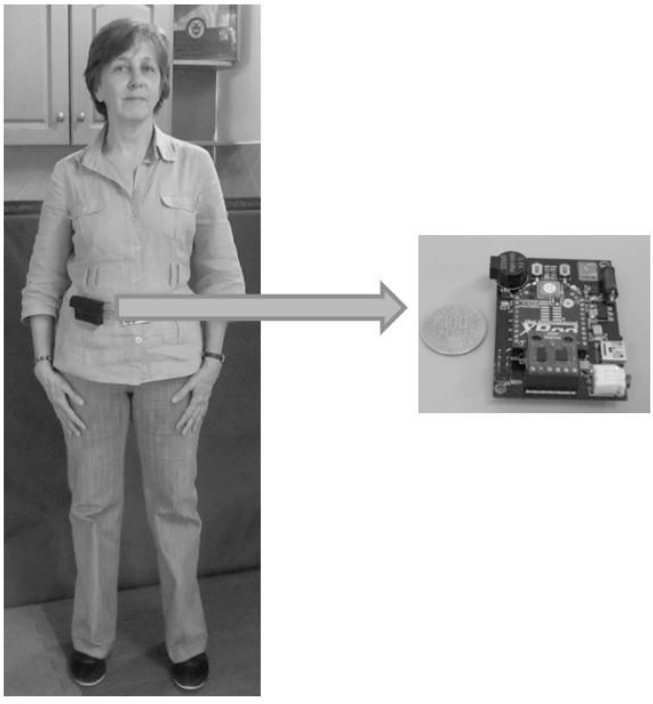


Figura 2.11: Ilustração do dispositivo utilizado por Vallejo et al. [72].

Zhao et al. [83] coletaram dados de um giroscópio acoplado à cintura do indivíduo, ilustrado na Figura 2.12, e utilizaram uma árvore de decisão para classificar as informações. Os experimentos foram executados em cinco adultos aleatórios.

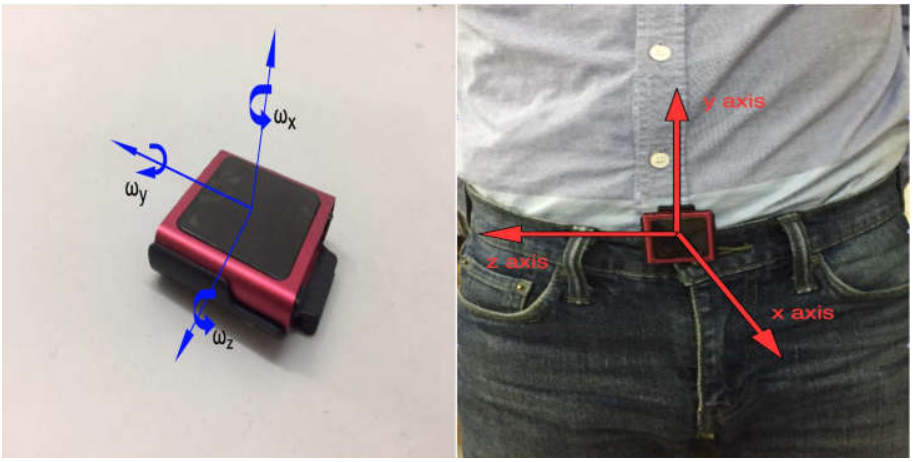


Figura 2.12: Ilustração do giroscópio, acoplado à cintura, utilizado por Zhao et al. [83].

Zigel et al. [85] utilizaram acelerômetros e microfones como sensores, porém, os sensores foram instalados no ambiente, em vez de acoplados ao sujeito. Os sensores detectaram vibrações e alimentaram um classificador quadrático. Os testes foram feitos com um boneco de testes, que era solto de uma posição ereta, exibido na Figura 2.13.

Figura 2.13: Ilustração da experimentação com um boneco de testes por Zigel et al. [85].

2.8.2 Métodos Com Vídeos

Nesta seção, os métodos foram agrupados cuja fonte principal de dados é formada por sequências de vídeos. Apesar da semelhança nos sensores, os métodos apresentam uma variedade de soluções como, por exemplo, os trabalhos a seguir que utilizam técnicas de ativação baseadas em limiar.

Para isolar a silhueta humana, Lee e Mihailidis [35] realizaram uma subtração do fundo em conjunto com uma extração de regiões. A partir disso, a postura foi determinada por meio de um limiar dos valores do perímetro da silhueta, velocidade do centro da silhueta e pelo diâmetro de Feret. A solução foi testada em uma base criada pelos autores.

Nizam et al. [52] usaram dois limiares, o primeiro verifica se a velocidade do corpo é alta e, neste caso, o segundo limiar verifica se a posição das juntas estão próximas ao chão. As informações das juntas foram obtidas após uma subtração do fundo, por meio de uma câmera Kinect. Os experimentos foram realizados em um conjunto de dados criado pelos autores.

Sase e Bhandari [56] utilizaram um limiar em que, se a região de interesse fosse menor do que um terço do corpo do indivíduo, uma queda era detectada. A região de interesse foi obtida pela extração do fundo e o método foi testado na base URFD [32]. Bhandari et al. [4] aplicaram um limiar sobre a velocidade e a direção da região de interesse. Uma combinação de Shi-Tomasi com Lucas-Kanade foi aplicada para determinar a região de interesse. Os autores testaram a abordagem no conjunto URFD [32] e reportaram 95% de acurácia.

Outra técnica de classificação bastante utilizada é o SVM, empregada por Abobakr et al. [1]. O método utilizou informações de profundidade para subtrair o fundo do quadro do vídeo, aplicou um algoritmo de floresta aleatória para estimar a postura e a classificou com SVM. Fan et al. [16] também separaram o quadro entre frente e fundo e encaixaram uma elipse na silhueta do corpo humano encontrado. A partir da elipse, seis características foram extraídas e serviram para alimentar uma função *slow feature*. As saídas dessa função passaram por um classificador SVM e foram testadas na base de dados SDUFall [46].

Harrou et al. [22] utilizaram um classificador SVM que recebe características extraídas dos quadros de vídeo. Durante os testes, os autores compararam o SVM com uma *multivariate exponentially-weighted moving average* (MEWMA) e testaram a solução nas bases URFD [32] e FDD [11].

Mohd et al. [49] alimentaram um classificador SVM com as informações de altura, velocidade, aceleração e posição das juntas do corpo humano e realizaram testes em três bases de dados: TST Fall Detection [19], URFD [32] e Fall Detection by Zhang [82]. Panahi e Ghods [55] subtraíram o fundo a partir das informações de profundidade, encaixaram uma elipse na forma do indivíduo, classificaram a elipse com SVM e realizaram testes no conjunto de dados URFD [32].

Preocupados com a privacidade dos usuários, os trabalhos a seguir defendem que as soluções para detectar quedas devem oferecer opções de anonimidade. Dessa forma, Edgcomb e Vahid [15] testaram a efetividade de um classificador árvore binária, sobre uma série temporal. Os autores compararam diversas formas de se esconder a identidade, como borramento, extração da silhueta, substituição do indivíduo por uma elipse opaca ou por uma caixa opaca. Eles realizaram testes em uma base própria, com 23 vídeos gravados. Lin et al. [41] investigaram uma solução focada em privacidade com a utilização de silhueta. Eles aplicaram um classificador KNN somado a um temporizador que verifica se a pose do indivíduo voltou ao normal. Os testes foram realizados por voluntários do laboratório.

Alguns trabalhos utilizaram redes neurais convolucionais, como o caso de Anishchenko [3], que implementou uma adaptação da arquitetura AlexNet para detectar quedas no conjunto de dados FDD [11]. Fan et al. [17] utilizaram uma CNN para monitorar e avaliar o grau de completude de um evento. Uma pilha de quadros de um vídeo foi utilizada em uma arquitetura VGG-16 e seu resultado foi associado com o primeiro quadro da

pilha. O método foi testado em dois conjuntos de dados: FDD [11] e Hockey Fights [51]. Seus resultados foram reportados em termos de completude das quedas.

Huang et al. [25] fizeram o uso do algoritmo OpenPose para obter as coordenadas das juntas do corpo. Dois classificadores (SVM e VGG-16) foram comparados para classificar as coordenadas. Os experimentos foram realizados nas bases URFD [32] e FDD [11]. Li et al. [39] criaram uma modificação da arquitetura de CNN, AlexNet. A solução foi testada no conjunto de dados URFD [32], além disso os autores também reportaram que a solução classificou entre ADLs e quedas em tempo real.

Min et al. [48] utilizaram uma R-CNN (CNN de regiões) para analisar uma cena, que gera relações espaciais entre a mobília e o ser humano em cena, tal que a relação espacial é classificada. Os autores experimentaram em três conjuntos de dados: o URFD [32], KTH [57] e uma base criada por eles. Núñez-Marcos et al. [53] realizaram a classificação com uma VGG-16. Os autores calcularam o fluxo óptico denso, que serviu de característica para a rede classificar. Eles testaram o método nas bases de dados URFD [32] e FDD [11].

Coincidentemente, todos os trabalhos que utilizaram redes neurais recorrentes fizeram o uso da mesma arquitetura. Lie et al. [40] aplicaram uma rede neural recorrente, com células LSTM (*Long Short-Term Memory*), para classificar a postura do indivíduo. A postura foi extraída por uma CNN e os experimentos foram realizados em uma base de dados criada pelos autores.

Shojaei-Hashemi et al. [59] utilizaram um aparelho Microsoft Kinect para obter a informação de postura do indivíduo e uma rede neural recorrente LSTM como classificador. Os experimentos foram realizados no conjunto de dados NTU RGB+D. Além disso, os autores relataram a utilização do Kinect como uma vantagem, pois a extração da postura pode ser realizada em tempo real. Lu et al. [42] propuseram a aplicação de uma LSTM logo em seguida de uma CNN 3D. Os autores realizaram testes nas bases URFD [32] e FDD [11].

Outros algoritmos de aprendizado de máquina, como o K-vizinhos mais próximos também foram encontrados para o problema de detecção de quedas. Kwolek e Kepski [33] fizeram o uso de uma combinação entre acelerômetro e Kinect. Durante a suspeita de uma queda, o acelerômetro ultrapassa um limiar e, a partir daí, a câmera Kinect começa a capturar quadros de profundidade da cena. Os autores compararam a classificação dos quadros entre KNN e SVM e testaram no conjunto de dados URFD [32] e em uma base independente.

Sehairi et al. [58] desenvolveram uma máquina de estados finita, para estimar a posição da cabeça do ser humano a partir da silhueta extraída. Os testes foram realizados na base de dados FDD [11].

A aplicação de filtros de Markov também foi utilizada na detecção de quedas, como no trabalho de Anderson et al. [2], em que a silhueta do indivíduo foi extraída para que suas características fossem classificadas pelo filtro de Markov. Os experimentos foram realizados em bases próprias.

Zerrouki e Houacine [81] descreveram as características do corpo por meio de coeficientes de curvelet e da razão entre as áreas do corpo. Um classificador SVM realizou a identificação da postura e o filtro de Markov discriminou entre quedas ou não quedas. Os autores reportaram experimentações nos conjuntos de dados URFD [32] e FDD [11].

Além dos métodos citados anteriormente, os seguintes trabalhos fizeram uso de diversas técnicas, como Yu et al. [79], que obtiveram suas características pela aplicação de técnicas de rastreamento de cabeça e análise de variação de forma. As características serviram de entrada a um classificador Gaussiano. Os autores criaram uma base própria para os testes. Zerrouki et al. [80] segmentaram os quadros entre frente e fundo e aplicaram mais uma segmentação sobre o corpo humano, dividindo-o em cinco partições. As segmentações do corpo foram passadas a um classificador AdaBoost, que obteve 96% de acurácia no conjunto URFD [32]. Finalmente, Xu et al. [78] publicaram um *survey* avaliando diversos sistemas de detecção de quedas.

Capítulo 3

Metodologia

Neste capítulo, descrevemos dois métodos multicanais que foram propostos e utilizados para a detecção de quedas em vídeos neste trabalho: (i) método 2D utilizando a arquitetura convolucional VGG-16 e (ii) um método 3D utilizando a arquitetura Inception 3D.

A Figura 3.1 exibe uma visão geral dos dois métodos propostos que se baseiam na hipótese levantada por Goodale e Milner [20], na qual o córtex visual humano é composto por duas partes que focam em processar diferentes aspectos da visão. Essa mesma hipótese inspirou Simonyan e Zisserman [61] a testar redes neurais com vários canais de informação que simulassem o córtex visual. Apesar da utilização de multicanais ter sido inspirada no córtex visual humano, as características extraídas quebram a analogia e representam um outro espaço de informações.

3.1 Pré-Processamento

Na literatura relacionada ao aprendizado de máquina profundo, o conhecimento sobre a influência positiva de alguns passos de pré-processamento sobre a eficácia das redes é ubiquamente presente. Dessa maneira, alguns processamentos foram selecionados de forma a melhor atenderem à tarefa em questão.

Neste trabalho, o pré-processamento, representado pelo bloco em cor verde na Figura 3.1, consiste na extração de características que possam capturar os vários aspectos de uma queda e na aplicação de técnicas de aumentação de dados (*data augmentation*).

3.1.1 Extração de Características

Neste trabalho, as características de estimação de pose, ritmo visual, saliência e fluxo óptico, apresentadas no Capítulo 2, foram utilizadas e obtidas das formas a seguir. A estimação de pose foi extraída por uma abordagem *bottom-up*, com o algoritmo OpenPose de Cao et al. [6], o algoritmo está publicamente disponível e apresentou boa performace ao estimar a pose de vários indivíduos no quadro. Os quadros extraídos foram alimentados à rede um por vez e os resultados são obtidos quadro-a-quadro (Figura 2.8). A escolha desta característica se deu pela observação empírica de que a detecção de quedas pode se beneficiar com informações da pose do idoso. Para o ritmo visual, um algoritmo

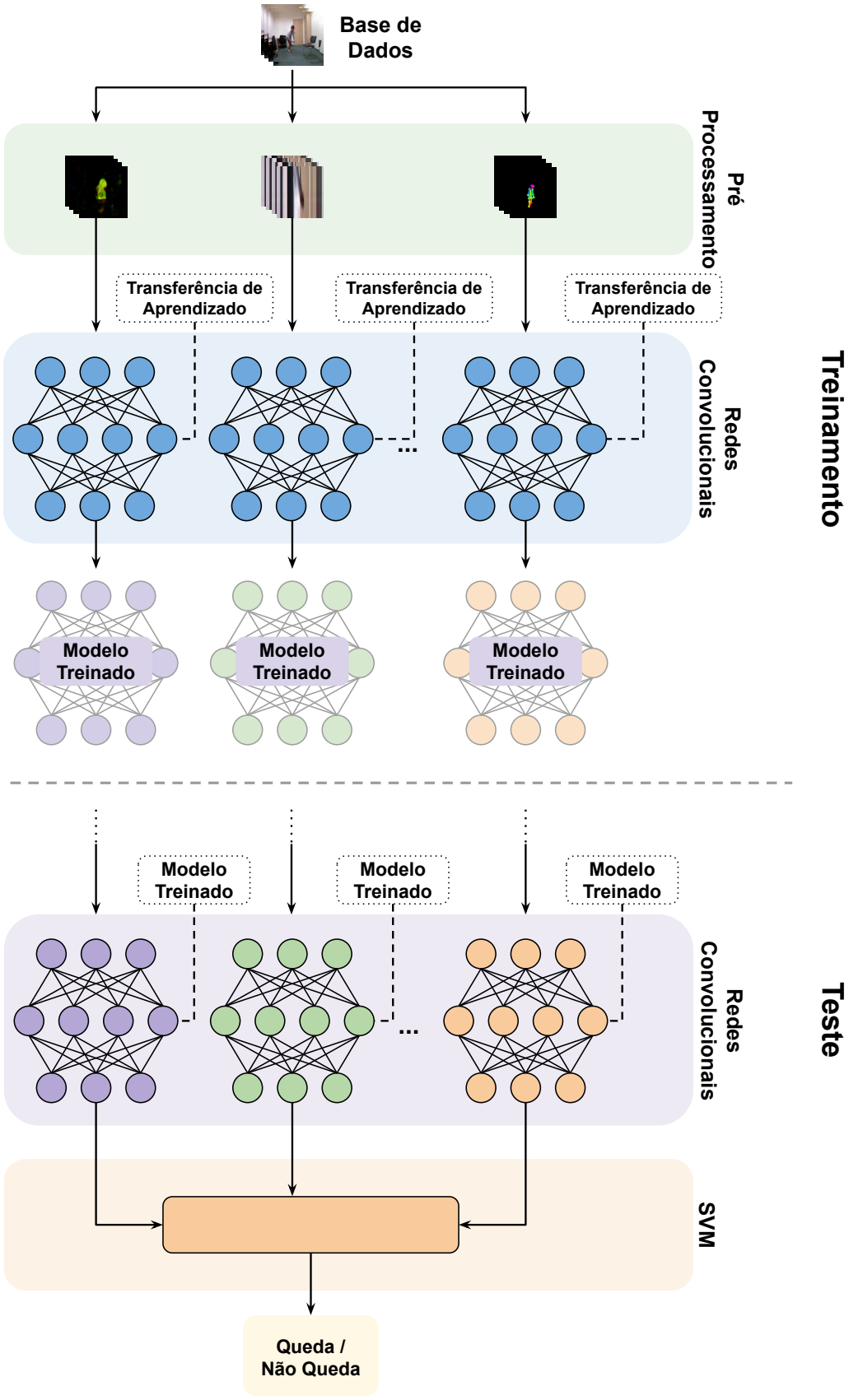


Figura 3.1: Diagrama geral para ambos os métodos propostos, ilustrando as fases de treinamento e teste.

foi implementado tal que cada vídeo possui somente um ritmo visual. Esse quadro de ritmo foi alimentado repetidas vezes para a rede e emparelhado com os quadros de outras características (Figura 2.6). A escolha do ritmo vem do fato dele comprimir ao mesmo tempo informações temporais e espaciais, que supomos ser essenciais para detecção das quedas. O mapa de saliência foi obtido com a utilização da técnica de *SmoothGrad*, proposta e implementada por Smilkov et al. [63], que atua sobre uma técnica previamente existente de Sundararajan et al. [65]. O algoritmo foi escolhido pela forma como foi treinado, os autores treinaram a rede que extraí a saliência em uma base com vídeos egocêntricos, visão da primeira pessoa, de forma que a rede ignorasse a plano de fundo dos quadros, similarmente, os eventos que quedas acontecem destacados do plano de fundo, dessa forma o mapa de saliência mantém informações espaciais do *foreground* da cena. Os quadros de saliência foram alimentados à rede de forma emparelhada com os outros canais (Figura 2.7).

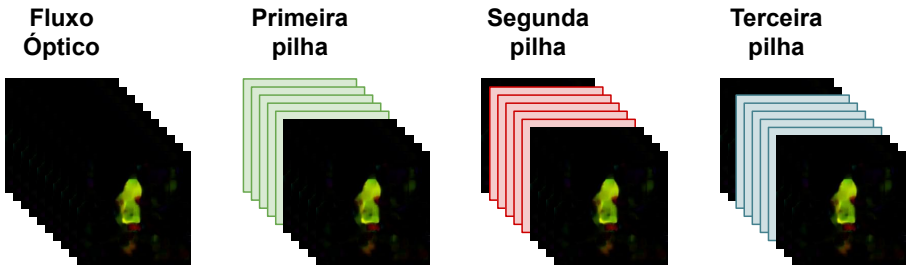


Figura 3.2: Ilustração da janela deslizante e seu movimento entre os quadros de vídeo.

A extração do fluxo óptico foi realizada com o algoritmo proposto por Farnebäck [18], que descreve o fluxo óptico denso (Figura 2.4). O algoritmo é publicamente disponível na biblioteca OpenCV e está presente em vários outros trabalhos da literatura que extraíram o fluxo óptico. A assumpção de que a classificação de eventos em vídeos está diretamente relacionada com a característica temporal das ações, nos motivou a escolher o fluxo óptico para descrever a relação temporal entre os quadros. Como um evento de queda acontece no decorrer de vários quadros e o fluxo representa somente a relação entre dois quadros, nós utilizamos uma abordagem de janela deslizante, sugerida por Wang et al. [74]. A janela deslizante alimenta a rede com uma pilha de dez quadros de fluxo óptico. A primeira pilha contém quadros de 0 ao 9 e a segunda pilha quadros de 1 ao 10, e assim por diante com *stride* igual a 1 (Figura 3.2). Dessa forma, cada vídeo possui $N - 10 + 1$ pilhas, assumindo N como o número de quadros de um vídeo, sendo que os últimos nove quadros não contribuem na avaliação. O resultado de cada pilha foi associado ao primeiro quadro da mesma. Dessa forma, ela pode ser emparelhada com os outros canais da rede.

3.1.2 Aumentação de Dados

Técnicas de aumentação de dados (*data augmentation*) foram utilizadas, quando aplicáveis, durante o pré-processamento da fase de treinamento. Os seguintes processos de aumentação foram empregados: espelhamento sobre o eixo vertical, transformação de perspectiva, corte e adição de bordas espelhadas, adição dos valores -20 e 20 aos pixels, adição dos valores -15 e 15 à matiz e à saturação.

Somente o canal de RGB (*Red-Green-Blue*) passou pelo processo, pois os outros canais sofreriam de forma negativa. Por exemplo, a informação do fluxo óptico depende estritamente da relação entre quadros e sua magnitude é expressa pelo brilho do pixel, um espelhamento quebraria a continuidade entre quadros e uma adição distorceria a magnitude do vetor.

3.2 Treinamento

Em função da pequena quantidade de dados disponíveis para nossa experimentação, a etapa de treinamento do método proposto necessita da utilização da técnica de transferência de aprendizado, descrita no Capítulo 2. A técnica em questão foi utilizada de forma semelhante tanto no método para a arquitetura VGG-16 quanto para a arquitetura Inception 3D.

Para o treinamento da Inception 3D, os pesos foram transferidos da base de dados ImageNet [13]. A partir deste ponto, a rede foi treinada sem congelar nenhuma de suas camadas. Na arquitetura VGG-16, os pesos foram novamente adquiridos da base ImageNet [13], porém, somente os pesos das primeiras 14 camadas da rede foram mantidos. A partir deste ponto, essas camadas foram novamente treinadas na base UCF101 [64] e seus pesos congelados, de forma que a etapa de treinamento ocorresse somente nas duas camadas totalmente conexas finais. O processo de transferência para as primeiras 14 camadas e o congelamento delas está ilustrado na Figura 3.3.

A escolha dos conjuntos de dados para a transferência do aprendizado foi baseada na qualidade do conjunto e na disponibilidade do mesmo. Em especial, no caso da base UCF101, a escolha se deu pelo fato de ela conter informações de vídeo sobre ações humanas e a transferência dessas informações poder impactar positivamente para a classificação de quedas.

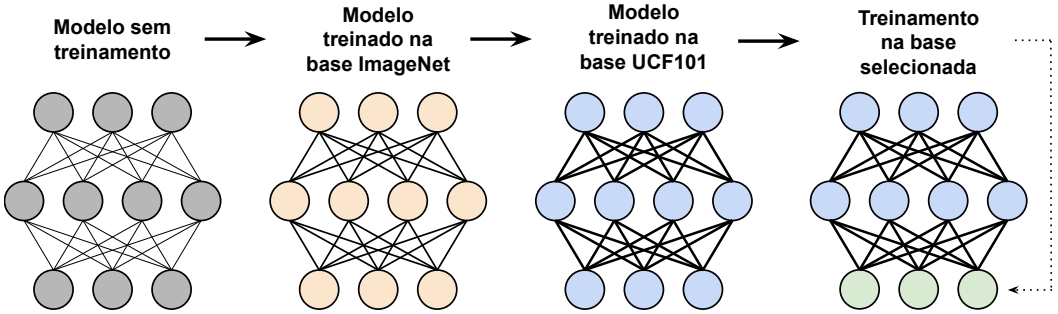


Figura 3.3: Processo de transferência de aprendizado. Da esquerda para direita, o modelo inicia sem nenhum treinamento, a seguir ele é treinado na base ImageNet [13], e posteriormente ele é treinado no conjunto UCF101 [64]. A direita, ilustração do congelamento das primeiras camadas da VGG-16, representadas pelos círculos de cor azul, bem como o treinamento realizado nas duas camadas finais, representadas pela cor verde.

Após a transferência de aprendizado, ambas as arquiteturas foram alimentadas com as características extraídas. Cada canal possui uma rede dedicada a aprender suas características e o número de canais simultâneos varia de dois a três, como ilustrado na

Figura 3.1. No caso da arquitetura VGG-16, somente as duas camadas finais foram treinadas, como explicado anteriormente e ilustrado na direita da Figura 3.3. No caso da arquitetura Inception 3D, toda a rede foi treinada na nova base de dados. Os pesos de cada base de dados, respectivos a cada canal, foram então salvos para serem reutilizados na fase de teste.

3.3 Teste

Alguns trabalhos da literatura revisada [6,69] utilizaram as características pré-selecionadas por esta metodologia para detectar ações humanas, entretanto, elas foram utilizadas de forma isolada. Estes trabalhos, juntamente com o trabalho de Simonyan et al. [61], pavimentaram nossa motivação na proposição de uma metodologia que unisse, de alguma forma, estas características, agindo como um ator ponderador dos canais da rede. Essa união pode ser realizada de diversas maneiras, desde uma simples média entre os canais, passando por uma média ponderada, até alguns métodos automáticos, por exemplo, a forma utilizada nesta metodologia: a aplicação de SVM para classificar os resultados dos canais da rede.

A fase de teste, ilustrada na Figura 3.1, utiliza como entrada os mesmos canais de características da fase de treinamento. Porém, as semelhanças se limitam a isso: ambas as etapas de aumentação de dados e de transferência de aprendizado não foram executadas nesta fase. Os pesos obtidos pelo treinamento foram carregados e as arquiteturas de CNN os utilizam para classificarem cada canal separadamente, quadro-a-quadro. Seguinte à etapa de classificação das CNNs, um classificador SVM foi aplicado para avaliar a concatenação dos vetores de saída das CNNs e, por fim, classificar os quadros entre queda e não-queda. Os parâmetros do SVM foram os seguintes: função *kernel* de base radial, regularização ou C igual a 1, mesmo peso para ambas as classes e gama (γ) conforme definido na Equação 3.1.

$$\gamma = \frac{1}{\textit{número de características} * \textit{variância da entrada}} \tag{3.1}$$

Capítulo 4

Experimentos e Resultados

Neste capítulo, nós apresentamos as experimentações realizadas sobre o método proposto. Na seção seguinte, nós descrevemos os conjuntos de dados utilizados nos experimentos. Posteriormente, nós reportamos os resultados que cada arquitetura obteve nestas bases de dados. Ao final, os resultados são comparados com a literatura relacionada e suas contribuições para a detecção de quedas discutidas.

4.1 Bases de Dados

Durante a revisão da literatura relacionada, várias bases de dados foram encontradas, porém, nem todas estão disponíveis publicamente, algumas referências para seus dados estão inativas ou os autores não responderam a nossa tentativa de contato. Dessa maneira, duas bases de dados relacionadas a quedas humanas foram selecionadas: (i) URFD [32] e (ii) FDD [11].

URFD

Publicada por Kwolek e Kepski [32], a base de dados URFD (*University of Rzeszow Fall Detection Dataset*) é composta de 70 sequências de vídeo, sendo 30 de quedas e 40 de atividades do dia-a-dia. Cada vídeo possui 30 quadros por segundo (FPS), com resolução de 640×240 pixels e duração variada.

As sequências de queda foram gravadas com um acelerômetro e duas câmeras Microsoft Kinect, uma com visão horizontal da cena e uma com visão de cima para baixo, no teto. As atividades da vida diária foram gravadas com uma única câmera de visão horizontal e um acelerômetro. As informações do acelerômetro foram excluídas dos experimentos por extrapolarem o escopo do projeto. A Figura 4.1 ilustra os cinco cenários de ADLs e um cenário de queda.

A base de dados está anotada com as seguintes informações:

- Postura do sujeito (não deitado, deitado no chão e transição).
- Razão entre altura e largura da caixa delimitadora.
- Razão entre eixo máximo e mínimo.

Figura 4.1: Ambientes da base de dados URFD [32]. (a) ilustração dos cenários de quedas gravados pela câmera horizontal; (b) ilustração dos cinco ambientes onde as ADLs foram gravadas.

- Razão da ocupação do sujeito na caixa delimitadora.
- Desvio padrão dos pixels para o centroide dos eixos X e Z.
- Razão entre altura do sujeito no quadro com a altura do sujeito em pé.
- Altura do sujeito.
- Distância do centro do sujeito ao chão.

FDD

O conjunto de dados FDD (*Fall Detection Dataset*) foi publicado por Charfi et al. [11] e contém 191 sequências de vídeo, com 143 sendo de quedas e 48 de atividades do dia-a-dia. Cada vídeo possui 25 FPS, com resolução de 320×240 pixels e duração variada.

Todas as sequências foram gravadas com uma única câmera, em quatro ambientes diferentes: casa, copa, escritório e sala de aula, ilustrados na Figura 4.2. Além disso, a base apresenta três protocolos de experimentação: (i) em que treinamento e teste são criados com vídeos dos ambientes casa e copa, (ii) em que o treinamento é composto de vídeos da copa e o teste com vídeos do escritório e da sala de aula e (iii) em que o treinamento contém vídeos da copa, do escritório e da sala de aula e o teste contém

vídeos de escritório e da sala de aula. Pelo fato da base URFD possuir somente um cenário onde as quedas foram gravados, e a FDD possuir vários, nós podemos afirmar que, por contraste, o conjunto URFD é um conjunto fácil e o conjunto FDD é um conjunto difícil, o que é refletido nos resultados dos experimentos.

Figura 4.2: Ambientes da base de dados FDD [11]. (a) ilustração dos vídeos contendo quedas; (b) ilustração dos vídeos de ADLs.

A base de dados está anotada com as seguintes informações:

- Quadro inicial da queda.
- Quadro final da queda.
- Altura, largura e coordenadas do centro da caixa delimitadora de cada quadro.

4.1.1 Métricas de Avaliação

Neste trabalho, nós abordamos o problema da detecção de quedas como uma classificação binária, em que um classificador deve decidir se um quadro de vídeo corresponde a uma queda ou não. Para tal, as métricas escolhidas e suas respectivas equações foram as seguintes: (i) precisão (Equação 4.1), (ii) sensibilidade (Equação 4.2), (iii) acurácia (Equação 4.3) e (iv) acurácia balanceada (Equação 4.4). Nas equações, os seguintes termos foram abreviados: TP (*true positive*), FP (*false positive*), TN (*true negative*) e FN

(*false negative*).

$$Precisão = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.1}$$

$$Sensibilidade = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{4.2}$$

$$Acurácia = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{4.3}$$

$$Acurácia\ Balanceada = \frac{1}{\sum \hat{w_i}} \sum_i (\hat{y_i} = y_i)\hat{w_i} \tag{4.4}$$

em que

$$\hat{w_i} = \frac{w_i}{\sum_j 1(y_j = y_i)w_j} \tag{4.5}$$

Essas métricas foram escolhidas pela necessidade de comparar nossos resultados com os encontrados na literatura que, em sua maioria, reportaram apenas: precisão, sensibilidade e acurácia. Como ambas as bases de dados são desbalanceadas, de maneira que a classe negativa possui mais do que o dobro de amostras do que a classe positiva (classe de quedas), nós selecionamos a acurácia balanceada para contrapor este fato, uma vez que ela é invariável ao desbalanceamento dos conjuntos de dados.

4.1.2 Recursos Computacionais

As arquiteturas propostas foram implementadas na linguagem de programação Python [73], que foi escolhida devido a sua ampla disponibilidade de bibliotecas para aplicações de análise de imagens e aprendizado profundo. Mais especificamente, foram utilizadas as bibliotecas, foram utilizadas SciPy [26], NumPy [54], OpenCV [5] e Keras [12].

Os algoritmos de aprendizado profundo são conhecidos por serem computacionalmente intensivos. Seus treinamentos e experimentos requerem mais poder computacional do que um *notebook* convencional pode fornecer e, portanto, foram realizados na nuvem em uma máquina alugada da Amazon AWS, g2.2xlarge, com as seguintes especificações: 1x GPU nVidia GRID K520 (Kepler), 8x vCPUs e 15GB de memória RAM.

4.2 Experimentos e Resultados

A seguir, os experimentos realizados são apresentados. Os resultados foram separados de acordo com a arquitetura utilizada e reportados na ordem: VGG-16 e Inception 3D. Dentro da seção de cada arquitetura, as combinações de multicanais foram comparadas entre si. Posteriormente, para as combinações multicanais, nós também realizamos testes cruzados entre as bases de dados e, ao conhecimento dos autores, esta comparação é inédita entre os conjuntos de dados selecionados. Por fim, os melhores resultados foram comparados aos trabalhos da literatura relacionada. As bases de dados foram divididas nas proporções: 65% para treinamento, 15% para validação e 20% para teste.

Resultados para VGG-16

As tabelas a seguir comparam as abordagens multicanais sobre a arquitetura VGG-16, com parâmetros de 500 épocas, empregando-se *early stopping* com valor de paciência igual a 10, taxa de aprendizado de 10^{-4}, *mini-batches* de 2^{10}, 50% de *dropout*, otimizador Adam, com treinamento para minimizar a função de perda da validação. Os parâmetros foram reutilizados de trabalhos semelhantes de classificação de eventos. Em ordem, os resultados foram reportados sobre a base URFD, seguidos dos resultados do conjunto FDD.

A Tabela 4.1 exibe os resultados obtidos para a base URFD, na qual a combinação dos canais de fluxo óptico e ritmo visual obteve o melhor resultado. Em sua maioria, os melhores resultados foram obtidos dos canais de fluxo, ritmo e RGB, ao contrário dos canais contendo a pose.

Tabela 4.1: Comparação VGG-16 dos multicanais em relação à base de dados URFD.

Canais	Precisão	Sensibilidade	Acurácia	Acurácia Balanceada
OF VR	**1,00**	0,98	**0,98**	**0,96**
OF RGB VR	**1,00**	0,97	0,97	0,94
OF RGB	0,99	0,98	0,97	0,94
RGB VR	**1,00**	0,90	0,90	0,92
OF RGB SA	0,99	0,98	0,97	0,91
OF RGB PE	0,99	0,98	0,97	0,90
OF SA	0,99	0,98	0,97	0,90
RGB SA	0,99	0,94	0,94	0,90
SA VR	0,99	0,95	0,94	0,90
OF PE	0,99	**0,99**	**0,98**	0,88
VR PE	0,99	0,98	0,97	0,88
RGB PE	0,99	0,97	0,96	0,87
SA PE	0,99	0,97	0,96	0,87

OF: Fluxo óptico, VR: Ritmo visual, SA: Saliência, PE: Estimativa de pose. Os resultados estão em ordem decrescente de acurácia balanceada e os maiores de cada coluna estão destacados em negrito.

A eficácia satisfatória do canal de fluxo vai de acordo com a hipótese de que a classificação de eventos é dependente de uma relação temporal entre os quadros e o mesmo ocorre com os canais espaciais, indicando que as informações temporal e espacial se complementam. Porém, tendo em vista a natureza do canal de pose, era esperado que a combinação entre pose e fluxo óptico gerasse bons resultados, enquanto nenhuma das combinações envolvendo a pose obteve mais de 90% de acurácia balanceada. Esse resultado é discutido posteriormente, durante os testes cruzados.

Os resultados relativos à base FDD são apresentados na Tabela 4.2. Nesta instância, o melhor resultado foi a combinação da saliência, que é espacial, e o ritmo visual, que é espaço-temporal. Nestes resultados, os canais de RGB sofreram uma queda brusca em eficácia, ao contrário dos canais de pose e saliência, que se encontraram entre os três melhores resultados. Apesar dos canais espaciais de saliência e pose terem ascendido entre os melhores resultados, há uma peculiaridade a ser notada: ambos os canais possuem

Tabela 4.2: Comparação VGG-16 dos multicanais em relação à base de dados FDD.

Canais	Precisão	Sensibilidade	Acurácia	Acurácia Balanceada
SA VR	**1,00**	**0,97**	**0,98**	**0,95**
OF SA	**1,00**	0,96	0,96	0,94
OF PE	0,99	0,95	0,95	0,92
SA PE	0,99	0,89	0,89	0,92
RGB SA	0,99	0,96	0,96	0,91
OF VR	0,99	0,95	0,95	0,88
OF RGB	0,99	0,94	0,94	0,88
OF RGB PE	0,99	0,92	0,92	0,88
OF RGB SA	0,99	0,91	0,91	0,87
OF RGB VR	0,99	0,90	0,89	0,87
RGB PE	0,99	0,81	0,82	0,83
RGB VR	0,99	0,84	0,83	0,80
VR PE	0,99	0,78	0,78	0,78

OF: Fluxo óptico, VR: Ritmo visual, SA: Saliência, PE: Estimativa de pose. Os resultados estão em ordem decrescente de acurácia balanceada e os maiores de cada coluna estão destacados em negrito.

pouca ou nenhuma informação do fundo, assim como os canais do melhor resultado, saliência e fluxo. Nós voltaremos a este ponto na discussão dos resultados cruzados.

Os testes cruzados, em que o treinamento ocorre em um conjunto de dados e o teste no outro, foram realizados em ordem: (i) treinamento na base URFD com teste na FDD e (ii) treinamento no conjunto FDD com teste na base URFD.

A Tabela 4.3 é referente ao primeiro teste cruzado. Há uma queda significativa na acurácia balanceada, saltando de 95% na Tabela 4.2 para 63%. Nós acreditamos que esta queda é um efeito colateral da qualidade das bases de dados, pois este resultado foi obtido do treino na base fácil (URFD). Nessa base, os vídeos de quedas foram gravados com diferentes atores, entretanto, no mesmo ambiente e com a câmera na mesma posição. Por contraste entre as bases, podemos afirmar que a URFD é fácil e a FDD é difícil. Ao treinar na base fácil e testar na base difícil, a maioria dos canais não conseguiu discriminar entre queda e não queda, obtendo 50% de acurácia balanceada.

Os melhores resultados são um reflexo da homogeneidade da base de dados, pois somente aqueles com pouco acesso ao fundo do quadro conseguiram operar em um novo cenário. A deficiência do conjunto de dados explica a completa inversão dos resultados da combinação de saliência com pose, obtendo o pior resultado na Tabela 4.2, em que a informação espacial era muito mais fácil de ser classificada e o melhor resultado no teste cruzado.

O segundo teste cruzado é reportado na Tabela 4.4. Novamente, houve uma queda da acurácia balanceada, de 96% na Tabela 4.1 para 78%, que é esperada durante uma troca de contexto dos dados. A importância da informação temporal é mantida, tendo em vista a presença do fluxo óptico entre os melhores resultados. Entre os testes cruzados, houve um aumento da acurácia balanceada, o que pode ser explicado ao considerarmos

Tabela 4.3: Comparação VGG-16 dos multicanais em teste cruzado, com treinamento na base URFD e teste no conjunto FDD.

Canais	Precisão	Sensibilidade	Acurácia	Acurácia Balanceada
SA PE	**0,96**	**1,00**	**0,96**	**0,63**
OF SA	0,95	**1,00**	0,95	0,55
OF PE	0,95	**1,00**	0,95	0,54
OF RGB PE	0,95	**1,00**	0,95	0,51
OF RGB VR	0,95	**1,00**	0,95	0,50
OF RGB SA	0,95	**1,00**	0,95	0,50
OF RGB	0,95	**1,00**	0,95	0,50
OF VR	0,95	**1,00**	0,95	0,50
RGB PR	0,95	**1,00**	0,95	0,50
RGB VR	0,95	**1,00**	0,95	0,50
RGB SA	0,95	**1,00**	0,95	0,50
SA VR	0,95	**1,00**	0,95	0,50
VR PE	0,95	**1,00**	0,95	0,50

OF: Fluxo óptico, VR: Ritmo visual, SA: Saliência, PE: Estimativa de pose. Os resultados estão em ordem decrescente de acurácia balanceada e os maiores de cada coluna estão destacados em negrito.

que, nesta instância, a rede foi treinada na base difícil e testada na base fácil.

Tabela 4.4: Comparação VGG-16 dos multicanais em teste cruzado, com treinamento na base FDD e teste no conjunto URFD.

Canais	Precisão	Sensibilidade	Acurácia	Acurácia Balanceada
OF PE	**0,98**	0,89	0,88	**0,78**
OF RGB PE	0,97	0,99	**0,96**	0,68
VR PE	0,97	0,94	0,91	0,68
OF SA	0,95	0,99	0,95	0,54
SA PE	0,95	0,94	0,90	0,54
RGB PE	0,95	**1,00**	0,95	0,52
OF RGB SA	0,95	**1,00**	0,95	0,51
OF RGB VR	0,95	**1,00**	0,95	0,50
OF RGB	0,95	**1,00**	0,95	0,50
OF VR	0,95	**1,00**	0,95	0,50
RGB VR	0,95	**1,00**	0,95	0,50
RGB SA	0,95	**1,00**	0,95	0,50
SA VR	0,95	**1,00**	0,95	0,50

OF: Fluxo óptico, VR: Ritmo visual, SA: Saliência, PE: Estimativa de pose. Os resultados estão em ordem decrescente de acurácia balanceada e os maiores de cada coluna estão destacados em negrito.

Em geral, houve uma melhora, vide a maioria das acurácias balanceadas acima de 50%. Dentre os canais com os melhores resultados, ambos os testes cruzados são consistentes,

pois as combinações sem acesso ao fundo do quadro continuam superando as opostas. Porém, alguns canais espaciais ascenderam nos resultados, que é o caso do RGB. Isso pode ser explicado pela heterogeneidade da base FDD, fazendo com o que treinamento focasse menos em aspectos do ambiente e generalizando melhor seu aprendizado.

Resultados para Inception 3D

As tabelas a seguir comparam as abordagens multicanais sobre a arquitetura Inception 3D, utilizando os seguintes parâmetros: 500 épocas, empregando-se *early stopping* com valor de paciência igual a 10, taxa de aprendizado de 10^{-5}, *mini-batches* de tamanho 192, 50% de *dropout*, otimizador Adam, com treinamento para minimizar a função de perda da validação. Em ordem, os resultados foram reportados sobre a base URFD, seguidos dos resultados do conjunto FDD.

O resultados obtidos sobre a base URFD são exibidos na Tabela 4.5, na qual a combinação dos canais de fluxo óptico e RGB obtiveram o melhor resultado. Considerando a natureza da arquitetura em si, a melhora na acurácia balanceada máxima era esperada, uma vez que a arquitetura 3D possui um relação temporal interna, subindo de 96% na Tabela 4.1 para 98%.

Tabela 4.5: Comparação 3D dos multicanais em relação à base de dados URFD.

Canais	Precisão	Sensibilidade	Acurácia	Acurácia Balanceada
OF RGB	**1,00**	**1,00**	0,97	**0,98**
OF VR	0,99	0,96	0,95	0,97
RGB VR	0,99	0,90	0,96	0,97
OF RGB SA	**1,00**	0,99	0,98	0,94
OF RGB VR	0,99	0,99	**0,99**	0,94
OF RGB PE	0,99	0,96	0,96	0,91
OF SA	0,99	0,98	0,94	0,91
SA VR	0,99	0,94	0,94	0,90
RGB SA	0,99	0,95	0,96	0,89
SA PE	0,99	**1,00**	0,91	0,89
RGB PE	0,99	0,99	0,92	0,89
VR PE	0,99	0,96	0,94	0,88
OF PE	0,99	0,97	0,92	0,87

OF: Fluxo óptico, VR: Ritmo visual, SA: Saliência, PE: Estimativa de pose. Os resultados estão em ordem decrescente de acurácia balanceada e os maiores de cada coluna estão destacados em negrito.

De maneira semelhante aos resultados da VGG-16, os canais temporais se encontram entre os melhores resultados. Como o treinamento e o teste desta instância foram realizados no mesmo conjunto de dados, não é surpresa que os canais espaciais obtiveram novamente uma boa eficácia, exceto novamente os canais envolvendo a pose.

A Tabela 4.6 apresenta os resultados referentes ao conjunto de dados FDD. A Inception 3D obteve uma melhora na acurácia balanceada máxima, de 95% na Tabela 4.2 para 98%.

Tabela 4.6: Comparação 3D dos multicanais em relação à base de dados FDD.

Canais	Precisão	Sensibilidade	Acurácia	Acurácia Balanceada
OF SA	0,99	0,99	0,98	**0,98**
SA VR	**1,00**	0,98	0,98	0,96
SA PE	**1,00**	**1,00**	0,97	0,96
RGB SA	**1,00**	0,99	**0,99**	0,95
OF PE	0,99	0,96	0,97	0,93
OF RGB VR	0,99	0,95	0,95	0,91
OF VR	0,99	0,93	0,94	0,91
RGB VR	0,99	0,94	0,89	0,91
OF RGB	0,99	0,91	0,99	0,90
VR PE	0,99	0,89	0,91	0,88
OF RGB PE	0,99	0,84	0,93	0,87
OF RGB SA	0,99	0,85	0,97	0,86
RGB PE	0,99	0,80	0,91	0,84

OF: Fluxo óptico, VR: Ritmo visual, SA: Saliência, PE: Estimativa de pose. Os resultados estão em ordem decrescente de acurácia balanceada e os maiores de cada coluna estão destacados em negrito.

A combinação dos canais fluxo óptico e saliência obtiveram o melhor resultado, mantendo a importância da relação temporal na classificação.

Um fato interessante a ser notado é o de que, ao contrário dos resultados (Tabela 4.2) da VGG-16, os canais aparentam estar mais distribuídos em sua eficácia. Nós acreditamos que esse fato é uma consequência combinada entre a base de dados ser mais heterogênea e pelo fato da arquitetura 3D definir uma relação temporal interna, sofrendo menor influência do fundo.

Os testes cruzados a seguir foram realizados na ordem: (i) treinamento na base URFD com teste na FDD (Tabela 4.7) e (ii) treinamento no conjunto FDD com teste na base URFD (Tabela 4.8). O primeiro teste obteve sua maior acurácia balanceada com o valor de 68% e o segundo teste com o valor de 84%. A maioria das combinações de canais obteve 50%. Ambos os testes apresentaram um cenário semelhante aos encontrados nos testes cruzados da VGG-16, pois as maiores acurácias balanceadas sofreram uma queda em relação aos testes não cruzados, sendo que a rede obteve melhores resultados no conjunto URFD do que na base FDD. Além das similaridades, o fato da Inception 3D assimilar melhor as informações temporais com as espaciais, contribuiu para resultados superiores aos testes cruzados da VGG-16.

Resultados Comparados com a Literatura Relacionada

Nas tabelas a seguir, nós comparamos nossos melhores resultados com os encontrados durante a revisão bibliográfica. As métricas utilizadas foram as mesmas reportadas pelos autores: precisão, sensibilidade e acurácia. Os resultados referentes ao conjunto de dados URFD são reportados na Tabela 4.9, enquanto os referentes à base de dados FDD são apresentados na Tabela 4.10.

Tabela 4.7: Comparação 3D dos multicanais em teste cruzado, com treinamento na base URFD e teste no conjunto FDD.

Canais	Precisão	Sensibilidade	Acurácia	Acurácia Balanceada
OF PE	**0,97**	**1,00**	**0,97**	**0,68**
OF SA	0,96	**1,00**	0,96	0,57
SA PE	0,95	**1,00**	0,96	0,55
OF RGB PE	0,95	**1,00**	0,96	0,50
OF RGB VR	0,95	**1,00**	0,96	0,50
OF RGB SA	0,95	**1,00**	0,96	0,50
OF RGB	0,95	**1,00**	0,96	0,50
OF VR	0,95	**1,00**	0,96	0,50
RGB PE	0,95	**1,00**	0,96	0,50
RGB VR	0,95	**1,00**	0,96	0,50
RGB SA	0,95	**1,00**	0,96	0,50
SA VR	0,95	**1,00**	0,96	0,50
VR PE	0,95	**1,00**	0,96	0,50

OF: Fluxo óptico, VR: Ritmo visual, SA: Saliência, PE: Estimativa de pose. Os resultados estão em ordem decrescente de acurácia balanceada e os maiores de cada coluna estão destacados em negrito.

Tabela 4.8: Comparação 3D dos multicanais em teste cruzado, com treinamento na base FDD e teste no conjunto URFD.

Canais	Precisão	Sensibilidade	Acurácia	Acurácia Balanceada
OF PE	**0,97**	0,90	0,89	**0,84**
VR PE	0,96	0,92	0,90	0,75
OF RGB PE	0,96	0,99	**0,97**	0,72
SA PE	0,96	0,93	0,95	0,60
OF SA	0,95	0,99	0,91	0,54
RGB PE	0,95	0,99	0,91	0,54
OF RGB SA	0,95	**1,00**	0,95	0,50
OF RGB VR	0,95	**1,00**	0,95	0,50
OF RGB	0,95	**1,00**	0,95	0,50
OF VR	0,95	**1,00**	0,95	0,50
RGB VR	0,95	**1,00**	0,95	0,50
RGB SA	0,95	**1,00**	0,95	0,50
SA VR	0,95	**1,00**	0,95	0,50

OF: Fluxo óptico, VR: Ritmo visual, SA: Saliência, PE: Estimativa de pose. Os resultados estão em ordem decrescente de acurácia balanceada e os maiores de cada coluna estão destacados em negrito.

Nossa arquitetura Inception 3D superou ou se igualou aos trabalhos revisados. A arquitetura VGG-16 superou a maioria dos resultados encontrados, porém, ficou abaixo do trabalho de Lu et al. [42]. Como citado no Capítulo 2, o método de Lu et al. [42]

Tabela 4.9: Comparação dos nossos resultados com a literatura para o conjunto de dados URFD.

Autores	Precisão	Sensibilidade	Acurácia
Método proposto Inception 3D	0,99	**0,99**	**0,99**
Lu et al. [42]	-	-	**0,99**
Método proposto VGG-16	**1,00**	0,98	0,98
Panahi e Ghods [55]	0,97	0,97	0,97
Zerrouki e Houacine [81]	-	-	0,96
Harrou et al. [22]	-	-	0,96
Abobakr et al. [1]	**1,00**	0,91	0,96
Bhandari et al. [4]	0,96	-	0,95
Kwolek e Kepski [33]	**1,00**	0,92	0,95
Núñez-Marcos et al. [53]	**1,00**	0,92	0,95
Sase e Bhandari [56]	0,81	-	0,90

Os trabalhos que não reportaram alguns de seus resultados foram retratados com um hífen (-). Os resultados estão em ordem decrescente de acurácia e os maiores de cada coluna estão destacados em negrito.

utiliza a arquitetura de rede neural recorrente LSTM que, assim como a nossa solução, foi desenvolvida para lidar com a relação temporal entre os dados por meio de um mecanismo de memória incluso na rede.

Tabela 4.10: Comparação dos nossos resultados com a literatura para o conjunto de dados FDD.

Autores	Precisão	Sensibilidade	Acurácia
Método proposto Inception 3D	**1,00**	**0,99**	**0,99**
Lu et al. [42]	-	-	**0,99**
Método proposto VGG-16	**1,00**	0,98	0,98
Sehairi et al. [58]	-	-	0,98
Zerrouki e Houacine [81]	-	-	0,97
Harrou et al. [22]	-	-	0,97
Núñez-Marcos et al. [53]	0,99	0,97	0,97
Charfi et al. [10]	0,98	**0,99**	-

Os trabalhos que não reportaram alguns de seus resultados foram retratados com um hífen (-). Os resultados estão em ordem decrescente de acurácia e os maiores de cada coluna estão destacados em negrito.

Capítulo 5

Conclusões e Trabalhos Futuros

Neste trabalho, nós apresentamos e comparamos dois métodos que utilizam redes neurais profundas para a detecção de quedas humanas em sequências de vídeos: um método 2D que utiliza a rede profunda VGG-16 e um método 3D que utiliza a rede Inception V1 3D. Os métodos foram treinados e avaliados em dois conjuntos de dados públicos. Ambas as abordagens superaram ou se igualaram às eficácias dos trabalhos relacionados.

Os resultados apontaram a importância da informação temporal na detecção de quedas, tanto na eficácia dos canais temporais, em especial o fluxo óptico, quanto na melhora obtida pelo método 3D (Questões 1 e 2 de Pesquisa descritas na Seção 1.3). Apesar de ambos os métodos se mostrarem eficazes para detectar quedas nas bases selecionadas, o método 3D, especificamente, generalizou seu aprendizado para um conjunto de dados no qual ele não foi treinado. A habilidade de generalizar o aprendizado indica que o método 3D pode ser considerado um forte candidato para compor um sistema de assistência ao idoso (Questão 4 de Pesquisa descrita na Seção 1.3).

Esse indício se dá pelo fato de que, ao longo de todos os resultados exibidos no Capítulo 4, os canais temporais sempre se mantiveram entre os mais eficazes, com exceção de dois casos mostrados nas Tabelas 4.6 e 4.7, em que o terceiro melhor resultado foi a combinação dos canais espaciais de saliência e de pose. Isso pode ser atribuído ao fato da própria arquitetura proporcionar uma relação temporal entre os dados.

Nossa conclusão sobre a importância da informação espacial na classificação de quedas é corroborada por outros trabalhos encontrados na literatura, como os de Meng et al. [47] e Carreira e Zisserman [9], que afirmaram o mesmo para a classificação de ações em vídeos.

Somado a isso, por superarem os trabalhos revisados, ambos os métodos se mostraram eficazes na detecção de quedas. Em uma instância específica, nosso método se igualou aos resultados do trabalho de Lu et al. [42], no qual o autor faz uso de uma arquitetura LSTM que, assim como a nossa, cria relações temporais entre seus dados.

De maneira inovadora, testes cruzados foram realizados entre os conjuntos de dados. Os resultados desses testes apresentaram uma conhecida, porém, interessante faceta das redes neurais, na qual a função de minimização encontra um mínimo local que não corresponde com o objetivo inicial do processamento. Durante o treinamento, alguns canais da rede, em especial a VGG-16, aprenderam aspectos do fundo das imagens para classificar as quedas, em vez de focarem nos aspectos do indivíduo em questão.

O método do trabalho é corroborado por alguns fatores, tais como: a avaliação pela

métrica de acurácia balanceada, os testes serem realizados em dois conjuntos de dados, a heterogeneidade da base de dados FDD, a execução dos testes cruzados e as comparações entre as várias combinações de canais. Em contrapartida, o trabalho também lida com algumas dificuldades, como: (i) a baixa variabilidade dos vídeos de queda na base URFD, (ii) o fato de que nos testes cruzados muitas combinações de canais obtiveram somente 50% de acurácia balanceada e (iii) a utilização da acurácia simples como meio de comparação com a literatura. Porém, o método proposto suprime essas contrapartidas, mantendo-se relevante (Questão 3 de Pesquisa descrita na Seção 1.3).

A eficácia dos métodos mostra que, se treinados em uma base robusta o suficiente, eles são capazes de extrair os padrões temporais necessários para classificar cenários entre queda e não queda. Admitidamente, há uma queda esperada da acurácia balanceada nos testes cruzados e ainda assim o método 3D obteve os melhores resultados em todos os experimentos nos quais ele participou. Para fins deste trabalho, esta é uma das abordagens mais acuradas para detectar quedas e seria um módulo de grande contribuição a um sistema integrado de assistência ao idoso.

Em relação a trabalhos futuros, alguns pontos podem ser mencionados: (i) explorar outras bases de dados que possam conter maior variedade de cenários e ações, (ii) integrar a detecção de quedas a um sistema multiclasse, (iii) experimentar com canais de características mais baratos de serem extraídos, (iv) adaptar o método de forma a funcionar em tempo real, seja por canais mais baratos ou uma rede mais leve e (v) lidar com os vídeos em fluxo, em vez de clipes, pois, em um cenário real, a câmera alimentaria continuamente o sistema, o que desbalancearia as classes ainda mais.

As contribuições deste trabalho se somam na forma de dois métodos para a detecção de quedas em seres humanos, implementados e publicamente disponíveis no repositório [37]. As experimentações entre multicanais e entre bases de dados, que geraram não apenas uma discussão sobre quais métricas são mais adequadas para avaliar as soluções, mas também uma discussão sobre a qualidade das bases utilizadas nestes experimentos.

Referências Bibliográficas

[1] A. Abobakr, M. Hossny, and S. Nahavandi. A Skeleton-Free Fall Detection System from Depth Images using Random Decision Forest. *IEEE Systems Journal*, 12(3):2994–3005, 2017.

[2] D. T. Anderson, J. M. Keller, M. Skubic, X. Chen, and Z. He. Recognizing Falls from Silhouettes. *International Conference of the IEEE Engineering in Medicine and Biology Society*, 2006.

[3] L. Anishchenko. Machine Learning in Video Surveillance for Fall Detection. In *Ural Symposium on Biomedical Engineering, Radioelectronics and Information Technology*, pages 99–102. IEEE, 2018.

[4] S. Bhandari, N. Babar, P. Gupta, N. Shah, and S. Pujari. A Novel Approach for Fall Detection in Home Environment. In *IEEE 6th Global Conference on Consumer Electronics*, pages 1–5. IEEE, 2017.

[5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[6] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[7] S. Carneiro, G. Silva, G. Leite, R. Moreno, S. Guimaraes, and H. Pedrini. Deep Convolutional Multi-Stream Network Detection System Applied to Fall Identification in Video Sequences. In *15th International Conference on Machine Learning and Data Mining (MLDM)*, pages 681–695, New York-NY, USA, July 2019.

[8] S. Carneiro, G. Silva, G. Leite, R. Moreno, S. Guimaraes, and H. Pedrini. Multi-Stream Deep Convolutional Network Using High-Level Features Applied to Fall Detection in Video Sequences. In *26th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 293–298, Osijek, Croatia, June 2019.

[9] J. Carreira and A. Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *Conference on Computer Vision and Pattern Recognition*, pages 6299–6308. IEEE, 2017.

[10] I. Charfi, J. Miteran, J. Dubois, M. Atri, and R. Tourki. Definition and Performance Evaluation of a Robust SVM Based Fall Detection Solution. *International Conference on Signal Image Technology and Internet Based Systems*, 12:218–224, 2012.

[11] I. Charfi, J. Miteran, J. Dubois, M. Atri, and R. Tourki. Optimized Spatio-Temporal Descriptors for Real-Time Fall Detection: Comparison of Support Vector Machine and Adaboost-based Classification. *Journal of Electronic Imaging*, 22(4):041106, 2013.

[12] F. Chollet. Keras, 2015. `https://keras.io`.

[13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A Large–Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[14] L. Deng and D. Yu. Deep Learning: Methods and Applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387, 2014.

[15] A. Edgcomb and F. Vahid. Automated Fall Detection on Privacy-Enhanced Video. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 252–255, 2012.

[16] K. Fan, P. Wang, and S. Zhuang. Human Fall Detection Using Slow Feature Analysis. *Multimedia Tools and Applications*, 78(7):9101–9128, 2019.

[17] Y. Fan, G. Wen, D. Li, S. Qiu, and M. D. Levine. Early Event Detection based on Dynamic Images of Surveillance Videos. *Journal of Visual Communication and Image Representation*, 51:70–75, 2018.

[18] G. Farnebäck. Two–Frame Motion Estimation Based on Polynomial Expansion. In *Scandinavian Conference on Image Analysis*, 2003.

[19] S. Gasparrini, E. Cippitelli, E. Gambi, S. Spinsante, J. Wåhslén, I. Orhan, and T. Lindh. Proposal and Experimental Evaluation of Fall Detection Solution Based on Wearable and Depth Data Fusion. In *International Conference on ICT Innovations*, pages 99–108. Springer, 2015.

[20] M. A. Goodale and A. D. Milner. Separate Visual Pathways for Perception and Action. *Trends in Neurosciences*, 15(1):20–25, 1992.

[21] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. *Deep Learning*. MIT Press, 2016.

[22] F. Harrou, N. Zerrouki, Y. Sun, and A. Houacine. Vision-based Fall Detection System for Improving Safety of Elderly People. *IEEE Instrumentation & Measurement Magazine*, 20(6):49–55, 2017.

[23] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[24] D. L. Heymann, T. Prentice, and L. T. Reinders. *The World Health Report: A Safer Future: Global Public Health Security in the 21st Century*. World Health Organization, 2007.

[25] Z. Huang, Y. Liu, Y. Fang, and B. K. Horn. Video-based Fall Detection for Seniors with Human Pose Estimation. In *4th International Conference on Universal Village*, pages 1–4. IEEE, 2018.

[26] E. Jones, T. Oliphant, and P. Peterson. SciPy: Open Source Scientific Tools for Python, 2001–. `http://www.scipy.org`.

[27] O. O. Khin, Q. M. Ta, and C. C. Cheah. Development of a Wireless Sensor Network for Human Fall Detection. In *International Conference on Real-Time Computing and Robotics*, pages 273–278. IEEE, 2017.

[28] Y. Kong, J. Huang, S. Huang, Z. Wei, and S. Wang. Learning Spatiotemporal Representations for Human Fall Detection in Surveillance Video. *Journal of Visual Communication and Image Representation*, 59:215–230, 2019.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105, 2012.

[30] I. Kukharenko and V. Romanenko. Picking a Human Fall Detection Algorithm for Wrist–Worn Electronic Device. In *IEEE First Ukraine Conference on Electrical and Computer Engineering*, pages 275–277, 2017.

[31] V. S. Kumar, K. G. Acharya, B. Sandeep, T. Jayavignesh, and A. Chaturvedi. Wearable Sensor–Based Human Fall Detection Wireless System. In *Wireless Communication Networks and Internet of Things*, pages 217–234. Springer, 2018.

[32] B. Kwolek and M. Kepski. Human Fall Detection on Embedded Platform Using Depth Maps and Wireless Accelerometer. *Computer Methods and Programs in Biomedicine*, 117(3):489–501, 2014.

[33] B. Kwolek and M. Kepski. Improving Fall Detection by the Use of Depth Sensor and Accelerometer. *Neurocomputing*, 168:637–645, 2015.

[34] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient–Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[35] T. Lee and A. Mihailidis. An Intelligent Emergency Response System: Preliminary Development and Testing of Automated Fall Detection. *Journal of Telemedicine and Telecare*, 11(4):194–198, 2005.

[36] G. Leite, G. Silva, and H. Pedrini. Fall Detection in Video Sequences Based on a Three-Stream Convolutional Neural Network. In *18th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 191–195, Boca Raton-FL, USA, Dec. 2019.

[37] G. Leite, G. Silva, and H. Pedrini. Fall Detection. `https://github.com/guilhermevleite/fall_detection`, 2020. [Online; acessado 07-Jan-2020].

[38] H. Li, K. Mueller, and X. Chen. Beyond Saliency: Understanding Convolutional Neural Networks from Saliency Prediction on Layer-Wise Relevance Propagation. *Computer Research Repository*, 2017.

[39] X. Li, T. Pang, W. Liu, and T. Wang. Fall Detection for Elderly Person Care Using Convolutional Neural Networks. In *10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics*, pages 1–6, 2017.

[40] W.-N. Lie, A. T. Le, and G.-H. Lin. Human Fall-Down Event Detection Based on 2D Skeletons and Deep Learning Approach. In *International Workshop on Advanced Image Technology*, 2018.

[41] B.-S. Lin, J.-S. Su, H. Chen, and C. Y. Jan. A Fall Detection System based on Human Body Silhouette. In *9th International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 49–52. IEEE, 2013.

[42] N. Lu, Y. Wu, L. Feng, and J. Song. Deep Learning for Fall Detection: 3D-CNN Combined with LSTM on Video Kinematic Data. *IEEE Journal of Biomedical and Health Informatics*, 2018.

[43] B. D. Lucas, T. Kanade, et al. An Iterative Image Registration Technique with an Application to Stereo Vision. *International Joint Conference on Artificial Inteligence*, 1981.

[44] F. Luna-Perejon, J. Civit-Masot, I. Amaya-Rodriguez, L. Duran-Lopez, J. P. Dominguez-Morales, A. Civit-Balcells, and A. Linares-Barranco. An Automated Fall Detection System Using Recurrent Neural Networks. In *Conference on Artificial Intelligence in Medicine in Europe*, pages 36–41. Springer, 2019.

[45] M. M. Lusardi, S. Fritz, A. Middleton, L. Allison, M. Wingood, E. Phillips, et al. Determining Risk of Falls in Community Dwelling Older Adults: A Systematic Review and Meta-Analysis Using Posttest Probability. *Journal of Geriatric Physical Therapy*, 40(1):1, 2017.

[46] X. Ma, H. Wang, B. Xue, M. Zhou, B. Ji, and Y. Li. Depth-Based Human Fall Detection Via Shape Features and Improved Extreme Learning Machine. *Journal of Biomedical and Health Informatics*, 18(6):1915–1922, 2014.

[47] L. Meng, B. Zhao, B. Chang, G. Huang, W. Sun, F. Tung, and L. Sigal. Interpretable Spatio-Temporal Attention for Video Action Recognition. *arXiv preprint arXiv:1810.04511*, 2018.

[48] W. Min, H. Cui, H. Rao, Z. Li, and L. Yao. Detection of Human Falls on Furniture Using Scene Analysis Based on Deep Learning and Activity Characteristics. *IEEE Access*, 6:9324–9335, 2018.

[49] M. N. H. Mohd, Y. Nizam, S. Suhaila, and M. M. A. Jamil. An Optimized Low Computational Algorithm for Human Fall Detection from Depth Images Based on

Support Vector Machine Classification. In *IEEE International Conference on Signal and Image Processing Applications*, 2017.

[50] T. P. Moreira, D. Menotti, and H. Pedrini. First-Person Action Recognition Through Visual Rhythm Texture Description. In *International Conference on Acoustics, Speech and Signal Processing*, pages 2627–2631. IEEE, 2017.

[51] E. B. Nievas, O. D. Suarez, G. B. García, and R. Sukthankar. Violence Detection in Video Using Computer Vision Techniques. In *International Conference on Computer Analysis of Images and Patterns*, pages 332–339. Springer, 2011.

[52] Y. Nizam, M. N. H. Mohd, and M. M. A. Jamil. Human Fall Detection from Depth Images Using Position and Velocity of Subject. *Procedia Computer Science*, 105:131–137, 2017.

[53] A. Núñez-Marcos, G. Azkune, and I. Arganda-Carreras. Vision-based Fall Detection with Convolutional Neural Networks. *Wireless Communications and Mobile Computing*, 2017, 2017.

[54] T. E. Oliphant. *Guide to NumPy*. USA: CreateSpace Independent Publishing Platform, USA, 2nd edition, 2015.

[55] L. Panahi and V. Ghods. Human Fall Detection using Machine Vision Techniques on RGB–D Images. *Biomedical Signal Processing and Control*, 44:146–153, 2018.

[56] P. S. Sase and S. H. Bhandari. Human Fall Detection using Depth Videos. In *5th International Conference on Signal Processing and Integrated Networks*, pages 546–549. IEEE, 2018.

[57] C. Schuldt, I. Laptev, and B. Caputo. Recognizing Human Actions: A Local SVM Approach. In *17th International Conference on Pattern Recognition*, volume 3, pages 32–36. IEEE, 2004.

[58] K. Sehairi, F. Chouireb, and J. Meunier. Elderly Fall Detection System based on Multiple Shape Features and Motion Analysis. In *International Conference on Intelligent Systems and Computer Vision*, pages 1–8. IEEE, 2018.

[59] A. Shojaei-Hashemi, P. Nasiopoulos, J. J. Little, and M. T. Pourazad. Video–Based Human Fall Detection in Smart Homes Using Deep Learning. In *IEEE International Symposium on Circuits and Systems*, 2018.

[60] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *Computer Research Repository*, 2013.

[61] K. Simonyan and A. Zisserman. Two–Stream Convolutional Networks for Action Recognition in Videos. In *Advances in Neural Information Processing Systems 27*, 2014.

[62] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014.

[63] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: Removing Noise by Adding Noise. *arXiv preprint arXiv:1706.03825*, 2017.

[64] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A Dataset of 101 Human Actions Classes from Videos in the Wild. *arXiv preprint arXiv:1212.0402*, 2012.

[65] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic Attribution for Deep Networks. In *34th International Conference on Machine Learning*, volume 70, pages 3319–3328. JMLR.org, 2017.

[66] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al. Going Deeper with Convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[67] S. K. Tasoulis, G. I. Mallis, S. V. Georgakopoulos, A. G. Vrahatis, V. P. Plagianakos, and I. G. Maglogiannis. Deep Learning and Change Detection for Fall Recognition. In J. Macintyre, L. Iliadis, I. Maglogiannis, and C. Jayne, editors, *Engineering Applications of Neural Networks*, pages 262–273, Cham, 2019. Springer International Publishing.

[68] The Joint Commission. Fall Reduction Program – Definition of a Fall, 2001. `https://www.jointcommission.org/standards_information/jcfaqdetails.aspx?StandardsFAQId=1522&StandardsFAQChapterId=4&ProgramId=0&ChapterId=0&IsFeatured=False&IsNew=False&Keyword=`, acessado 10-set-2018.

[69] B. S. Torres and H. Pedrini. Detection of Complex Video Events Through Visual Rhythm. *The Visual Computer*, 34(2):145–165, 2018.

[70] U.S. Department of Veterans Affairs. Falls Policy Overview. `http://www.patientsafety.va.gov/docs/fallstoolkit14/05_falls_policy_overview_v5.docx`, acessado 10-set-2018.

[71] F. B. Valio, H. Pedrini, and N. J. Leite. Fast Rotation-Invariant Video Caption Detection Based on Visual Rhythm. In *Iberoamerican Congress on Pattern Recognition*, pages 157–164. Springer, 2011.

[72] M. Vallejo, C. V. Isaza, and J. D. Lopez. Artificial Neural Networks as an Alternative to Traditional Fall Detection Methods. In *35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2013.

[73] G. Van Rossum and F. L. Drake Jr. *Python Reference Manual*. Centrum voor Wiskunde en Informatica, Amsterndam, 1995.

[74] L. Wang, Y. Xiong, Z. Wang, and Y. Qiao. Towards Good Practices for very Deep Two-Stream Convnets. *arXiv preprint arXiv:1507.02159*, 2015.

[75] World Health Organization. Global Health and Aging, 2011.

[76] World Health Organization. Fact Sheet Falls, 2012. `http://www.who.int/en/news-room/fact-sheets/detail/falls`, acessado 10-set-2018.

[77] World Health Organization. World Report on Ageing and Health, 2015.

[78] T. Xu, Y. Zhou, and J. Zhu. New Advances and Challenges of Fall Detection Systems: A Survey. *Applied Sciences*, 8(3):418, 2018.

[79] M. Yu, S. M. Naqvi, and J. Chambers. A Robust Fall Detection System for the Elderly in a Smart Room. In *IEEE International Conference on Acoustics Speech and Signal Processing*, pages 1666–1669, 2010.

[80] N. Zerrouki, F. Harrou, Y. Sun, and A. Houacine. Vision-Based Human Action Classification Using Adaptive Boosting Algorithm. *IEEE Sensors Journal*, 18(12):5115–5121, 2018.

[81] N. Zerrouki and A. Houacine. Combined Curvelets and Hidden Markov Models for Human Fall Detection. *Multimedia Tools and Applications*, 77(5):6405–6424, 2018.

[82] Z. Zhang and V. Athitsos. Fall Detection by Zhong Zhang and Vassilis Athitsos. `http://vlm1.uta.edu/~zhangzhong/fall_detection/`. [Online; acessado 07-Jan-2020].

[83] S. Zhao, W. Li, W. Niu, R. Gravina, and G. Fortino. Recognition of Human Fall Events Based on Single Tri–Axial Gyroscope. In *IEEE 15th International Conference on Networking, Sensing and Control*, 2018.

[84] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. A Comprehensive Survey on Transfer Learning. *arXiv preprint arXiv:1911.02685*, 2019.

[85] Y. Zigel, D. Litvak, and I. Gannot. A Method for Automatic Fall Detection of Elderly People Using Floor Vibrations and Sound–Proof of Concept on Human Mimicking Doll Falls. *IEEE Transactions on Biomedical Engineering*, 56(12):2858–2867, 2009.

[86] Z. Zuo, B. Wei, F. Chao, Y. Qu, Y. Peng, and L. Yang. Enhanced Gradient-Based Local Feature Descriptors by Saliency Map for Egocentric Action Recognition. *Applied System Innovation*, 2(1):7, 2019.