



**UEPB**

**UNIVERSIDADE ESTADUAL DA PARAÍBA  
CAMPUS VII – GOVERNADOR ANTÔNIO MARIZ  
CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS – CCEA  
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**DAVID DANILO ALVES DE OLIVEIRA**

**VISÃO COMPUTACIONAL PARA DETECÇÃO DE QUEDAS EM PESSOAS**

**IDOSAS**

**PATOS - PB  
2019**

DAVID DANILO ALVES DE OLIVEIRA

**VISÃO COMPUTACIONAL PARA DETECÇÃO DE QUEDAS EM PESSOAS  
IDOSAS**

Trabalho de Conclusão de Curso  
apresentado como requisito parcial para a  
obtenção do título de Bacharelado em  
Ciência da computação da Universidade  
Estadual da Paraíba.

**Orientador:** Fábio Júnior Francisco da  
Silva (Esp.)

**PATOS - PB  
2019**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

O48v Oliveira, David Danilo Alves de.  
Visão computacional para detecção de quedas em pessoas idosas [manuscrito] / David Danilo Alves de Oliveira. - 2019.  
46 p. : il. colorido.  
Digitado.  
Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências Exatas e Sociais Aplicadas, 2019.  
"Orientação : Prof. Esp. Fábio Júnior Francisco da Silva, Coordenação do Curso de Computação - CCEA."  
1. Visão computacional. 2. Detecção de quedas. 3. Inteligência artificial. I. Título

21. ed. CDD 006.3

David Danilo Alves de Oliveira

**VISÃO COMPUTACIONAL PARA DETECÇÃO DE QUEDAS EM PESSOAS IDOSAS**

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Ciências da Computação da Universidade Estadual da Paraíba, em cumprimento à exigência para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 28/11/2019

**BANCA EXAMINADORA**

Fábio Junior Francisco da Silva  
Prof. Esp. Fábio Júnior F. da Silva  
(Orientador)

Wellington C. Araújo  
Prof. Dr. Wellington Candeia de Araújo  
(Examinadora)

Amanda Mayara Sobral Rodrigues  
Prof. Amanda Mayara S. Rodrigues  
(Examinadora)

## RESUMO

Nos tempos atuais, pesquisas indicam que idosos com 65 anos ou mais têm tendências maiores a sofrerem quedas e se lesionarem em decorrência delas, o que pode causar desde problemas psicológicos, como medo em realizar atividades diárias, e até levar a morte. Com a intenção de evitar maiores problemas resultantes da falta de atendimento em tempo hábil e melhorar a segurança dos mais velhos em suas casas, é proposto neste trabalho um algoritmo de detecção de quedas, implementado na linguagem Python e que utiliza métodos de visão computacional com base na biblioteca OpenCV, para enviar alertas por email que devem informar um cuidador ou familiar sobre o evento ocorrido. Os resultados mostram que 80% dos testes realizados foram bem-sucedidos e 20% apresentaram falso-positivos em um teste específico. O alerta de quedas teve sucesso em todos os testes, realizando um envio em uma média de 6 segundos logo após a detecção da queda.

**Palavras-Chave:** Visão Computacional. Processamento de Imagens. Detecção de quedas. OpenCV.

## **ABSTRACT**

In current times, research indicates that older people 65 and older have a greater tendency to fall and injure themselves, which can cause psychological problems such as fear of daily activities and even death. With the intention of avoiding greater problems resulting from the lack of timely care and improving the safety of older people in their homes, is proposed in this work a fall detection algorithm, implemented in Python language and which uses OpenCV library based computer vision methods, to send email alerts that should inform a caregiver or family member about the event that occurred. The results show that 80% of the tests performed were successful and 20% showed false positives in a specific test, still so yet a good effectiveness is demonstrated. The fall alert was successful in all tests, performing a send at an average of 6 seconds shortly after detection of the fall.

**Keywords:** Computer Vision. Image Processing. Fall Detection. OpenCV.

## LISTA DE ABREVIATURAS E SIGLAS

AGMM	Adaptive Gaussian mixture model
GMM	Modelo de Mistura Gaussiana
IA	Inteligência Artificial
MOG2	Uma variação do AGMM de Zivkovic (2004 & 2006)
OpenCV	Open Source Computer Vision Library
PDI	Processamento Digital de Imagens
SMTP	Simple Mail Transfer Protocol
Yagmail	Yet Another Gmail

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	<b>8</b>
1.1	Objetivo geral .....	8
1.1	Objetivos específicos .....	9
<b>2</b>	<b>REFERENCIAL TEÓRICO</b> .....	<b>10</b>
2.1	Pessoas idosas que moram sozinhas .....	19
2.2	Detecção de quedas .....	11
2.2.1	<i>Sensores vestíveis</i> .....	12
2.2.2	<i>Sensores de ambiente</i> .....	14
2.2.3	<i>Tecnologias baseadas em visão</i> .....	14
2.3	Inteligência Artificial .....	16
2.3.1	<i>Visão computacional</i> .....	16
2.3.1.1	<i>Aquisição de imagens</i> .....	18
2.3.1.2	<i>Processamento de imagens</i> .....	18
2.3.1.3	<i>Segmentação de imagens</i> .....	19
2.3.1.4	<i>Extração de características</i> .....	19
2.3.1.5	<i>Reconhecimento de padrões</i> .....	20
2.3.2	<i>Detecção de quedas com visão computacional</i> .....	21
2.3.2.1	<i>Métodos analíticos</i> .....	21
2.3.2.2	<i>Métodos de aprendizado de máquina</i> .....	21
2.4	Trabalhos relacionados .....	22
<b>3</b>	<b>METODOLOGIA</b> .....	<b>24</b>
3.1	Alterações na detecção de queda do algoritmo .....	25
3.1.1	<i>Detecção de queda na horizontal</i> .....	25
3.1.2	<i>Detecção de queda na vertical</i> .....	26
3.1.3	<i>Detecção de queda na diagonal</i> .....	27
3.2	Alterações no envio de alerta do algoritmo .....	27
<b>4</b>	<b>RESULTADOS E DISCUSSÕES</b> .....	<b>28</b>
4.1	Ambientes Simulados .....	28
4.2	Pseudocódigo .....	29
4.2.1	<i>Pseudocódigo do algoritmo da queda na horizontal</i> .....	29
4.2.2	<i>Pseudocódigo do algoritmo da queda na vertical</i> .....	29

4.2.3	<i>Pseudocódigo do algoritmo de alerta de quedas</i> .....	30
4.3	Validação do algoritmo de detecção de quedas .....	30
4.3.1	<i>Teste de permanecer em pé</i> .....	31
4.3.2	<i>Teste de sentar</i> .....	32
4.3.3	<i>Teste de queda com o corpo na horizontal</i> .....	33
4.3.4	<i>Teste de queda com o corpo na vertical</i> .....	34
4.3.5	<i>Teste de queda com o corpo na diagonal</i> .....	35
4.3.6	<i>Resultados</i> .....	35
4.4	Testes do envio de alertas .....	36
5	<b>CONSIDERAÇÕES FINAIS</b> .....	38
	<b>REFERÊNCIAS</b> .....	39
	<b>APÊNDICE A – MÉTODO DA DETECÇÃO DE QUEDAS NA HORIZONTAL</b> .....	42
	<b>APÊNDICE B – MÉTODO DA DETECÇÃO DE QUEDAS NA VERTICAL</b> .....	43
	<b>APÊNDICE C – MÉTODO DO ENVIO DE ALERTAS</b> .....	44
	<b>ANEXO A: TERMO DE LIVRE ESCLARECIMENTO</b> .....	45

## 1 INTRODUÇÃO

A queda é uma síndrome geriátrica bem conhecida e representa ameaça à saúde das pessoas idosas. De acordo com estudo publicado pela *World Health Organization* no ano de 2007, estima-se que entre 28% e 35% das pessoas com mais de 65 anos sofrem pelo menos uma queda a cada ano, esse número aumenta para 42% para pessoas com mais de 70 anos. A detecção precoce da queda permite resgatar os sujeitos e evitar o mau prognóstico, pois se a ajuda imediata não for fornecida, o resultado pode ser a morte.

Existem diversas tecnologias baseadas em sensores vestíveis, semelhantes a relógios e pulseiras inteligentes, que foram criadas com a capacidade de realizar a detecção de quedas. Entretanto, tais abordagens requerem que os idosos utilizem os dispositivos com sensores o tempo todo, porém alguns deles tendem a esquecer de usar tais acessórios. Outros métodos conhecidos consistem na utilização de sensores de ambiente, porém os mesmos são mais custosos e são capazes somente de analisar áreas específicas em que foram instalados.

Outra abordagem se concretiza pela utilização de sistemas capazes de processar imagens através de câmeras de vídeo, as quais estão cada vez mais presentes em residências devido aos crescentes índices de violência (CASTRO; PEDRO, 2010). Esse monitoramento não requer o uso contínuo de sensores vestíveis e são economicamente mais viáveis na detecção de quedas.

Todo esforço e estratégia dedicados ao auxílio do cuidado da saúde da pessoa é bem visto e apresenta aplicabilidade na realidade. A detecção de quedas por processamento de imagens adquiridas a partir de câmeras se torna viável ao passo que esses dispositivos estão bem disseminados e o custo não chega a ser alto, bem como não precisa da interferência direta no monitorado para manter o sistema funcionando. Nesse sentido esta pesquisa tem como objetivos:

### 1.1 Objetivo geral

Desenvolver, usando visão computacional, uma solução *home care* para alerta de quedas em pessoas idosas.

## 1.2 Objetivos específicos

- Realizar o levantamento do estado da arte;
- Desenvolver algoritmo para detecção de quedas utilizando visão computacional;
- Desenvolver algoritmo de envio de mensagens alertando sobre a ocorrência de quedas;
- Testar o algoritmo criado.

A solução desenvolvida nesta pesquisa utiliza câmeras de baixo custo para realizar a captação das imagens, e um computador de hardware básico (processador Intel Core i3-6100U, 4GB de memória RAM), o qual receberá os dados visuais para detectar uma queda. Além disso, é importante salientar que o sistema será voltado a pessoas que estejam sozinhas, pois no caso da presença de mais alguém durante a ocorrência da queda, esta poderá pedir ajuda, como também, serão evitados possíveis erros de detecção quando existir dois ou mais indivíduos no campo de visão da câmera.

## 2 REFERENCIAL TEÓRICO

### 2.1 Pessoas idosas que moram sozinhas

Segundo a *World Health Organization* (2007), as quedas representam mais de 50% das internações de idosos e cerca de 40% das mortalidades não naturais para esse segmento da população.

Lara et al. (2017), afirma que o tropeço é a principal causa de quedas em idosos com mais de 65 anos, representando aproximadamente 60% dos casos.

O impacto adverso mais grave das quedas na saúde da pessoa afetada é a fratura de quadril, que pode causar 37% de mortalidade um ano depois do acidente; 35% a 50% não conseguiram recuperar o estado ambulatorial da pré-fratura; 20% permaneceram em condição sem ambulatório e apenas metade de todos os pacientes com fratura de quadril retornaram às comunidades (LYONS, 1997).

Com a idade, surgem tanto problemas de memória, declínio do tempo de reação e de mobilidade física, como também, enfraquecimento da força muscular que complicam muito a realização de atividades cotidianas. Os afetados não conseguem administrar suas vidas como antes e isso traz vários problemas (TINETTI; WILLIAMS, 1998). Os idosos solitários estão associados a menor comprometimento social, pior estado nutricional e maior vulnerabilidade a várias doenças que foram classificadas como grupo de alto risco de morbidade e mortalidade (WORLD HEALTH ORGANIZATION, 1992 apud LEE, Wei-Ju et al., 2011).

De acordo com Ramos, Menezes e Meira (2011), os motivos que levam os idosos a morarem desacompanhados são: o falecimento de pessoas da família, divórcio, procura pela autonomia e carência de recursos monetários para amparar os familiares.

As quedas são particularmente perigosas para pessoas que moram sozinhas porque uma quantidade significativa de tempo pode passar antes de receber assistência. Aproximadamente um terço dos idosos (maiores de 65 anos) na Europa mora sozinho e espera-se que a população idosa aumente significativamente nos próximos vinte anos (RODRIGUES, 2012).

Os projetos de prevenção de quedas da Organização Mundial da Saúde indicaram que qualquer estratégia de prevenção ou intervenção contra quedas não

devem ignorar o efeito do status econômico, do engajamento social e de diferentes questões culturais (WORLD HEALTH ORGANIZATION, 2007).

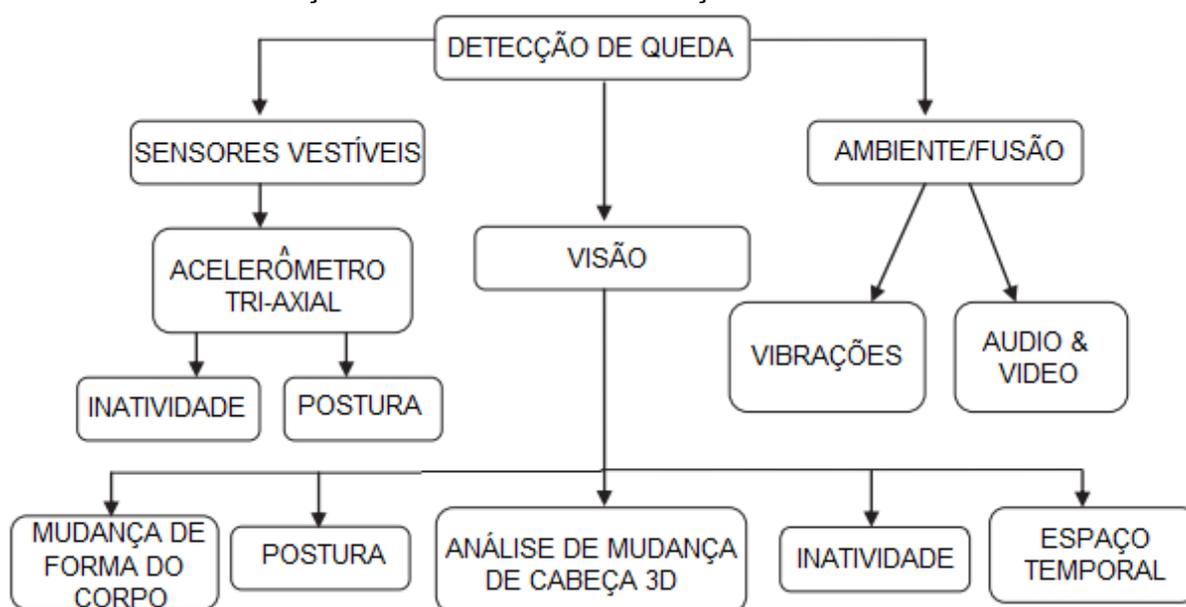
No estudo realizado com arranjo de 2601 pessoas com 65 anos ou mais, um terço moram sozinhos (860) e os outros dois terços (1741) vivem com outra pessoa. 17% dos que moram sozinhos sofreram múltiplas quedas nos últimos 12 meses e 36% limitam suas atividades devido ao medo de cair, o que é significativamente pior em relação àqueles que moram com outros, tendo 8,2% e 20,0% respectivamente (KHARICHA, 2007).

Para Hamill (2009), a vida solitária está fortemente associada a quedas, que podem se tornar uma condição de risco de morte. Isso aumenta a necessidade de programas de prevenção contra quedas para essas pessoas. O desenvolvimento de uma solução de notificação de alerta de quedas permite que mais idosos vivam nas comunidades com maior autonomia e privacidade além de minimizar o efeito das lesões decorrentes das quedas.

## 2.2 Detecção de quedas

O estado da arte para tecnologias baseadas em quedas, conforme mostrado na Figura 1, pode ser dividido em três categorias: sensores vestíveis, sensores de ambientes e tecnologias baseadas em visão computacional (MUBASHIR, 2013).

**FIGURA 1** - CLASSIFICAÇÃO DE MÉTODOS DE DETECÇÃO DE QUEDA.



Fonte: Adaptado de Mubashir, 2013.

### **2.2.1 Sensores vestíveis**

Abordagens baseadas em dispositivos vestíveis (*Wearable*) dependem de sensores instalados em diferentes regiões do corpo, os quais podem ser embutidos nas roupas, para detectar o movimento e localização da estrutura física do sujeito (MUBASHIR, 2013).

Ainda segundo o autor, a acelerometria é composto da medida de aceleração do corpo ou partes do corpo. É um dos métodos mais amplamente usados para medir atividades físicas e monitorar padrões. O objeto usado para fazer a medição é chamado de acelerômetro, que de acordo com Paula (2012) “[...] é um aparelho portátil de pequenas dimensões, leve e não invasivo, capaz de detectar as acelerações produzidas pelo corpo humano”.

O método que envolve os dispositivos vestíveis resumem-se à mecanismos elaborados com relação a utilização diária e usa sensores apropriados para adquirir dados fisiológicos ou mecânicos da pessoa monitorada de acordo com as exigências do objetivo almejado. Os acelerômetros são dispositivos vestíveis que se destacam por serem um dos mais utilizados (SILVA, 2013).

[...] a abordagem realizada por alguns sistemas baseados em sensores “vestíveis”, que se utilizam de diversos sensores acelerômetros e ou giroscópios [...] pode ocasionar uma limitação na execução das atividades diárias, devendo ser levada em consideração (SILVA, p. 2).

Os dispositivos vestíveis têm vantagens e desvantagens. A maior vantagem é a eficiência de custo, como também, a fácil instalação e configuração, fazendo com que seja relativamente simples a sua operação. As desvantagens incluem intrusão e relações relativas fixas com o objeto, logo o dispositivo pode ser facilmente desconectado. Tais desvantagens tornam os dispositivos vestíveis uma opção desfavorável para os idosos (MUBASHIR, 2013).

**FIGURA 2 - EXEMPLO DE DISPOSITIVOS VESTÍVEIS USADOS EM CONTATO DIRETO COM O CORPO.**



Fonte: Adaptado de Aliverti, 2017

A Figura 2 apresenta exemplos de dispositivos vestíveis que são utilizados em contato direto com o usuário por baixo da roupa.

**FIGURA 3 - EXEMPLO DE DISPOSITIVOS VESTÍVEIS USADOS SOBRE A ROUPA.**



Fonte: Adaptado de Aliverti, 2017.

A Figura 3 apresenta outros exemplos de dispositivos vestíveis que são utilizados com ou por cima da roupa do usuário.

### 2.2.2 Sensores de ambiente

Os métodos associados com os sensores de ambiente demandam a instalação de vários dispositivos nos locais a serem monitorados os quais podem ser desde uma área controlada para testes (laboratório) como também a habitação do sujeito analisado (SILVA, 2013).

Alguns exemplos de sensores ambientes utilizados para este fim, de acordo com o Quadro 1, são: placas de pressão, sensores infravermelhos, interruptores magnéticos, microfones, entre outros.

**QUADRO 1 – SENSORES USADOS EM AMBIENTES INTELIGENTES**

Sensor	Medição	Formato dos Dados
PIR <sup>1</sup>	Movimento	Categórico
Infravermelho ativo	Movimento/Identificação	Categórico
RFID <sup>2</sup>	Informação do Objeto	Categórico
Pressão	Pressão no tapete, cadeira, etc.	Numérico
Ladrilho Inteligente	Pressão no chão	Numérico
Interruptores magnéticos	Porta/Abertura do Armário/Fechamento	Categórico
Ultrassônico	Movimento	Numérico
Câmera	Atividade	Imagem
Microfone	Atividade	Som

<sup>1</sup> Sensor de Movimento Infravermelho Passivo

<sup>2</sup> Identificação por Radio Frequência

**Fonte:** Adaptado de Rashidi; Mihailidis, 2013.

O uso de sensores de ambiente possuem custo razoavelmente elevado, pois é necessário realizar a instalação de vários dispositivos no local a ser monitorado, como também, podem haver dificuldades em diferenciar o indivíduo de interesse aos demais presentes no mesmo local (GODFREY et al., 2008).

### 2.2.3 Tecnologias baseadas em visão computacional

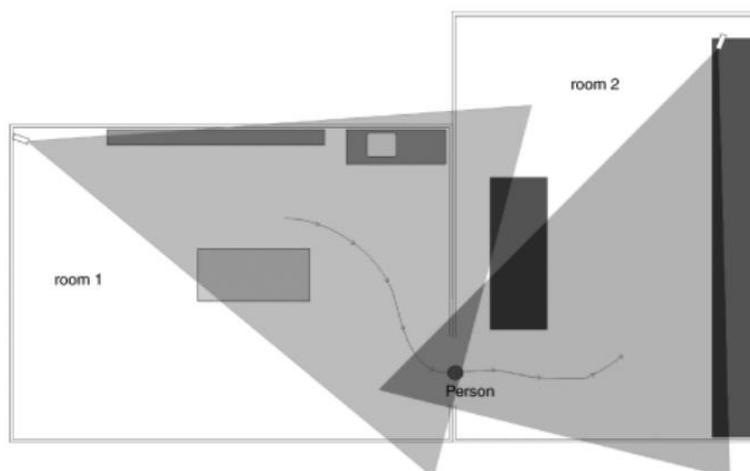
Mubashir (2013) afirma que a maior parte da ênfase no contexto da vigilância por visão computacional é dedicada as abordagens focadas na execução em tempo real do algoritmo usando plataformas de computação padrão e câmeras de baixo custo. Tais abordagens podem ser classificadas em: mudança de corpo e forma, detecção de postura, inatividade, mudança espaço-temporal e da cabeça em 3D.

As tecnologias baseadas em visão computacional estão associadas ao uso de câmeras, geralmente em ambiente controlado, que monitoram as ações do indivíduo continuamente. As desvantagens estão relacionadas à invasão de privacidade do sujeito e as dificuldades relacionadas ao processamento de imagens, as quais são afetadas pelas variações de luminosidade, mudanças no ambiente, ou múltiplas pessoas nas cenas capturadas (KHAN, 2011).

Entre as vantagens de realizar a detecção de quedas utilizando tecnologias baseadas em visão computacional, tem-se que as câmeras, as quais estão cada vez mais presentes em sistemas internos de segurança, são sensores menos intrusivos que podem ser utilizados para identificar vários eventos ao mesmo tempo sem precisar da utilização de quaisquer outros sensores, isso acontece pois os dados captados são semanticamente mais ricos e precisos que os fornecidos por demais dispositivos padrões (CUCCHIARA; PRATI; VEZZANI, 2007).

As desvantagens estão relacionadas ao fato de que as câmeras possuem um campo de visão limitado por oclusões e, no caso do monitoramento em ambientes internos, a área de cobertura fica restringida a uma área específica, sendo necessário realizar a instalação de mais câmeras e coordena-las para obter uma cobertura visual maior como demonstrado na Figura 4. Além disso, é necessário realizar o processamento dos dados, os quais requerem maior poder computacional (CUCCHIARA; PRATI; VEZZANI, 2007).

**FIGURA 4 – CAMPO DE VISÃO DE CÂMERAS COBRINDO ÁREAS ESPECÍFICAS.**



**Fonte:** Cucchiara; Prati; Vezzani, 2007

Cucchiara, Prati e Vezzani, (2007) também falam que utilizando a visão computacional pode-se detectar a queda de uma pessoa analisando mudanças

bruscas na postura. Além disso, é importante pensar na estratégia a ser usada quando uma queda for detectada, podendo ser transmitido o vídeo ao vivo ao operador ou usuário remoto que receberá uma notificação de alarme pelo celular, ajudando-o a entender o que aconteceu. Neste caso é importante considerar os recursos limitados do dispositivo móvel avaliando os recursos de exibição, potência computacional e largura de banda disponível.

## **2.3 Inteligência Artificial**

Inteligência Artificial (IA) é um campo de pesquisa moderno que envolve áreas da ciência computacional e engenharia. Ela surgiu em meados de 1956, posterior a Segunda Guerra Mundial e busca realizar a compreensão de agentes inteligentes e tentar produzi-los. Uma IA pode ser considerada como um agente inteligente que obtém dados de um determinado ambiente e através de análises, realiza a compreensão do mesmo exercendo determinadas atividades sobre ele (NORVIG; RUSSELL, 2014).

As disciplinas que abordam a maior parte da área de Inteligência Artificial são o processamento de linguagem natural, que realiza a comunicação do sistema com uma pessoa utilizando apenas o idioma natural; a representação de conhecimento, a qual realiza o procedimento de armazenamento dos dados conhecidos e daquilo que é percebido; raciocínio automatizado, que faz uso das informações memorizadas para deduzir e responder perguntas e argumentos; aprendizado de máquina, utilizada para adequar o sistema a um novo cenário e realizar o reconhecimento de padrões sobre o mesmo; visão computacional, que realiza um entendimento de alto nível sobre dados visuais como imagens ou vídeos através de processamento e análise dos mesmos; e robótica, relacionada à construção, operação e uso de robôs, bem como sistemas de computacionais para os mesmos (NORVIG; RUSSELL, 2014).

### **2.3.1 Visão computacional**

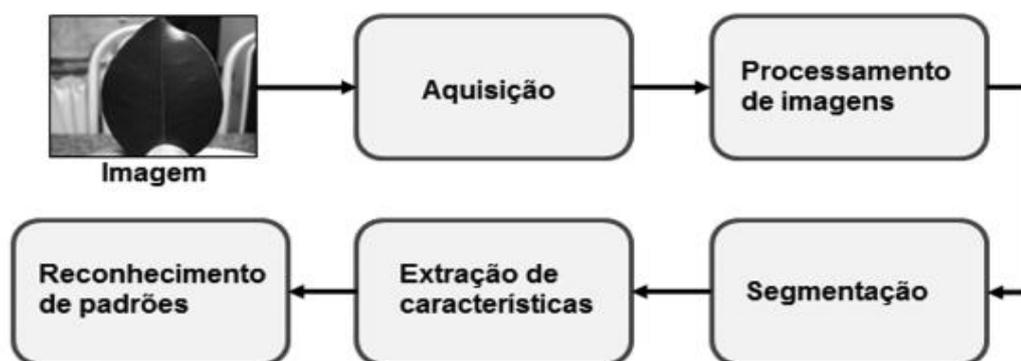
Visão computacional é a ciência que realiza a pesquisa e desenvolvimento de tecnologias as quais possibilitam que máquinas observem e extraiam características de algum objeto ou ambiente por meio de imagens capturadas de diferentes

aparelhos e equipamentos. As informações capturadas possibilitam realizar o reconhecimento, manipulação e processamento dos dados mediante os objetos que formam a imagem (BALLARD; BROWN, 1982).

Podemos definir visão computacional como a área de estudo que tenta repassar para máquinas a incrível capacidade da visão. Quando falamos de visão, não estamos nos referindo apenas ao ato de captar imagens. Apesar de essa capacidade ser impressionante [...], ela é apenas o início de um processo muito mais vasto e rico. A visão consiste em captar imagens, melhorá-las (por exemplo, com retirada de ruídos, aumento de contraste etc.), separar as regiões ou objetos de interesse de uma cena, extrair várias informações dependendo da imagem analisada, como, por exemplo, forma, cor e textura, e, finalmente, relacionar as imagens com outras vistas previamente (BACKES; JUNIOR, 2019, p. 1).

Um sistema de visão computacional geralmente apresenta um fluxo em comum, o qual é constituído de várias fases demonstradas na Figura 5. São elas: Aquisição, processamento, segmentação, extração de características e reconhecimento de padrões.

**FIGURA 5 – ESQUEMA DE UM SISTEMA DE VISÃO COMPUTACIONAL.**



Fonte: Backes e Junior, 2019.

É importante enfatizar que não há uma descrição única para as fases de um sistema de visão computacional. Por exemplo, alguns autores chamam de “pré-processamento” a etapa que denominamos “processamento de imagens”, e chamam de “processamento de imagens” todas as cinco fases apresentadas. O número de fases também varia de acordo com o ponto de vista dos autores. Além disso, algumas fases podem ser suprimidas dependendo do problema. Por exemplo, uma imagem pode seguir direto para a fase de extração de características sem passar pela fase de segmentação. Isso pode ocorrer tanto pela segmentação ter sido efetuada manualmente (às vezes esse procedimento é necessário para problemas mais difíceis) quanto porque a imagem original já representa toda a região de interesse para análise (BACKES; JUNIOR, 2019, p. 3).

### 2.3.1.1 Aquisição de imagens

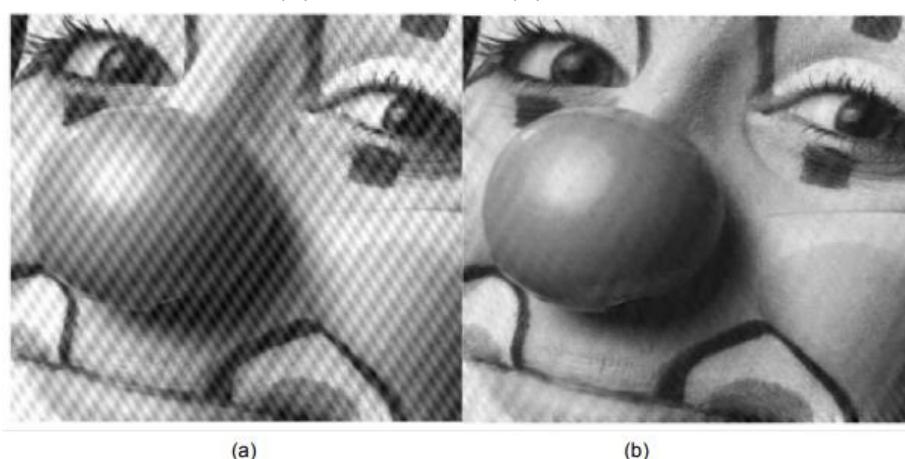
Esta é a primeira etapa de um sistema de Processamento Digital de Imagens (PDI) e todas as etapas seguintes dependem desta. Aqui, os dispositivos e sensores são utilizados para realizar a captação das imagens, sejam elas bidimensional, tridimensional ou um vídeo, gerando uma imagem digitalizada como resultado. Alguns equipamentos que realizam essa tarefa são: scanners, filmadoras, webcams etc. (BURGER; BURGE, 2007, BACKES; JUNIOR, 2019).

### 2.3.1.2 Processamento de imagens

A imagem adquirida pode apresentar imperfeições, que prejudicam a interpretação ou o reconhecimento de objetos contidos na imagem (MARENGONI; STRINGHINI, 2009).

Esta fase é responsável por aplicar alguns recursos de pré-processamento (para reduzir ou remover ruídos, como mostrado na Figura 6, destacar bordas e formas geométricas, etc.) na imagem para realçá-la, proporcionando assim a fácil aquisição de suas informações pelo modelo de visão computacional (BACKES; JUNIOR, 2019).

**FIGURA 6 – IMAGEM (A) COM RUÍDO E (B) APÓS FILTRAGEM.**



**Fonte:** Marengoni; Stringhini, 2009.

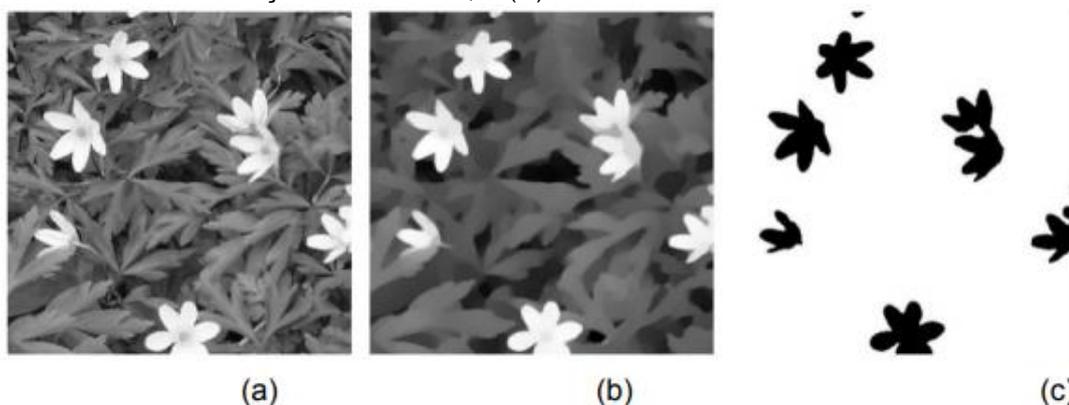
É importante salientar que essa fase compreende tanto o que usualmente se denomina “pré-processamento”, como rotação da imagem, equalização de histograma etc., quanto processamentos mais complexos, como, por exemplo, filtragens e aplicação de operadores morfológicos (BACKES; JUNIOR, 2019, p. 3).

Após o processamento de imagens, em seguida é realizada a etapa de segmentação.

### 2.3.1.3 Segmentação de imagens

Nesta etapa é feita a fragmentação da imagem original em regiões significativas (frontais, de fundo ou de um objeto específico) (SOLEM, 2012).

**Figura 7** – EXEMPLO DE UM PROCESSO DE SEGMENTAÇÃO: (A) UMA IMAGEM ORIGINAL; (B) IMAGEM COM APLICAÇÃO DE FILTRO; E (C) IMAGEM SEGMENTADA.



Fonte: Solem, 2012.

Logo após são segmentados os itens relevantes em uma ou mais imagens, com a intenção de facilitar a extração de características do mesmo, como demonstrado na Figura 7 (BACKES; JUNIOR, 2016).

### 2.3.1.4 Extração de características

Logo após o processo de segmentação, que realizou a detecção dos objetos na imagem, são extraídas várias propriedades ou características com base nestes objetos (Figura 8), como histograma, tamanho, área, perímetro, textura, contornos, etc. (DE QUEIROZ; GOMES, 2006).

[...] responsável por obter um conjunto de características do objeto de interesse. Em outras palavras, essa fase é responsável por encontrar uma codificação numérica que represente determinada imagem, como uma espécie de “impressão digital” (analogia imperfeita) que permita identificá-lo (BACKES; JUNIOR, 2019, p. 3).

**FIGURA 8 – EXTRAÇÃO DE CARACTERÍSTICA (ÁREA) DE UM MORANGO.**



Fonte: Ribeiro, 2015.

Após isso, é realizado o reconhecimento de padrões, se o mesmo for necessário para o sistema almejado.

#### *2.3.1.5 Reconhecimento de padrões*

Nesta última etapa, realiza-se o processamento de alto nível com a finalidade de identificar o item segmentado por meio de suas propriedades e classificar ou agrupá-lo de acordo com seus conjuntos de características (BACKES; JUNIOR, 2019).

O objetivo do reconhecimento de padrões é identificar objetos na cena a partir de um conjunto de medições. Cada objeto é um padrão e os valores medidos são as características desse padrão. Um conjunto de objetos similares, com uma ou mais características semelhantes, é considerado como pertencente à mesma classe de padrões. Há diversos tipos de características, cada uma das quais é obtida a partir de uma técnica específica. [...] Outras características relevantes para um objeto 2D ou 3D são a área, volume, perímetro, superfície, dentre outras, as quais podem ser medidas a partir da contagem de pixels. (DE QUEIROZ; GOMES, 2006, p. 26).

Segundo Rezende (2003), um exemplo de reconhecimento de padrões que pode ser utilizado é o *machine learning*.

### **2.3.2 Detecção de quedas com visão computacional**

De acordo com Noury et al. (2007), os métodos para a detecção de quedas são baseados principalmente em modelos analíticos e alguns usam as técnicas de aprendizado de máquina.

#### **2.3.2.1 Métodos analíticos**

O processamento de imagem de sinais de vídeo também pode ser usado para detectar uma queda identificando a postura deitada usando análise de cena ou detectando movimentos bruscos usando análise vetorial. Este método consiste tipicamente em subtrair imagens sucessivas para manter apenas as variações, que são então classificadas de acordo com sua direção e/ou sua amplitude (NOURY et al., 2007).

O uso destas técnicas não representam problemas em ambientes de testes controlados, como um laboratório, porém, nos demais cenários não controlados é necessário que elas sejam ajustadas quando não se pode controlar a iluminação e o enquadramento, pois o sujeito monitorado precisa ser visto claramente e necessita estar no campo de visão do sensor de vídeo. Além disso, à medida que o sujeito se movimenta num espaço de 3 dimensões, é também necessário recorrer a técnicas mais complexas, ou seja, a utilização de 2 câmaras (“*stereovision*”) (NOURY et al., 2007).

#### **2.3.2.2 Métodos de aprendizado de máquina**

De acordo com Rezende (2003, p. 89), aprendizado de máquina é:

[...] uma área de IA cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado bem como a construção de sistemas capazes de adquirir conhecimento de forma automática. Um sistema de aprendizado é um programa de computador que toma decisões baseado em experiências acumuladas por meio da solução bem-sucedida de problemas anteriores. Os diversos sistemas de Aprendizado de Máquina possuem características particulares e comuns que possibilitam sua classificação quanto à linguagem de descrição, modo, paradigma e forma de aprendizado utilizado.

Com o aprendizado de máquina é possível realizar uma abordagem "intuitiva" para o desenvolvimento de sistemas de detecção de queda, começando pela observação (um período de treinamento) e depois pela classificação, o qual necessita que seja estabelecido critérios suficientemente significativos e independentes (NOURY et al., 2007).

Se for utilizado um treinamento supervisionado, o qual de acordo com Norvig e Russell (2014), "[...] o agente observa alguns exemplos de pares de entrada e saída e aprende uma função que faz o mapeamento da entrada para a saída", pode-se treinar uma rede neural que será então usada para classificar automaticamente futuras situações. Além disso, apenas as situações encontradas durante o treinamento podem ser classificadas (NOURY et al., 2007).

Se o treinamento é "não supervisionado", que segundo Norvig; Russell, (2014) "[...] o agente aprende padrões na entrada, embora não seja fornecido nenhum *feedback* explícito", uma classe "Queda" pode de ser isolada se o período de treinamento for longo, até mesmo infinito se o evento de queda for raro ou não acontecer. Além disso, o primeiro evento da queda pode ser perdido antes de sua primeira ocorrência, pois a classe "Queda" ainda é desconhecida (NOURY et al., 2007).

## 2.4 Trabalhos relacionados

Na pesquisa intitulada "Reconhecendo quedas de silhuetas" é realizado o reconhecimento de quedas segmentando primeiro o indivíduo do segundo plano, resultando em uma silhueta em mapa binário (imagem em preto e branco), indicando apenas a posição do corpo do indivíduo em uma imagem. Depois que as silhuetas são obtidas, é feito a extração das características e são treinados modelos ocultos de Markov para reconhecer desempenhos futuros das atividades conhecidas (ANDERSON et al., 2006).

No trabalho de Hsu, Perng e Liu (2015), intitulado "Desenvolvimento de um sistema de detecção de quedas de pedestres baseado em visão com rede neural de retro propagação", os autores realizam a obtenção de sequências de imagens de um sistema de câmeras internas. De início, as imagens são usadas para construir um modelo de fundo, usando o modelo de mistura Gaussiana (GMM) com a extração de imagens de primeiro plano obtidas por meio de subtração. Depois, operações

morfológicas são utilizadas para reparar danos à imagem e a rotulagem dos componentes conectados é usada para eliminar o ruído. A partir dos objetos em primeiro plano, a proporção da caixa delimitadora, a orientação da elipse, e a velocidade vertical do ponto central são extraídas para uso como recursos de entrada em um algoritmo de aprendizado. A detecção de queda é baseada nos resultados de classificação do algoritmo.

Na pesquisa publicada sob o título “Sistema de detecção de queda usando o sensor infravermelho do Kinect”. O sistema é executado em tempo real sendo capaz de detectar quedas durante a caminhada com precisão, desconsiderando quaisquer atividades falso-positivas (como deitar no chão). Os cálculos de velocidade e inatividade são realizados para decidir se uma queda ocorreu. Assim, é medida a velocidade com base na contração ou expansão da largura, altura e profundidade da caixa delimitadora 3D, a qual é utilizada para que não seja requerido o pré-conhecimento da cena, pois o conjunto de ações detectadas é adequado para concluir o processo de detecção de queda (MASTORAKIS; MAKRIS, 2014).

No algoritmo desenvolvido por Salmi (2016), “Melhorando a segurança de pacientes em atendimento domiciliar com uma solução de visão computacional de baixo custo”. O pré-processamento converte cada quadro adquirido em uma imagem na escala de cinza. A segmentação dá suporte ao MOG2 (uma variação do AGMM) e a abordagem dinâmica, as quais servem para extrair somente a parte da imagem referente ao corpo do sujeito do plano de fundo. Logo após, a extração de características ocorre para detectar os contornos externos (caixa delimitadora) das imagens segmentadas captando a posição  $(x,y)$ , a altura  $(h)$  e a largura  $(w)$ . E por fim, acontece o reconhecimento de padrões através de métodos analíticos.

### 3 METODOLOGIA

O levantamento do estado da arte foi elaborado através de pesquisas nos sites científicos “Google Acadêmico”, “Periódicos CAPES” e “IEEE Xplore” utilizando nas consultas as palavras-chave: “Detecção de quedas”; “Detecção de pessoas em imagens”; “Visão computacional”; “Pessoas idosas”. Em inglês: “*Machine learning*”; “*Computer vision*”; “*Fall detection*”; “*Elderly people*”.

O algoritmo criado nesta pesquisa é uma modificação do sistema de monitoramento automático de quedas, desenvolvido por Salmi (2016), cujo código foi desenvolvido e testado com o Python na versão 2.7.11, OpenCV versão 2.4.13 e um *Raspberry Pi 3 Modelo B* com uma câmera genérica conectada a ele. Além disso, o código está licenciado sob a GPLv3, a qual garante liberdade de usar, modificar e compartilhar o software (SMITH, 2018).

A modificação e testes do algoritmo responsável por realizar a detecção de quedas ocorreu em um computador com processador Intel Core i3-6100U, 4GB de memória RAM, com sistema Linux Ubuntu 18.04 fazendo uso da linguagem de programação Python 2.7.15 com o uso de módulos da biblioteca multiplataforma livre OpenCV 4.1.1, a qual requer que algumas correções sejam realizadas na sintaxe do algoritmo de Salmi (2016) devido à incompatibilidade com a versão a qual foi produzido.

O ambiente virtual simulado, que foi usado nos testes do algoritmo, foi criado na Game Engine Unity 2018.3.0f2, na qual construiu-se um piso e paredes de forma a simular um cômodo de uma casa. Além disso, o pacote *Basic Motions Pack*, adquirido gratuitamente na *Unity Asset Store*, também foi utilizado para fazer a representação de um personagem que estará presente na cena. A aquisição das imagens aconteceu pela captura de tela, realizada pelo programa *SimpleScreenRecorder*, que gerou um vídeo no formato mp4.

Na simulação real, os testes ocorreram com um indivíduo que realizou ações de queda em local coberto por câmera. A simulação aconteceu de maneira que não causasse danos físicos ou psíquicos ao participante da simulação e o mesmo assinou termo de livre e esclarecimento. Os eventos analisados foram: ficar parado em pé, ficar sentado em uma cadeira e simular quedas.

### 3.1 Alterações na detecção de queda do algoritmo

A detecção de queda do algoritmo original desenvolvido por Salmi (2016) acontece quando um corpo detectado permanece imóvel na cena por um breve período de tempo, não havendo assim alteração nos valores da posição da caixa delimitadora. Porém, este método está sujeito a muitas respostas falso-positivas, pois desde que uma pessoa permaneça em pé e imóvel (ou quase) na cena a ser observada pela câmera, uma queda ainda é detectada pelo algoritmo, o qual aciona o alarme do *webservice* RESTful sem a devida necessidade.

Esta pesquisa adicionou métodos analíticos adicionais para reduzir a quantidade de falso-positivos gerados pelo sistema, tornando o algoritmo mais inteligente. Para tanto, a queda só deverá ser detectada pelo sistema se a mesma se encaixar em uma das três possíveis alternativas: queda na horizontal, queda na vertical ou queda na diagonal.

#### 3.1.1 Detecção de queda na horizontal

A queda horizontal é retratada quando um corpo, que antes estava de pé, cai paralelo ao horizonte observado pela câmera. Na Figura 9, um corpo caído se encontra deitado na posição Leste-Oeste, supondo que ele estivesse sendo monitorado pelo lado Norte ou Sul, então é dito que houve uma queda na horizontal.

**FIGURA 9 – EXEMPLO DE QUEDA NA HORIZONTAL.**



Fonte: Autoria própria, 2019.

Para detectar a queda na horizontal, é tomado como verdadeiro a proposta que todo corpo observado que possuir uma largura ( $w$ ) maior que a sua altura ( $h$ ) deve estar horizontalmente deitado. Quando isto acontece e o corpo não está em movimento, então a queda é detectada.

### 3.1.2 Detecção de queda na vertical

A queda vertical acontece quando um corpo, que antes estava de pé, cai perpendicular ao horizonte observado pela câmera. Na Figura 10, um corpo caído se encontra deitado na posição Sul-Norte, supondo que ele estivesse sendo monitorado pelo lado Norte ou Sul, então é dito que houve uma queda na Vertical.

**FIGURA 10** – EXEMPLO DE QUEDA NA VERTICAL.



**Fonte:** Autoria própria, 2019.

Uma queda na vertical pode ser detectada assumindo-se como verdadeiro a proposta que todo corpo observado, em um dado momento, possui uma altura ( $h$ ) e uma largura ( $w$ ) que podem aumentar ou diminuir em uma determinada proporção a medida que o corpo se aproxima ou se afasta da câmera. Caso haja variação repentina na altura do corpo, adquirindo um valor menor que no momento anterior, sem uma presença significativa de uma alteração na largura, então o corpo deve ter sofrido uma queda vertical.

### 3.1.3 Detecção de queda na diagonal

A queda na diagonal acontece quando um corpo, que antes estava de pé, cai na transversal em relação ao horizonte observado pela câmera. Na Figura 11, há dois corpos caídos, o da esquerda se encontra deitado na posição Sudoeste-Nordeste e o da direita no Sudeste-Noroeste, supondo que eles estivessem sendo monitorados pelo lado de um dos 4 pontos cardeais, então é dito que houve uma queda na diagonal.

FIGURA 11 – EXEMPLO DE UM CORPO NA DIAGONAL.



Fonte: Autoria própria, 2019.

## 3.2 Alterações no envio de alerta do algoritmo

No sistema de Salmi (2016), quando a queda é detectada, o algoritmo dispara um alerta para um *backend* via API RESTfull (*webservice*), o qual notifica algum responsável, seja cuidador profissional ou a família, sobre o problema sem enviar qualquer tipo de imagem para a devida comprovação, tornado assim necessário que uma ligação para a pessoa monitorada seja realizada para averiguar se o mesmo está bem.

O algoritmo de alerta de quedas desta pesquisa usa a biblioteca Python *Yagmail*, um cliente *Gmail* que utiliza o protocolo SMTP para o envio de *e-mails*, para enviar mensagem de texto com uma imagem do ocorrido a um destinatário (responsável) cadastrado, para que possa assim ser analisada e as medidas cabíveis, dependendo da situação, sejam tomadas (KOOTEN, 2019).

## 4 RESULTADOS E DISCUSSÕES

Este capítulo aborda os resultados sobre o desenvolvimento dos ambientes de simulação dos métodos analíticos implementados com base no algoritmo de Salmi (2016), do algoritmo de alerta de quedas e sobre os testes realizados em ambos.

### 4.1 Ambientes Simulados

Os testes de avaliação do algoritmo de detecção de quedas aconteceram em local iluminado e sem presença de objetos que gerem imagens refletidas ou oclusão no personagem de testes e foram realizados em dois ambientes distintos, o real e o virtual.

No ambiente virtual, desenvolvido com o *Unity*, representado pela Figura 12, foi construído o piso na cor verde e a parede de fundo acinzentada, simulando um cômodo que foi monitorado pelo sistema. O personagem amarelo, adquirido do pacote *Basic Motions Pack*, foi utilizado para realizar as simulações de quedas.

**FIGURA 12** – AMBIENTE VIRTUAL SIMULADO COM PERSONAGEM DE TESTES.



Fonte: Autoria própria, 2019.

Os testes em ambiente real simulado foram realizados na sala de uma casa, removendo objetos e móveis que gerassem oclusão da pessoa a ser monitorada. O local foi escolhido por possuir espaço suficiente para realizar a gravação das simulações de quedas.

## 4.2 Pseudocódigo

Nesta seção será explicado o funcionamento dos métodos analíticos, responsáveis por fazer o reconhecimento de queda. Os pseudocódigos foram inspirados na obra *Algoritmos: Teoria e prática* de Cormen (2002).

### 4.2.1 Pseudocódigo do algoritmo da queda na horizontal

No algoritmo, para detectar a queda horizontal, é comparada a cada execução do código a largura em relação à altura, neste caso, se a largura da caixa delimitadora ficar maior que a altura da mesma então a variável (*QuedaHorizontal*) recebe o valor verdadeiro (*true*), a exemplo do Apêndice A.

Assim, quando isto acontecer e a posição da caixa delimitadora não mudar (estado imóvel), então o algoritmo determina que houve uma queda e chama a função alerta de quedas.

### 4.2.2 Pseudocódigo do algoritmo da queda na vertical

No algoritmo representado no Apêndice B, para detectar a queda vertical:

1. A variável que armazena o valor da altura da caixa delimitadora no segundo anterior recebe o valor da altura atual da mesma assim que uma pessoa entra no campo de visão da câmera.
2. É verificado se houve uma mudança na variável que armazena o número inteiro dos segundos do sistema, comparando com o valor atual.
  - a. Se o valor da variável for diferente então haverá outra verificação na qual a altura atual da caixa delimitadora é somada com 20 (limite aceitável) e verificado se é menor que a altura da caixa no segundo anterior, como também, a largura atual é somada com 5 (limite aceitável) e verificado se é maior ou igual a largura no segundo anterior.
    - i. Se as preposições forem verdadeiras, então a variável (*QuedaVertical*) recebe o valor verdadeiro (*true*);

ii. Se pelo menos uma das preposições for falsa, nada acontece.

Após isso, as variáveis que armazenam a altura e a largura da caixa delimitadora do segundo anterior recebem respectivamente a altura e a largura atual da mesma.

b. Caso a variável possua o mesmo valor dos segundos atual do sistema, então nada acontece.

3. A variável que recebe o valor inteiro dos segundos do sistema recebe o valor atual do mesmo.

Se a variável (*QuedaVertical*) possuir valor verdadeiro (*true*) e a caixa delimitadora se encontra imóvel, então o algoritmo determina que houve uma queda e chama a função do alerta de quedas.

#### **4.2.3 Pseudocódigo do algoritmo de alerta de quedas**

O algoritmo de alerta de quedas, representado em pseudocódigo pelo Apêndice C, precisa receber algumas informações para conseguir realizar o envio do *email* com o Yagmail. É necessário informar o endereço de *email* do destinatário, como também, passar o endereço e senha do *email* remetente, os quais serão utilizados para iniciar uma conexão SMTP com o servidor de envio. Além disso, informações textuais do título e corpo do *email* podem ser personalizados, junto com o caminho da imagem que será passada em anexo.

#### **4.3 Validação do algoritmo de detecção de quedas**

A validação do algoritmo de detecção de quedas foi feita através da realização de 5 testes nos dois ambientes de simulação, o real e o virtual. Os testes são: permanecer em pé, sentar, simular queda horizontal, vertical e na diagonal.

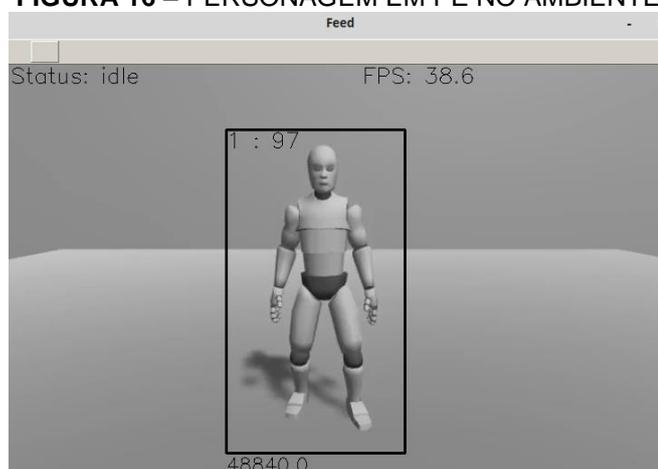
Nas Figuras a serem apresentadas (16 à 25) existem três valores numéricos na caixa delimitadora. O primeiro valor é o ID (identidade) da pessoa rastreada. O segundo valor é o contador de alarme que será iniciado se a pessoa não estiver se movendo acima de um limite pré-definido no algoritmo, sendo redefinido quando houver movimento. O terceiro valor é o tamanho da caixa delimitadora. Na parte de

cima pode ser observado o FPS do vídeo e a classificação em “Idle” (ocioso) quando houver movimento da caixa delimitadora e não houver queda ou “Alarm, not moving” (Alarme, sem movimento) quando uma queda for detectada e não haver movimentação.

#### 4.3.1 Teste de permanecer em pé

Neste teste, o personagem entrou na área monitorada pela câmera e ficou parado em pé durante algum tempo.

**FIGURA 16 – PERSONAGEM EM PÉ NO AMBIENTE VIRTUAL.**



Fonte: Autoria própria, 2019.

No ambiente simulado virtual, apresentado na Figura 16, o personagem foi detectado corretamente, não havendo classificação de queda.

**FIGURA 17 – PERSONAGEM EM PÉ NO AMBIENTE REAL.**



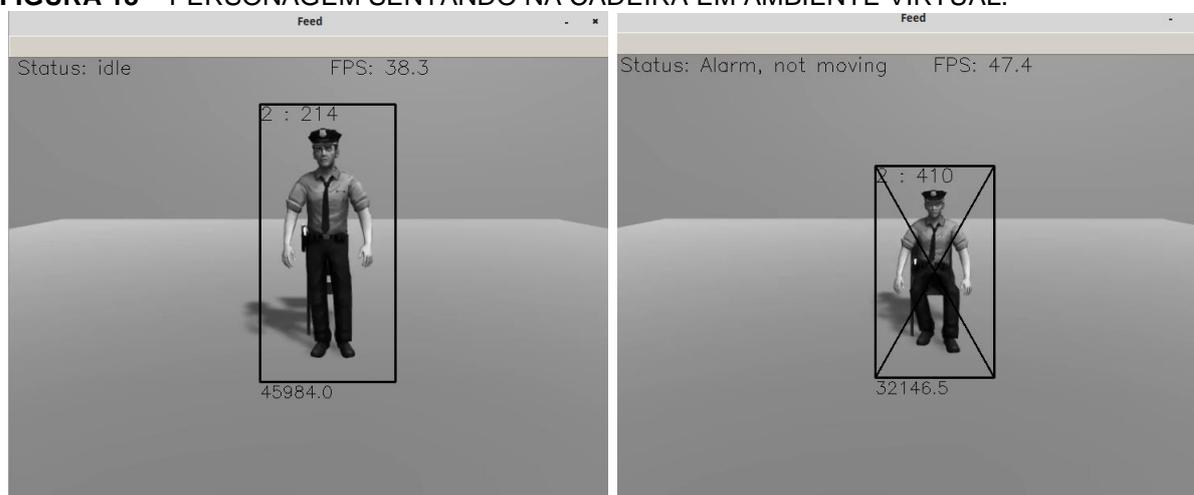
Fonte: Autoria própria, 2019.

No ambiente simulado real, apresentado na Figura 17, a pessoa presente na cena também foi detectada corretamente. Em ambos os casos não houve presença de queda e nem detecção por parte do algoritmo.

#### 4.3.2 Teste de sentar

Neste teste, o personagem entrou na área monitorada e sentou-se em uma cadeira durante algum tempo.

**FIGURA 18 – PERSONAGEM SENTANDO NA CADEIRA EM AMBIENTE VIRTUAL.**



Fonte: Autoria própria, 2019.

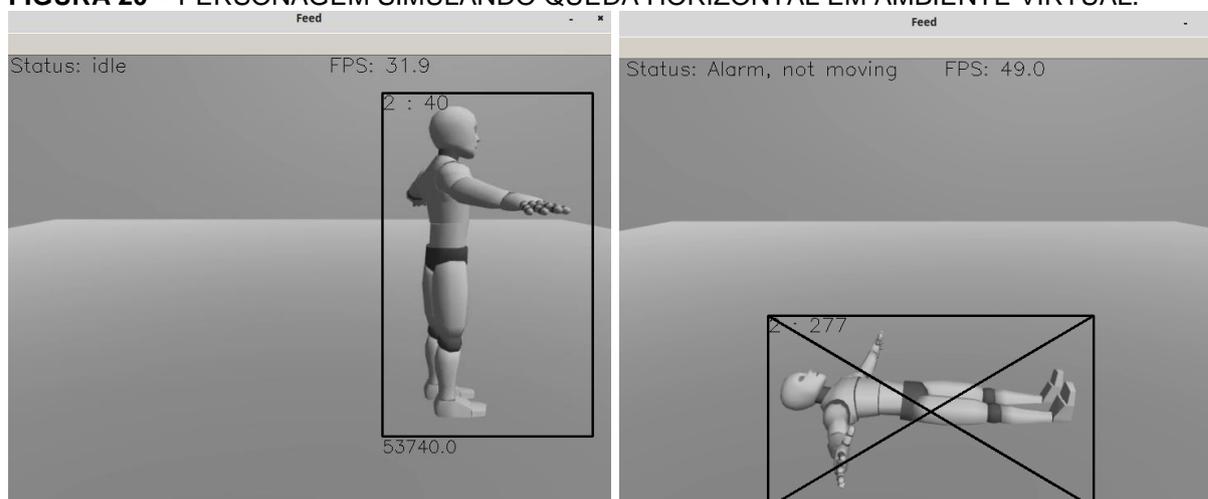
No ambiente simulado virtual, apresentado na Figura 18, o personagem quando estava em pé (à esquerda) foi identificado corretamente, não havendo classificação de queda neste instante, porém ao sentar (à direita), o algoritmo erroneamente classificou a ação como uma queda. O mesmo aconteceu no teste em ambiente simulado real, Figura 19. Assim, em ambos os casos a detecção resultou em um falso positivo, pois o algoritmo entendeu que aconteceu uma queda na vertical dado que houve uma mudança significativa na altura da caixa delimitadora sem haver redução similar na largura.

**FIGURA 19 – PERSONAGEM SENTANDO NA CADEIRA EM AMBIENTE REAL.**

Fonte: Autoria própria, 2019.

#### 4.3.3 Teste de queda com o corpo na horizontal

Neste teste, o personagem entrou na área monitorada e simulou uma queda horizontal, ficando deitado sem se mexer durante algum tempo.

**FIGURA 20 – PERSONAGEM SIMULANDO QUEDA HORIZONTAL EM AMBIENTE VIRTUAL.**

Fonte: Autoria própria, 2019.

No ambiente simulado virtual, apresentado na Figura 20, o personagem (imagem à esquerda) foi identificado corretamente. Ao cair na horizontal (imagem à direita), o algoritmo reconheceu a ação e após um tempo realizou o alerta de queda. O mesmo ocorreu no ambiente real, Figura 21, sendo que em ambos os casos a detecção de queda funcionou com sucesso.

**FIGURA 21 – PERSONAGEM SIMULANDO QUEDA HORIZONTAL EM AMBIENTE REAL.**

Fonte: Autoria própria, 2019.

#### **4.3.4 Teste de queda com o corpo na vertical**

Neste teste, o personagem entrou na área monitorada e simulou uma queda vertical, ficando deitado sem se mexer durante algum tempo.

**FIGURA 22 – PERSONAGEM SIMULANDO QUEDA VERTICAL EM AMBIENTE VIRTUAL.**

Fonte: Autoria própria, 2019.

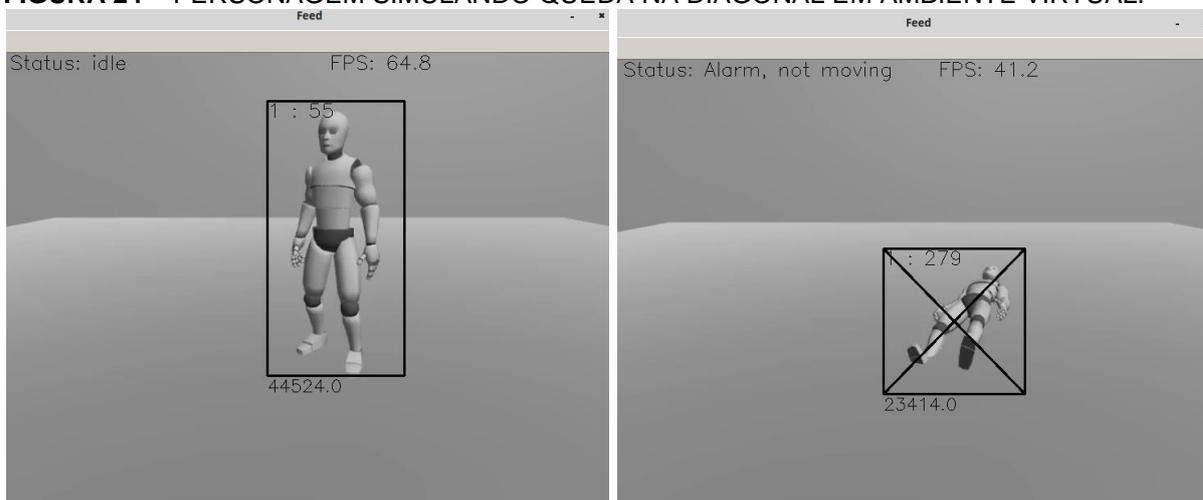
Em ambos os casos, tanto no teste em ambiente simulado virtual (Figura 22) e real (Figura 23), a identificação do personagem aconteceu adequadamente, como também, a detecção de queda na vertical foi bem-sucedida, enviando o alerta de quedas da maneira esperada.

**FIGURA 23 – PERSONAGEM SIMULANDO QUEDA VERTICAL EM AMBIENTE REAL.**

Fonte: Autoria própria, 2019.

#### 4.3.5 Teste de queda com o corpo na diagonal

Neste teste, o personagem entrou na área monitorada e simulou uma queda na diagonal, ficando deitado sem se mexer durante algum tempo.

**FIGURA 24 – PERSONAGEM SIMULANDO QUEDA NA DIAGONAL EM AMBIENTE VIRTUAL.**

Fonte: Autoria própria, 2019.

No ambiente simulado virtual, apresentado na Figura 24, o personagem foi identificado corretamente (imagem à esquerda). Após a queda na diagonal, o algoritmo funcionou corretamente enviando o alerta.

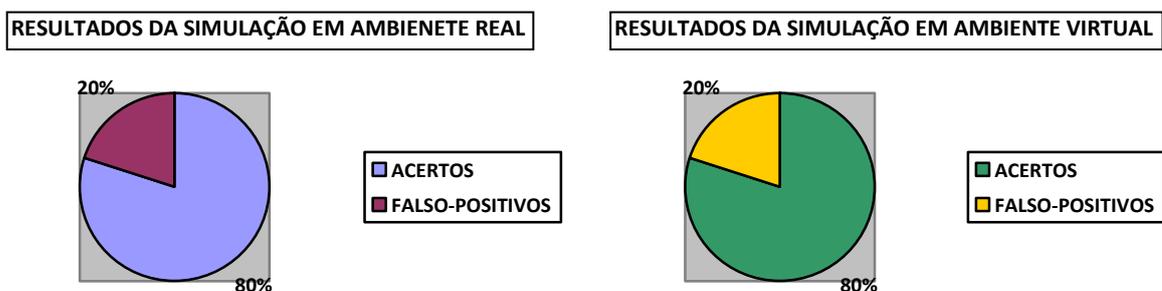
O mesmo ocorreu em ambiente simulado real, Figura 25, mas houve um problema na detecção que identificou uma parte da janela como um corpo caído (imagem à direita). Ainda assim o algoritmo de detecção agiu de maneira esperada, enviando dois alertas de queda por email conforme programado.

**FIGURA 25 – PERSONAGEM SIMULANDO QUEDA NA DIAGONAL EM AMBIENTE REAL.**

Fonte: Autoria própria, 2019.

#### 4.3.6 Resultados

Como observado Gráfico 1, tanto nos testes realizados em ambiente simulado real como no ambiente virtual, resultaram em 80% de sucesso e 20% de falso-positivos.

**GRÁFICO 1 – RESULTADOS DA SIMULAÇÃO EM AMBIENTE REAL E VIRTUAL.**

Fonte: Autoria própria, 2019.

#### 4.4 Testes do envio de alertas

Neste teste, foi avaliado o funcionamento do algoritmo de alerta de quedas, verificando se funcionou da maneira esperada e o tempo de duração do envio da mensagem.

Para a realização da validação do algoritmo, foram realizados 5 testes que deveriam enviar uma mensagem de alerta de queda com uma imagem em anexo de confirmação, dentre os quais, todos foram bem-sucedidos tendo um tempo médio de envio de 6 segundos, conforme observado na Tabela 1.

**TABELA 1 – TESTES DO ALGORITMO DE ALERTA DE QUEDAS.**

<b>TESTE</b>	<b>TEMPO (segundos)</b>	<b>VALIDAÇÃO</b>
Teste 1	7s	Sucesso
Teste 2	6s	Sucesso
Teste 3	6s	Sucesso
Teste 4	6s	Sucesso
Teste 5	6s	Sucesso

**Fonte:** Autoria própria, 2019.

Assim, segundo os testes realizados, o algoritmo de alerta de quedas possui 100% de eficácia além de ser bem eficiente, realizando um envio em um curto espaço de tempo.

## 5 CONSIDERAÇÕES FINAIS

A ideia defendida neste trabalho considera uma solução *home care* que utilize visão computacional para realizar a detecção de quedas, como também, realizar o envio de uma mensagem programada, a qual tem a função de alertar sobre o evento ocorrido. Assim, são utilizadas câmeras a fim de monitorar o indivíduo.

Além disso, a validação ocorreu através da realização de 10 testes em ambientes simulados, sendo 5 testes em ambiente real e 5 no virtual, no qual 8 (precisão de 80%) deles obtiveram sucesso e 2 retornaram um falso positivo, ambos realizando a ação de sentar.

O desenvolvimento do algoritmo de alerta de quedas tem como objetivo realizar o envio de mensagens pelo *Gmail* com uma imagem em anexo para comprovar a ocorrência de queda. Diante dos testes de validação realizados, foi comprovado que o envio foi realizado com sucesso em todos os testes, com tempo médio de apenas 6 segundos, demonstrando uma boa eficiência.

Como sugestão de trabalhos futuros, pode ser utilizada uma elipse aproximada para substituir a forma delimitadora na tentativa de melhorar a precisão, como também podem ser utilizados métodos de *stereovision* para obter dados em 3D da cena observada aplicando aprendizagem de máquina na tentativa de evitar os falso-positivos que ocorreram.

## REFERÊNCIAS

ALIVERTI, Andrea. Wearable technology: Role in respiratory health and disease. **Breathe**, v. 13, n. 2, p. e27-e36, 2017.

ANDERSON, Derek et al. Recognizing falls from silhouettes. In: **2006 International Conference of the IEEE Engineering in Medicine and Biology Society**. IEEE, 2006. p. 6388-6391.

BACKES, André Ricardo; JUNIOR, Jarbas Joaci de Mesquita Sá. **Introdução à visão computacional usando Matlab**. Rio de Janeiro: Alta Books Editora, 2019.

BALLARD, Dana H.; BROWN, Christopher M. **Computer Vision**. Englewood Cliffs: J: Prentice Hall, 1982.

BURGER, W.; BURGE, M. J. Digital Image Processing: An Algorithmic Approach Using Java. 2ª Edição. London: **Springer**, 2007.

CASTRO, Rafael Barreto; PEDRO, R. M. L. R. Redes de vigilância: experiência da segurança e da visibilidade articuladas às câmeras de monitoramento urbano. **Vigilância e visibilidade: espaço tecnologia e identificação. Porto Alegre, Sulina**, p. 36-60, 2010.

CORMEN, Thomas H. et al. **Algoritmos: teoria e prática**. Rio de Janeiro: Elsevier, 2002.

CUCCHIARA, Rita; PRATI, Andrea; VEZZANI, Roberto. A multi-camera vision system for fall detection and alarm generation. **Expert Systems**, v. 24, n. 5, p. 334-345, 2007.

DE QUEIROZ, José Eustáquio Rangel; GOMES, Herman Martins. Introdução ao processamento digital de imagens. **RITA**, v. 13, n. 2, p. 11-42, 2006

GODFREY, ACRMDOG et al. Direct measurement of human movement by accelerometry. **Medical engineering & physics**, v. 30, n. 10, p. 1364-1386, 2008.

HAMILL, Melinda et al. Development of an automated speech recognition interface for personal emergency response systems. **Journal of NeuroEngineering and Rehabilitation**, v. 6, n. 1, p. 26, 2009.

HSU, Ya-Wen; PERNG, Jau-Woei; LIU, Hui-Li. Development of a vision based pedestrian fall detection system with back propagation neural network. In: **2015 IEEE/SICE International Symposium on System Integration (SII)**. IEEE, 2015. p. 433-437.

KHAN, A. M. **Human activity recognition using a single tri-axial accelerometer**. Thesis for Degree of Sctor of Philosophy. Department of Computer Engineering. Seoul: Kyung Hee University, 2011.

KHARICHA, Kalpa et al. Health risk appraisal in older people 1: are older people living alone an 'at-risk'group?. **Br J Gen Pract**, v. 57, n. 537, p. 271-276, 2007.

KOOTEN, P. V. **Yagmail Documentation**, 2019. Disponível em: <<https://buildmedia.readthedocs.org/media/pdf/yagmail/latest/yagmail.pdf>>. Acesso em: 14 de out. de 2019.

LARA, J.P. R; MAYOR, J.J.V; OLIVEIRA, A. C. P. ; RODACKI, A.L.F. . IDENTIFICAÇÃO DO PADRÃO DE MARCHA DE IDOSOS ATRAVÉS DO SINAL DE ACELEROMETRIA BASEADO EM REDES NEURAIS ARTIFICIAIS. In: V Congresso Internacional do Envelhecimento Humano, 2017, Maceió. **Anais do V Congresso Internacional do Envelhecimento Humano**. João Pessoa: Realize, 2017. v. 2017.

LEE, Wei-Ju et al. Living alone as a red flag sign of falls among older people in rural Taiwan. **Journal of Clinical Gerontology and Geriatrics**, v. 2, n. 3, p. 76-79, 2011.

LYONS, Anthony R. Clinical outcomes and treatment of hip fractures. **The American journal of medicine**, v. 103, n. 2, p. S51-S64, 1997.

MARENGONI, M.; STRINGHINI, S. Tutorial: Introdução à visão computacional usando opencv. **Revista de Informática Teórica e Aplicada**, v. 16, n. 1, p. 125-160, 2009.

MASTORAKIS, Georgios; MAKRIS, Dimitrios. Fall detection system using Kinect's infrared sensor. **Journal of Real-Time Image Processing**, v. 9, n. 4, p. 635-646, 2014.

MUBASHIR, Muhammad; SHAO, Ling; SEED, Luke. A survey on fall detection: Principles and approaches. **Neurocomputing**, v. 100, p. 144-152, 2013

NORVIG, Peter; RUSSELL, Stuart. **Inteligência Artificial: Tradução da 3ª Edição**. Rio de Janeiro: Elsevier Brasil, 2014.

NOURY, Norbert et al. Fall detection-principles and methods. In: **2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society**. IEEE, 2007. p. 1663-1666.

PAULA, André Guilherme; DUARTE, Castro. **Validação de modelos de acelerometria para estimar a quantidade de atividade física habitual em adultos**. 2012. Dissertação (Mestrado em Exercício e Saúde) - Faculdade de Motricidade Humana, Universidade Técnica de Lisboa, Lisboa, 2012.

RAMOS, José Lúcio Costa; DE MENEZES, Maria do Rosário; MEIRA, Edméia Campos. Idosos que moram sozinhos: desafios e potencialidades do cotidiano. **Revista Baiana de Enfermagem**, v. 24, n. 1, 2, 3, 2011.

RASHIDI, Parisa; MIHAILIDIS, Alex. A survey on ambient-assisted living tools for older adults. **IEEE journal of biomedical and health informatics**, v. 17, n. 3, p. 579-590, 2013.

REZENDE, Solange Oliveira. **Sistemas inteligentes: fundamentos e aplicações**. Barueri: Editora Manole Ltda, 2003.

RIBEIRO, Sergio Silva. **Extração de características de imagens aplicada à detecção de grãos ardidos de milho**. 2015. 85 f. Dissertação (Mestrado em Computação para Tecnologias em Agricultura) - UNIVERSIDADE ESTADUAL DE PONTA GROSSA, Ponta Grossa, 2015.

RODRIGUES, Ricardo et al. Facts and figures on healthy ageing and long-term care. **European Centre for Social Welfare Policy and Research**, Vienna, 2012.

SALMI, K. **Improving safety for home care patients with a low cost computer vision solution**. 2016. 37 p. Thesis (Bachelor's Thesis - Degree Programme in IT) - Haaga-Helia University of Applied Sciences, Helsinki, Finland, 2016.

SILVA, Fernando Ginez da. **Reconhecimento de movimentos humanos utilizando um acelerômetro e inteligência computacional**. 2013. Dissertação (Mestrado em Microeletrônica) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2013.

SMITH, B. **Um guia rápido para a GPLv3**, 2018. Disponível em: <<https://www.gnu.org/licenses/quick-guide-gplv3.pt-br.html>>. Acesso em: 10 de out. de 2019.

SOLEM, J. E. **Programming Computer Vision with Python: Tools and algorithms for analyzing images**. " O'Reilly Media, Inc.", 2012.

TINETTI, Mary E.; WILLIAMS, Christianna S. The effect of falls and fall injuries on functioning in community-dwelling older persons. **The Journals of Gerontology Series A: Biological Sciences and Medical Sciences**, v. 53, n. 2, p. M112-M119, 1998.

WORLD HEALTH ORGANIZATION. **Global report on falls prevention in older age**. Geneva, p. 1 – 53, 2007. Acesso em: 19 mar. 2019. Disponível em: <[http://www.who.int/ageing/publications/Falls\\_prevention7March.pdf](http://www.who.int/ageing/publications/Falls_prevention7March.pdf)>.

ZIVKOVIC, Zoran. Improved adaptive Gaussian mixture model for background subtraction. In: **Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004**. IEEE, 2004. p. 28-31.

ZIVKOVIC, Zoran; VAN DER HEIJDEN, Ferdinand. Efficient adaptive density estimation per image pixel for the task of background subtraction. **Pattern recognition letters**, v. 27, n. 7, p. 773-780, 2006.

## APÊNDICE A – MÉTODO DA DETECÇÃO DE QUEDAS NA HORIZONTAL

Detecção-de-Quedas-Horizontal()

1. if larguraAgora > alturaAgora
2.       QuedaHorizontal <- true

## APÊNDICE B – MÉTODO DA DETECÇÃO DE QUEDAS NA VERTICAL

Detecção-de-Quedas-Vertical()

```
1. if alturaAnterior == 0
2.     alturaAnterior <- novaAltura
3.     larguraAnterior <- novaLargura
4. if tempoAgora > tempoAnterior
5.     if (alturaAgora + 20 < alturaAnterior) e (larguraAgora + 5 >= larguraAnterior)
6.         QuedaVertical <- true
7.     alturaAnterior <- alturaAgora
8.     larguraAnterior <- larguraAgora
9. tempoAnterior <- tempoAgora
```

## APÊNDICE C – MÉTODO DO ENVIO DE ALERTAS

Envio-de-Alerta-de-Quedas()

```
1. destinatario <- "Email de quem receberá a mensagem"
2. mensagem <- "informações textuais da mensagem"
3. imagem <- "caminho da localização da imagem a ser enviada como anexo"
4. Yag <- yagmail.SMTP ("Email do remetente", "senha do remetente")
5. Yag Envia(
6.     to <-destinatario,
7.     subject <- texto-do-titulo,
8.     contents <- mensagem,
9.     attachments <- imagem
10. )
```

**ANEXO A: TERMO DE LIVRE ESCLARECIMENTO**

UNIVERSIDADE ESTADUAL DA PARAÍBA – UEPB  
CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS - CCEA  
CAMPUS VII – GOVERNADOR ANTÔNIO MARIZ  
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Prezado (a) Senhor (a),

O (a) Sr. (a) está sendo convidado a participar desta pesquisa que tem por objetivo **desenvolver, usando visão computacional, uma solução home care para alerta de quedas em pessoas idosas.**

Sua participação no estudo consistirá em realizar simulações de queda e não queda diante de uma câmera que registrará as ações realizadas.

Os riscos com essa pesquisa são MÍNIMOS, caso o Sr (a). sinta-se desconfortável em realizar as simulações, tem a total liberdade de não realizar ou interromper a simulação em qualquer momento, sem nenhum prejuízo para seu atendimento.

**O (a) Sr (a). tem a liberdade de não participar da pesquisa** ou retirar seu consentimento a qualquer momento, mesmo após o início das simulações, sem qualquer prejuízo. **Está assegurada a garantia do sigilo das suas informações.** O(a) Sr(a). não terá nenhuma despesa e não há compensação financeira relacionada à sua participação na pesquisa.

Caso tenha alguma dúvida sobre a pesquisa o(a) Sr(a). poderá entrar em contato com o coordenador responsável pelo estudo: Fabio Júnior Francisco da Silva, que pode ser localizado na Universidade Estadual da Paraíba - UEPB (telefone 083-99600-6465) das 08 às 12h ou pelo e-mail fjr0206@gmail.com.

Sua participação é importante e voluntária e vai gerar informações que serão úteis para o desenvolvimento do algoritmo de detecção de quedas proposto nesta pesquisa.

Este termo será assinado em duas vias, pelo senhor e pelo responsável pela pesquisa, ficando uma via em seu poder.

Acredito ter sido suficientemente informado a respeito do que li ou foi lido para mim, sobre a pesquisa: "Visão computacional para detecção de quedas em pessoas idosas". Discuti com **David Danilo Alves de Oliveira**, responsável pela pesquisa, sobre minha decisão em participar do estudo. Ficaram claros para mim os propósitos do estudo, os procedimentos, garantias de sigilo, de esclarecimentos permanentes e isenção de despesas. Concordo voluntariamente em participar deste estudo.

*Nicete Alves de Oliveira*

27/10/2019

Assinatura do entrevistado

Declaro que obtive de forma apropriada e voluntária o Termo de Consentimento Livre e Esclarecido deste entrevistado OU REPRESENTANTE LEGAL para a sua participação neste estudo.

David Danilo Alves de Oliveira

Assinatura do responsável pelo estudo.

27/10/2019