

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
PRÓ-REITOR DE PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Rebeca Matos Silva

**Avaliação de Técnicas de Aprendizagem de Máquinas na Classificação de Imagens
Térmicas Extraídas de Câmeras de Baixa Resolução para Detecção de Quedas em
Idosos**

Fortaleza – CE

fevereiro de 2023

Rebeca Matos Silva

**Avaliação de Técnicas de Aprendizagem de Máquinas na Classificação de Imagens
Térmicas Extraídas de Câmeras de Baixa Resolução para Detecção de Quedas em
Idosos**

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, como requisito para obtenção do grau de Mestre em Ciência da Computação

Linha de Pesquisa: Inteligência Artificial

Orientador: Professor Doutor Elias Teodoro da Silva Júnior

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

Pró-Reitoria de Pesquisa, Pós-Graduação e Inovação (PRPI)

Programa de Pós-Graduação em Ciências da Computação (PPGCC)

Fortaleza – CE

Fevereiro de 2023

Dados Internacionais de Catalogação na Publicação
Instituto Federal do Ceará - IFCE
Sistema de Bibliotecas - SIBI
Ficha catalográfica elaborada pelo SIBI/IFCE, com os dados fornecidos pelo(a) autor(a)

- S586a Silva, Rebeca Matos.
Avaliação de Técnicas de Aprendizagem de Máquinas na Classificação de Imagens Térmicas Extraídas de Câmeras de Baixa Resolução para Detecção de Quedas em Idosos / Rebeca Matos Silva. - 2023.
66 f. : il.
- Dissertação (Mestrado) - Instituto Federal do Ceará, Mestrado em Ciência da Computação, Campus Fortaleza, 2023.
Orientação: Prof. Dr. Elias Teodoro da Silva Júnior.
1. Classificação de Imagens. 2. Aprendizado de máquina. 3. Sistemas Embarcados. I. Título.
CDD 004
-



Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)
Pró-Reitoria de Pesquisa, Pós-Graduação e Inovação (PRPI)
Programa de Pós-Graduação em Ciências da Computação (PPGCC)

Rebeca Matos Silva

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação, sendo aprovada pela Coordenação do Programa de Pós-Graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará e pela banca examinadora.

Orientador: Professor Doutor Elias Teodoro da Silva Júnior
Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

Professor Doutor Francisco Nivando Bezerra
Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

Professor Doutor Edgard de Faria Côrrea
Universidade Federal do Rio Grande do Norte (UFRN)

Fortaleza – CE
Fevereiro de 2023

Este trabalho é dedicado à minha família, em especial meus pais Joaquim Leite da Silva e Lusanira Bezerra de Matos Silva, irmãos Samuel Matos Silva e Daniel Matos Silva e aos meus sobrinhos João Vinicius Ferreira Matos Silva e Maria Clara Ferreira Matos Silva.

Agradecimentos

Quero deixar aqui registrado a minha mais sincera gratidão para com todos os que colaboraram, diretamente ou indiretamente, neste trabalho:

A Deus, que me orienta em todos os momentos. Ao professores Dr. Elias T. Silva Jr., pelos, ensinamentos, orientações, amizade, confiança, paciência e dedicação depositados ao longo deste trabalho, e principalmente, por acreditar em mim até o final.

A banca composta pelos professores Dr. Francisco Nivando Bezerra (IFCE) e Dr. Edgard de Faria Côrrea (UFRN) pelas contribuições oferecidas ao trabalho.

Aos meus pais Joaquim Leite da Silva e Lusanira Bezerra de Matos Silva, irmãos Samuel Matos Silva e Daniel Matos Silva pelo apoio incondicional, conselhos, força, amor e amizade sem igual durante todos esses anos.

A minha amiga Kamilla da Silva Ribeiro, por sempre me apoiar e incentivar em todos os projetos que participo.

A todos docentes do Programa de Pós-Graduação em Ciência da Computação (PPGCC-IFCE), pelo ensino de boa qualidade, atenção e apoio.

*“Eu não descobri um modo de fazer uma lâmpada.
Descobri mil modos de como não fazê-la.”*
Thomas Edison

Resumo

A queda tende a gerar danos físicos e psicológicos, principalmente para as pessoas com idade acima dos 65 anos. Tais danos podem ser agravados caso as mesmas morem sozinhas e não consigam ajuda no tempo adequado. Este é um dos fatores que acarretam o aumento do uso de dispositivos inteligentes nas residências, atuando como forma de monitorar alguma ação ou incidente. Tais dispositivos tem um bom nível de aceitação por proporcionarem aos usuários sensações de bem-estar, comodidade e/ou segurança. Esses dispositivos realizam suas finalidades através da captura de dados do ambiente em que estão localizados e posterior análise. Atualmente, técnicas de aprendizagem de máquina estão atuando como forma de otimizar e trazer cada vez mais autonomia e eficácia ao funcionamento desses dispositivos. O aprendizado profundo ganha destaque ao se tratar de soluções que realizam algum tipo de análise em imagens. Porém, ao se tratar de soluções embarcadas, um modelo com muitas camadas de abstração pode ser inviável para processamento local. Este trabalho apresenta um estudo de técnicas de aprendizagem de máquina, com o intuito de analisar a viabilidade do desenvolvimento de um sistema de monitoramento do ambiente, capaz de detectar incidentes de queda, sem invadir a privacidade do usuário, podendo ser instalado em qualquer ambiente da casa. Para isso, é realizado a classificação de imagens térmicas utilizando os descritores de atributos, GLCM (Gray Level Co-occurrence Matriz) e LBP (Local Binary Patterns), combinado a técnicas de aprendizado de máquina e aprendizado profundo. Durante o processo de treinamento do modelo, foram utilizadas imagens de dois sensores distintos, redimensionadas para o tamanho padrão de 64x64 *pixels*. É observado uma acurácia média de 91,66% usando a rede LSTM (Long short-term memory ou memória de longo prazo) bidirecional, e 97,88% utilizando a rede VGG16. Utilizando arquiteturas mais simples, como o descritor GLCM combinado ao classificador KNN (K-Nearest Neighbor) chegamos a obter uma acurácia média de 89,73%, já com o classificador SVM (Support Vector Machine) a acurácia média chegou a 91,23%. Enquanto com LBP a melhor acurácia média obtida foi de 98,20% com o classificador SVM e de 96,58% com o KNN. Esses resultados apresentam a viabilidade da detecção de quedas, utilizando imagens térmicas de baixa resolução e arquiteturas de classificação que possuem poucas camadas de processamento, tornando viável a possibilidade de obtermos uma solução em microcontroladores.

Palavras-chave: Imagens térmicas, Classificação de Imagens. GLCM, LBP. Aprendizado de máquina, KNN, SVM, Bi-LSTM, VGG16. IoT, Sistemas Embarcados.

Abstract

Falls tend to generate physical and psychological damage, especially for people over 65 years of age. Such damage can be aggravated if they live alone and do not get help in a timely manner. This is one of the factors that lead to the increase in the use of smart devices in homes, acting as a way to monitor some action or incident. Such devices have a good level of acceptance for providing users with feelings of well-being, comfort and/or safety. These devices accomplish their purposes by capturing data from the environment in which they are located and subsequent analysis. Currently, machine learning techniques are acting as a way to optimize and bring more and more autonomy and effectiveness to the operation of these devices. Deep learning gains prominence when dealing with solutions that perform some type of analysis on images. However, when dealing with embedded solutions, a model with many layers of abstraction may be unfeasible for local processing. This work presents a study of machine learning techniques, with the aim of analyzing the feasibility of developing an environment monitoring system, capable of detecting incidents of falls, without invading the user's privacy, and can be installed in any environment of the company. House. For this, thermal image classification is performed using the attribute descriptors, GLCM (Gray Level Co-occurrence Matrix) and LBP (Local Binary Patterns), combined with machine learning and deep learning techniques. During the model training process, images from two different sensors were used, resized to the default size of 64x64 pixels. An average accuracy of 91.66% is observed using the bidirectional LSTM (Long short-term memory) network, and 97.88% using the VGG16 network. Using simpler architectures, such as the GLCM descriptor combined with the KNN (K-Nearest Neighbor) classifier, we obtained an average accuracy of 89.73%, while with the SVM (Support Vector Machine) classifier, the average accuracy reached 91.23%. While with LBP the best average accuracy obtained was 98.20% with the SVM classifier and 96.58% with the KNN. These results show the feasibility of detecting falls, using low resolution thermal images and classification architectures that have few processing layers, making it feasible to obtain a solution in microcontrollers.

Keywords: Thermal Imaging, Image Classification. GLCM, LBP. Machine Learning, KNN, SVM, Bi-LSTM, VGG16. IoT, Embedded Systems.

Lista de Ilustrações

Figura 1 - Porcentagem de pessoas de diferentes idades no mundo em diferentes anos.....	25
Figura 2 – Etapas de um sistema de visão computacional	29
Figura 3 – Comparação de Imagens Redimensionadas	30
Figura 4 – Representação da GLCM.....	32
Figura 5 – Algoritmo kNN	34
Figura 6 - Classificação kNN.....	34
Figura 7 – Arquitetura das redes recorrentes	37
Figura 8 - Arquitetura do modelo VGG16.....	37
Figura 9 - Saída do sensor IRISYS	39
Figura 10 - Frames obtidos durante a simulação de uma queda.....	40
Figura 11 - Imagem extraída do sensor MLX90620	42
Figura 12 - Imagem extraída do sensor MLX90640	42
Figura 13 - Fluxo de procedimentos adotado na pesquisa	45
Figura 14 - Amostra de imagem original do dataset utilizado	47
Figura 15 - Processo para gerar as imagens extraídas dos sensores	47
Figura 16 - Imagem extraída do sensor MLX90640	48
Figura 17 - Imagem extraída do sensor D6T-8L-06.....	48
Figura 18 - Imagens do sensor MLX90640 em diferentes mapas de cores	49
Figura 19 - Comparação das imagens redimensionadas do sensor MLX90640.....	49
Figura 20 - Processo de ampliação das imagens extraídas do sensor D6T-8L-06	50
Figura 21 – Comparação de Imagens 93x125 – Original e Binarizada	50
Figura 22 – Comparação de Imagens 32x32 – Original e Binarizada	50
Figura 23 – Resumo das camadas da rede Bi-LSTM	52
Figura 24 - Resumo da rede VGG16 adaptada	53
Figura 25 – Imagens extraídas do dataset 2.....	56
Figura 26 - Imagens extraídas do dataset 1.....	57
Figura 27 – Distribuição de valores dos atributos extraídos da GLCM com imagem original	58
Figura 28 - Distribuição de valores dos atributos extraídos da GLCM com imagem binarizada.....	59
Figura 29 – Imagem binarizada em comparação com a imagem em escala de cinza.....	59
Figura 30 – Imagem obtida da extração LBP em comparação com a original do dataset 1 para raio 1, 2 e 3.....	60
Figura 31 - Imagem obtida da extração LBP em comparação com a original do dataset 2 para raio 1, 2 e 3.....	60

Lista de Tabelas

Tabela 1 - Resumo de kernels mais populares	35
Tabela 2 - Citações com foco em detecção de quedas e uso de câmeras térmicas	38
Tabela 3 - Descrição dos sensores utilizados para criação do dataset.....	46
Tabela 4 - Características gerais dos voluntários	46
Tabela 5 - Resultado do treinamento das arquiteturas apresentadas	57
Tabela 6 – Resultados Obtidos para o dataset 1.....	60
Tabela 7 – Resultados obtidos para o dataset 2.....	61
Tabela 8 – Tempo Médio para Extração e Classificação do dataset 1.....	61
Tabela 9 - Tempo Médio para Extração e Classificação do dataset 2.....	62

Lista de abreviaturas e siglas

IoT	Internet of Things (Internet das Coisas)
IR	Infravermelho
SVA	Sistema de Visão Artificial
GLCM	Gray-Level Co-occurrence Matrix (Matriz de Coocorrência de Níveis de Cinza)
LPB	Local Binary Pattern (Padrão Binário Local)
SVM	Máquinas de Vetores de Suporte
LSTM	Long Short-Term Memory (Memória de Longo Prazo)
Bi-LSTM	Bidirectional Long Short-Term Memory (Memória de Longo Prazo Bidirecional)
k-NN	K – Nearest Neighbors (K-ésimo Vizinho Mais Próximo)
HQX	High-quality magnification filter (Filtro de ampliação de alta qualidade)
ReLU	Rectified Linear Units (Unidade Linear Retificadora)
MLP	Multilayer Perceptron (Perceptron Multicamada)
3D-CNN	3D Convolutional Neural Network (Rede Neural Convolutacional 3D)
ROC	Receiver Operating Characteristic (Características de Operação do Receptor)
AOA	Ângulo de chegada
SDA	Serial Data
SCL	Serial Clock
RMSprop	Root Mean Squared Propagation (Propagação Quadrática Média)

Sumário

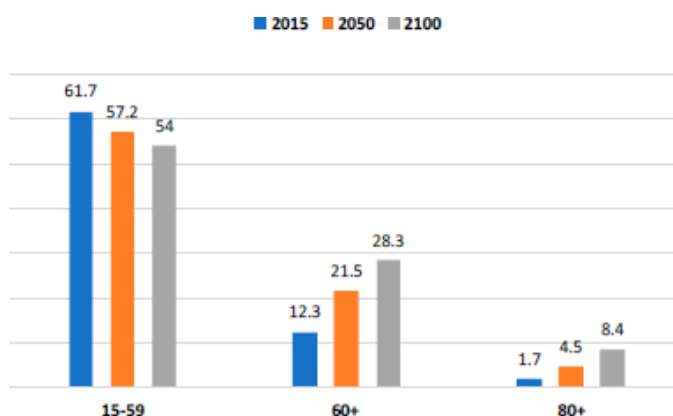
1.	INTRODUÇÃO	25
1.1	MOTIVAÇÃO PARA A DISSERTAÇÃO	27
1.2	DESCRIÇÃO DO PROBLEMA	27
1.3	OBJETIVOS GERAL E ESPECÍFICO.....	27
1.4	RELEVÂNCIA	28
1.5	ESTRUTURA DA DISSERTAÇÃO.....	28
2.	FUNDAMENTAÇÃO TEÓRICA	29
2.1.	PROCESSAMENTO DAS IMAGENS TÉRMICAS	29
2.1.1.	REDIMENSIONAMENTO DE IMAGENS COM OPENCV	30
2.1.2.	REDIMENSIONAMENTO DE IMAGENS COM HQX	30
2.2.	EXTRATORES DE CARACTERÍSTICAS.....	31
2.2.1.	MATRIZ DE COCORRÊNCIA DE NÍVEIS DE CINZA	31
2.2.2.	PADRÕES BINÁRIOS LOCAIS.....	33
2.3.	MODELOS DE APRENDIZAGEM	34
2.3.1.	KNN	34
2.3.2.	SVM	35
2.3.3.	LSTM BIDIRECIONAL	35
2.3.4.	VGG16	37
3.	TRABALHOS RELACIONADOS	38
3.1.	ESTADO DA ARTE UTILIZANDO SENSORES COM MATRIZ QUADRADA	39
3.2.	ESTADO DA ARTE UTILIZANDO SENSORES COM MATRIZ RETANGULAR	41
3.3.	EXTRATORES DE ATRIBUTOS GLCM E LBP EMBARCADOS	43
3.4.	INFERÊNCIA DE MODELOS DE REDES NEURAI NA BORDA DA REDE.....	44
3.5.	CONSIDERAÇÕES	44
4.	MATERIAIS E MÉTODOS	45
4.1.	ESCOLHA DO DATASET	45
4.1.1.	DATASET UTILIZADO.....	46
4.1.2.	PRÉ PROCESSAMENTO DOS DADOS	48
4.2.	TREINAMENTO DAS ARQUITETURAS.....	51
4.2.1.	APRENDIZADO PROFUNDO	51
4.2.1.1.	Bi-LSTM.....	51
4.2.1.2.	VGG16.....	53
4.2.2.	EXTRATOR DE ATRIBUTOS COM CLASSIFICADOR.....	54
4.2.2.1.	PARÂMETROS GLCM.....	54
4.2.2.2.	PARÂMETROS LBP.....	54

4.2.3. CLASSIFICAÇÃO E AVALIAÇÃO.....	54
5. RESULTADOS OBTIDOS	56
5.1. APRENDIZADO PROFUNDO	56
5.2. EXTRATORES DE ATRIBUTOS.....	57
5.2.1. GLCM.....	58
5.2.2. LBP.....	59
5.3. CLASSIFICAÇÃO COM KNN E SVM	60
5.4. ANÁLISE DOS RESULTADOS	62
6. CONSIDERAÇÕES FINAIS.....	64
Referências	65

1. INTRODUÇÃO

De acordo com a United Nations (2017), em 2030 a população mundial com idade acima de 60 anos atingirá de cerca de 1,4 bilhão de pessoas, e de acordo com o World Health Statistics (2008), 28% a 35% das pessoas com idade acima de 65 anos caem ao menos uma vez ao ano, sendo que este percentual pode chegar a 42% se considerarmos o grupo de pessoas com idade acima de 70 anos. Baseado na figura 1, é possível observar que a população idosa (com 60 anos ou mais) terá um crescimento maior do que a população com idade entre 15 e 59 anos. Se a tendência continuar, não haverá pessoas suficientes para cuidar dos idosos em um futuro distante, portanto, tecnologias de vida assistida serão necessárias no futuro para cuidar de pessoas idosas e ajudá-las a viver de forma independente (UDDIN; KHAKSAR; TORRESEN, 2018).

Figura 1 - Porcentagem de pessoas de diferentes idades no mundo em diferentes anos.



Fonte: (UNITED NATIONS, 2015).

Muito além da real necessidade de se ter um acompanhante, outro fator importante a se observar no que tange os cuidados a população idosa é a independência, que muitas vezes lhe é privada por excesso de cuidados de familiares. O medo de cair e não conseguir ter uma devida recuperação posterior também amplifica as consequências negativas do pós-queda e pode diminuir a confiança do paciente (FRIEDMAN, 2002). Tais situações acarretam na limitação do paciente, redução na interação social e depressão (A.C.SCHEFFER, 2008).

Em um recente trabalho apresentado por Gawronska e Lorkowski (2021) é levantado o fato de que, com a eclosão da pandemia de SARS-CoV-2 e a consequente diminuição no número de lesões resultantes de acidentes de trânsito e lesões esportivas, problemas por fraturas decorrentes de atividades de baixa energia como, por exemplo, escorregões, tropeços e quedas, ganham destaque. Em um estudo realizado por LV et al. (2020), mostrou que 58,3% de todas as fraturas registradas na China ocorria no quadril e o mecanismo dominante da lesão foi um queda de uma posição em pé (89,4%). Verificou-se também que 72,7% de todas as fraturas ocorreram na casa do paciente. Outro fator importante a se observar é o tempo de detecção e acionamento de resgate no caso de quedas, pois pode minimizar os prejuízos decorrentes do acidente (WANG *et al.*, 2014).

Soluções microcontroladas com a finalidade de automatizar alguma tarefa, monitorar ambiente e emitir alertas baseado em dados capturados por sensores são cada vez mais comuns

em nosso dia-a-dia e se mostram como um dos principais meios de solucionar o problema em questão. Conforme cita Mubashir (2013), tais soluções são geralmente separadas em 3 grupos: sistemas baseados em dispositivos vestíveis, sistemas que capturam informações obtidas a partir de sensores instalado no ambiente e câmeras de monitoramento. Os sistemas baseados em dispositivos vestíveis têm sua eficácia limitada ao uso direto do dispositivo por parte do paciente, ou seja, caso o mesmo venha a esquecer de acoplar a solução ao seu corpo tal dispositivo passa a não ter eficiência. Sensores instalados no ambiente possuem seu campo de atuação limitado ao local de instalação, assim como as câmeras de monitoramento, esta por sua vez, podendo se tornar desagradável por invadir a privacidade do usuário e inviabilizar seu uso em compartimentos mais restritos da residência, como quartos e banheiros.

Uma das formas de caracterizar as informações obtidas através dos sensores e/ou câmeras como queda ou não é através de algoritmos de aprendizagem de máquina, que permite que um algoritmo descubra regras e padrões através dos dados passados para ele, ou seja, podemos criar programas que fazem previsões com base em dados complexos sem ter que entender toda a complexidade do sistema. Através de um processo chamado treinamento o algoritmo de aprendizagem de máquina cria um modelo baseado nos dados fornecidos. Novos dados são passados por esse modelo para fazer previsões, em um processo chamado de inferência (WARDEN; SITUNAYAKE, 2019). As soluções mais comuns costumam usar o dispositivo embarcado para capturar os dados e realizam a inferência em um servidor em nuvem, tornando a conexão à internet uma característica relevante a ser observada nas soluções microcontroladas apresentadas.

Técnicas de extração de atributos das imagens são utilizadas como forma de simplificar a quantidade de recursos necessários para descrever um grande conjunto de dados, afinal, realizar a análise de um grande número de variáveis geralmente requer uma grande quantidade de memória e poder computacional (MOHANAIHA; SATHYANARAYANA; GURUKUMAR, 2013). Neste trabalho foi utilizado dois descritores baseados em texturas: o GLCM (Gray Level Co-Occurrence Matrix) que utiliza uma técnica de extração de características a partir de uma matriz de coocorrência de níveis de cinza e o LBP (Local Binary Patterns) que se baseia em padrões binários locais.

Como classificador, foi utilizado quatro arquiteturas baseadas em aprendizagem de máquina: KNN (K-nearest neighbors), que realiza a classificação avaliando a distância dos dados em relação ao seu vizinho mais próximo, SVM (Support Vector Machine), que é um classificador discriminativo definido por um hiperplano de separação, Bi-LSTM (Bidirectional Long Short Term Memory), uma rede neural recorrente, e a VGG-16, uma rede neural convolucional. Os classificadores foram escolhidos por terem apresentado bons desempenhos em trabalhos similares e possuírem níveis de complexidade distintos, a fim de se avaliar a melhor arquitetura viável para o projeto.

Como resultado, este estudo visa avaliar a possibilidade de detectar quedas em idosos usando imagens de sensores térmicos de baixa resolução e o uso de técnicas de aprendizagem de máquina para identificar acidentes. Será realizada uma avaliação de vários modelos com o objetivo de desenvolver uma arquitetura aceitável para embarcação, evitando assim o envio de dados para um servidor externo. Este objetivo é criar uma solução otimizada que realiza a classificação de imagens localmente.

1.1 MOTIVAÇÃO PARA A DISSERTAÇÃO

A principal motivação para este trabalho é analisar a viabilidade de uso de imagens térmicas capturadas a partir de sensores de baixa resolução monitorando o ambiente como forma de detectar um movimento característico de queda. Dentre as soluções existentes para a solução do problema proposto, algumas apresentadas por Uddin, Khaksar e Torresen (2018), a utilização de imagens térmicas se mostra menos invasiva, por não expor diretamente o usuário, e mais cômoda, por não necessitar de um acoplamento direto ao usuário.

Apesar de encontrarmos na literatura soluções que trabalham diretamente com imagens térmicas, imagens com melhores resoluções, que detalham melhor a cena capturada, como as extraídas de câmeras de monitoramento, costumam ser mais usadas. No entanto, quanto maior a resolução da imagem a ser analisada, mais exposto estará o usuário.

O envio de imagens para processamento em nuvem, o que aumenta o tempo de resposta, é um elemento relevante adicional. Estudos relacionados na literatura mostram isso. Como resultado, analisamos também a possibilidade de embarcar um modelo de aprendizado de máquina com melhor desempenho como substituto do processamento de dados em nuvem.

1.2 DESCRIÇÃO DO PROBLEMA

Um dos fatores que mais afetam a qualidade de vida e independência de pessoas idosas são as consequências de uma queda. Acidentes deste tipo em ambientes residenciais se tornam cada vez mais comum, e os casos foram aumentando nos últimos dois anos diante do enfrentamento a pandemia de SARS-CoV-2. Em paralelo a isto, duas áreas de estudo apresentam uma constante evolução: o aprendizado de máquina e a Internet das Coisas (IoT). Essas áreas apresentam destaque no que diz respeito a soluções embarcadas que trazem algum tipo de automação, segurança, comodidade e bem-estar ao usuário, e estão cada vez mais presentes no cotidiano da população em geral. O caso de estudo abordado neste trabalho é a análise da viabilidade do uso de uma solução embarcada que realize a detecção de queda em idosos sem violar a privacidade, através do uso de imagens térmicas e processamento local.

1.3 OBJETIVOS GERAL E ESPECÍFICO

O trabalho em questão visa avaliar a viabilidade do desenvolvimento de uma solução embarcada inteligente, capaz de capturar imagens térmicas do ambiente e realizar a identificação de quedas, realizando o processamento dos dados localmente.

Os objetivos listados abaixo apresentam o que é necessário para que a finalidade proposta seja cumprida:

- Preparação de um dataset adequado para o treinamento das arquiteturas a serem validadas para uso.
- Uso de descritores de características com a finalidade de reduzir o número de informações a serem processadas pelo classificador.
- Uso de algoritmos de aprendizagem de máquina como classificador de imagens, de forma a detectar se o movimento capturado se caracteriza como queda ou não.
- Comparar os valores obtidos com o uso de cada arquitetura, a fim de encontrar um modelo adequado a solução proposta.

1.4 RELEVÂNCIA

A presente dissertação de mestrado é relevante pois apresenta um estudo completo da viabilidade de se ter uma solução embarcada que realize inferências localmente de um modelo de aprendizagem de máquina profundo com o intuito de detectar a ocorrência de uma queda, por parte de usuários idosos, através da análise de imagens térmicas de baixa resolução capturadas do ambiente.

As propostas similares encontradas na literatura utilizam apenas o sensor local como forma de capturar as informações e realizam o envio para um servidor externo realizar o processamento das informações capturadas. Portanto, o poder computacional exigido para realizar a classificação não é uma preocupação apresentada nas soluções.

1.5 ESTRUTURA DA DISSERTAÇÃO

Para além da introdução, este trabalho de qualificação contém mais 4 seções, contendo os conteúdos apresentados a seguir:

- No capítulo 2, temos a apresentação dos principais conceitos e tecnologias utilizadas no desenvolvimento deste trabalho
- No capítulo 3, é descrito o estado da arte e são apresentados alguns trabalhos com finalidade similar a desta pesquisa.
- No capítulo 4, é apresentado os materiais e métodos utilizados no desenvolvimento, assim como as etapas do trabalho desenvolvido.
- O capítulo 5 apresenta os resultados obtidos até o momento.
- No capítulo 6, temos as considerações finais relacionados aos resultados obtidos até o presente momento e as próximas etapas a serem desenvolvidas.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os principais conceitos utilizados neste trabalho, focando no que é mais pertinente ao entendimento dos experimentos realizados. Inicialmente, será apresentado os conceitos que caracterizam uma imagem térmica e os processamentos utilizados para adequação das mesmas aos testes realizados. Em um segundo momento, será abordado as características dos extratores de atributos e, por fim, são explicadas as arquiteturas de classificação utilizadas no experimento como forma de comparação dos resultados obtidos e escolha da melhor a ser usada em uma solução final.

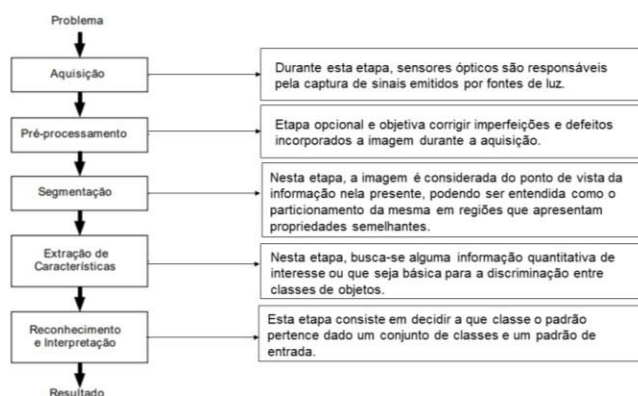
2.1. PROCESSAMENTO DAS IMAGENS TÉRMICAS

A distribuição de temperatura na superfície de um objeto pode ser representada usando um método chamado de imagem térmica, muitas vezes também referenciado como termografia (HRVATSKO PRIRODOSLOVNO DRUŠTVO. BIOLOŠKA SEKCIJA. *et al.*, 2011).

A energia infravermelha é emitida por todos os materiais acima de 0° Kelvin (-273° Celsius). A radiação IR (infravermelha) faz parte do espectro eletromagnético e ocupa frequências entre a luz visível e as ondas de rádio. A parte de RI do espectro abrange comprimentos de onda de 0,7 micrômetro (μm) a 1000 μm . Dentro desta largura de banda, somente frequências entre 0,7 μm e 20 μm são atualmente usadas para medição de temperatura (PAŠAGIĆ; MUŽEVIĆ, 2008). Essa energia é convertida em sinais elétricos por sensores de imagem na câmera e exibido em um monitor como uma imagem térmica colorida ou monocromática que representa as variações dos valores de temperatura (HILDEBRANDT; RASCHNER; AMMER, 2010).

Segundo Gonzales e Woods (2010), o processamento digital de imagens envolve processos cuja entradas e saídas são imagens e, além disso, envolve processos de extração de atributos de imagens até – e inclusive – o reconhecimento de objetos individuais. Marques Filho e Vieira Neto (1999) definem um Sistema de Visão Artificial (SVA) como um sistema computadorizado capaz de adquirir, processar e interpretar imagens correspondentes a cenas reais. A figura 2 apresenta um diagrama de um sistema de visão computacional com as respectivas explicações de cada etapa.

Figura 2 – Etapas de um sistema de visão computacional



Fonte: (MARQUES FILHO; VIEIRA NETO, 1999)

Nesta dissertação, o problema consiste em classificar em imagens térmicas de baixa resolução. Para isso, três algoritmos foram usados como forma de preparar as imagens para o processo de classificação, dois relacionados ao redimensionamento da imagem e um voltado para segmentação das mesmas. Tais algoritmos são abordados a seguir.

2.1.1. REDIMENSIONAMENTO DE IMAGENS COM OPENCV

A biblioteca de código aberto OpenCV, iniciada como um projeto de pesquisa da Intel em 1998, visa oferecer as ferramentas necessárias para resolver problemas de visão computacional (PULLI; BAKSHEEV, 2012). Uma das funções disponibilizadas por esta biblioteca tem a finalidade de redimensionar as imagens e é chamada de Resize.

É possível realizar o redimensionamento das imagens ao longo de cada eixo (altura e largura), considerando os fatores de escala especificados ou apenas definindo a altura e largura desejada. Neste trabalho, a função foi utilizada para aumentar o tamanho das imagens de baixa resolução, de forma a serem melhor aceitas pelos algoritmos de aprendizado profundo. Ao ampliar uma imagem, é preciso se atentar que:

- Aumentar o tamanho de uma imagem requer reconstrução da imagem. Isso significa que você precisa interpolar novos *pixels*;
- A proporção da imagem original (largura por altura) pode ser mantida na imagem redimensionada, para tanto, é preciso conhecê-la.

2.1.2. REDIMENSIONAMENTO DE IMAGENS COM HQX

HQX (Filtro de ampliação de alta qualidade) é um dos algoritmos de escala de pixel art desenvolvido por Maxim Stepin (STEPIN, 2003). O algoritmo HQX examina um bloco de pixel de tamanho 3x3 e compara o pixel central com seus 8 vizinhos. Cada vizinho é classificado como semelhante ou diferente em cores, o que leva a 256 combinações possíveis. O algoritmo usa uma tabela de pesquisa para aplicar um esquema de interpolação personalizado para cada combinação. Isso permite que ele produza várias formas, como cantos vivos, aumentando a qualidade dos resultados. As tabelas de pesquisa existem apenas para ampliação de 2 (hq2x), 3 (hq3x) e 4 (hq4x) fatores (KOPF; LISCHINSKI, 2011).

Figura 3 – Comparação de Imagens Redimensionadas



(a) Utilizando interpolação nearest-neighbor

(b) Utilizando hq4x

Fonte: (KOPF; LISCHINSKI, 2011)

De acordo com a figura 3, é possível observar que utilizando o algoritmo hq4x, obtemos uma qualidade de imagem ampliada bem melhor do que usando os algoritmos convencionais, com destaque para os cantos dos objetos ampliados na imagem, que aparecem bem mais definidos na ampliação com algoritmo hq4x.

O HQX usa o espaço de cores YUV (Y sendo o componente de brilho e U e V são os componentes de cromaticidade) para calcular as diferenças de cores, de modo que as diferenças de brilho sejam mais ponderadas para imitar a percepção humana.

2.1.3. LIMIAZIZAÇÃO DAS IMAGENS

Uma forma simples de segmentar uma imagem é utilizando a limiarização. A seleção de um limiar é uma técnica que permite separar o elemento principal do fundo da imagem. De acordo com Gonzales e Woods (2010), qualquer ponto (x, y) na imagem em que $f(x, y) > T$ é chamado de ponto do objeto; caso contrário, o ponto é chamado de ponto de fundo. Em outras palavras, a imagem segmentada, $g(x, y)$, é dada por:

$$g(x, y) = \begin{cases} 1, & \text{se } f(x, y) > T \\ 0, & \text{se } f(x, y) \leq T \end{cases} \quad (2.1)$$

Devido ao fato da limiarização produzir uma imagem binária à saída, o processo também é denominado, muitas vezes, binarização. A forma mais simples de limiarização consiste na bipartição do histograma, convertendo os pixels cujo tom de cinza é maior ou igual a um certo valor de limiar (T) em brancos e os demais em pretos (MARQUES FILHO; VIEIRA NETO, 1999).

2.2. EXTRATORES DE CARACTERÍSTICAS

A extração de características é uma etapa importante na construção de qualquer classificação de padrões e visa a extração das informações relevantes que caracterizam cada classe. Segundo Kumar e Bhatia (2014), as características devem conter informações necessárias para distinguir entre classes, ser insensível a irrelevantes variabilidades na entrada, e também ser limitado em número, para permitir o cálculo eficiente das funções discriminantes e para limitar a quantidade de dados exigidas no treinamento.

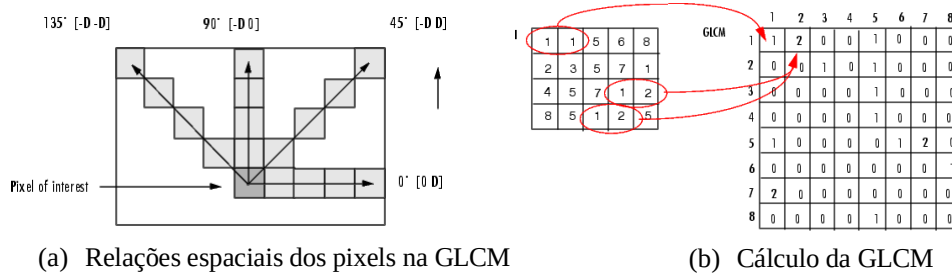
2.2.1. MATRIZ DE COOCORRÊNCIA DE NÍVEIS DE CINZA

A Matriz de Coocorrência de Níveis de Cinza (GLCM), proposta por Haralick, Shanmugam, e Dinstein (1973), é uma técnica utilizada para medir a textura de imagens, onde cada entrada (i, j) na Matriz de Coocorrência corresponde ao número de ocorrências do par de níveis de cinza i e j que estão a uma distância D e direção θ na imagem original.

A figura 4(a) ilustra as relações espaciais de pixels que são definidas por essa matriz de deslocamentos, onde D é a distância do pixel de interesse. A orientação θ pode ser representada em quatro direções: horizontal, diagonal, vertical e anti-diagonal (0° , 45° , 90° e 135°).

Já a figura 4(b) exemplifica como a matriz é calculada. Na saída GLCM, o elemento (1,1) contém o valor 1 porque há apenas uma instância na imagem de entrada onde dois pixels adjacentes horizontalmente têm os valores 1 e 1, respectivamente. O elemento (1,2) contém o valor 2 porque há duas instâncias em que dois pixels adjacentes horizontalmente têm os valores 1 e 2. O elemento (1,3) no GLCM tem o valor 0 porque não há instâncias de dois pixels adjacentes horizontalmente com os valores 1 e 3.

Figura 4 – Representação da GLCM



Fonte: (MATHWORKS, 2017)

Dentre as medidas estatísticas que podem ser usadas para representar características a partir de uma Matriz de Coocorrência, será descrito abaixo apenas as adotadas, inicialmente, neste trabalho.

Notação:

- $p(i, j)$: $p(i, j)$ -ésima entrada em uma matriz normalizada de dependência espacial de níveis de cinza $p(i, j)/R$;
- $p_x(i)$: i -ésima entrada em uma matriz de probabilidade marginal obtida pela soma das linhas de $p(i, j)$, $\sum_{j=1}^{N_g} p(i, j)$;
- $p_y(j) = \sum_{i=1}^{N_g} p(i, j)$;
- N_g : Número de níveis de cinza na imagem quantizada;
- $\sum i$ e $\sum j$: forma simplificada de $\sum_{i=1}^{N_g} i$ e $\sum_{j=1}^{N_g} j$, respectivamente;
- $p_{x-y}(k) = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i, j)$ para $k = 2, 3, \dots, 2N_g - 1$
 $|i-j=k|$ g

Atributos:

- Segundo Momento Angular (ASM):

$$f_1 = \sum_i \sum_j [p(i, j)]^2 \quad (2.2)$$

- Contraste (Contrast):

$$f_2 = \sum_{k=0}^{N_g-1} k^2 [p_{x-y}(k)] \quad (2.3)$$

- Correlação (*Correlation*):

$$f_3 = \frac{\sum_i \sum_j (i, j) p(i, j) - \mu_x \mu_y}{\sigma_x \sigma_y} \quad (2.4)$$

- Homogeneidade (*Homogeneity*):

$$f_4 = \sum_i \sum_j \frac{p(i, j)}{1 + |i - j|} \quad (2.5)$$

- Dissimilaridade (*Dissimilarity*):

$$f_5 = \sum_i \sum_j p(i, j) |i - j| \quad (2.6)$$

- Energia (*Energy*):

$$f_6 = \sqrt{ASM} \quad (2.7)$$

2.2.2. PADRÕES BINÁRIOS LOCAIS

Os Padrões Binários Locais (Local Binary Pattern, LBP) são operadores de textura simples, mas muito eficiente, que rotulam os pixels de uma imagem limitando a vizinhança de cada pixel e considera o resultado como um número binário. Devido ao seu poder discriminativo e simplicidade computacional, o operador de textura LBP tornou-se uma abordagem popular em várias aplicações. Pode ser visto como uma abordagem unificadora para os modelos estatísticos e estruturais tradicionalmente divergentes da análise de textura. Outra propriedade importante é sua simplicidade computacional, que possibilita a análise de imagens em configurações desafiadoras em tempo real (PIETIKÄINEN *et al.*, 2011).

O operador LBP original, descrito por Ojala et al. (1996), forma rótulos para os pixels da imagem limitando a vizinhança 3 x 3 de cada pixel com o valor central e considerando o resultado como um número binário. O histograma desses 256 rótulos diferentes pode então ser usado como um descritor de textura. Este operador usado em conjunto com uma medida de contraste local simples forneceu um desempenho muito bom na segmentação de textura não supervisionada (OJALA; PIETIKÄINEN, 1999).

Pietikäinen et al. (2011) propuseram uma metodologia, denominada Padrões Uniformes (Uniform Patterns), para reduzir, significativamente, a dimensionalidade gerada pelo operador LBP original. Essa extensão foi inspirada pelo fato de que alguns padrões binários ocorrem mais comumente em imagens de textura do que outros. Um padrão binário local é chamado de uniforme se o padrão binário contém no máximo duas transições bit a bit de 0 para 1 ou vice-versa quando o padrão de bits é percorrido circularmente. Por exemplo, os padrões 00000000 (0 transições), 01110000 (2 transições) e 11001111 (2 transições) são uniformes, enquanto os padrões 11001001 (4 transições) e 01010010 (6 transições) não são. Na computação dos rótulos LBP, padrões uniformes são usados para que haja um rótulo separado para cada padrão uniforme e todos os padrões não uniformes são rotulados com um único rótulo. Portanto, o número de rótulos diferentes para a vizinhança de P pixels é $P(P - 1) + 3$. Uma vizinhança 3 x 3 possui 59 padrões únicos (58 para os padrões uniformes e 1 para os não uniformes). Ojala,

Pietikainen e Maenpaa (2002) observaram em seus experimentos que uma vizinhança 3×3 de padrões uniformes representa, aproximadamente, 90% de todos os padrões da textura.

2.3. MODELOS DE APRENDIZAGEM

Durante o desenvolvimento deste trabalho, utilizamos algumas arquiteturas de classificação, como forma de encontrar uma que seja mais adequada a solução do problema. Dentre os modelos já disponíveis e consolidados com finalidade de classificação, neste trabalho abordaremos: kNN (k-Nearest Neighbours ou k Vizinhos mais próximos), SVM (Support Vector Machine ou Máquina de Vetores de Suporte), LSTM (Long short-term memory ou memória de longo prazo) Bidirecional, ou Bi-LSTM, sem pesos e VGG16 com pesos herdados da Imagenet.

2.3.1. KNN

O algoritmo KNN foi introduzido por Aha et al. (1991), e a ideia geral desse algoritmo consiste em encontrar os k exemplos rotulados mais próximos do exemplo não classificado e, com base no rótulo desses exemplos mais próximos, é tomada a decisão relativa a classe do exemplo não rotulado. As etapas de um algoritmo kNN são apresentadas na figura 5.

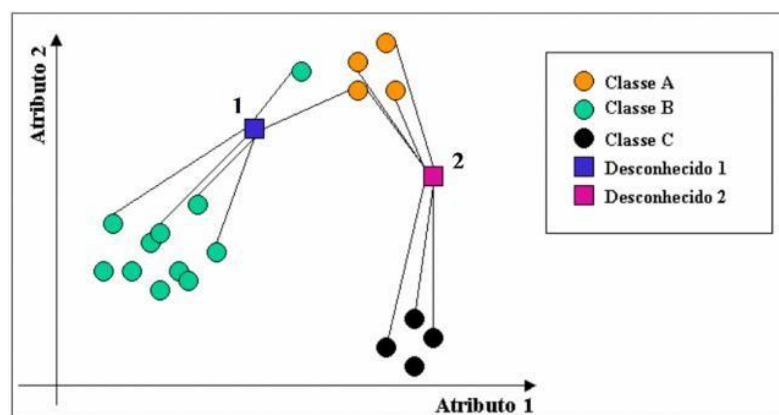
Figura 5 – Algoritmo kNN



Fonte: Autor

Baldiasserotto (2005) explica, conforme a Figura 6, que para classificar um ponto, primeiro é preciso recuperar os k-vizinhos mais próximos e, dentro desse conjunto, encontrar a classe mais representativa. Na Figura 6, o ponto desconhecido 1 será classificado como a classe B, e o ponto desconhecido 2 será classificado como a classe A. Baldiasserotto (2005) ainda adverte que a escolha do parâmetro K, deve ser um número ímpar, pois um número par, poderia causar conflito quando um ponto tivesse o mesmo número de vizinhos de cada classe.

Figura 6 - Classificação kNN



Fonte: (BALDISSEROTTO, 2005)

Uma das formas de se calcular a distância é através da fórmula da distância Euclidiana, descrita abaixo, onde as distâncias são definidas por p_i e p_j , e p_{ik} e p_{jk} para $k = 1, \dots, n$ são os n atributos que descrevem as instâncias de p_i e p_j , respectivamente.

$$d = \sqrt{\sum_{k=1}^n (p_{ik} - p_{jk})^2} \quad (2.8)$$

2.3.2. SVM

A SVM (Support Vector Machine ou Máquina de Vetores Suporte) é uma técnica embasada na Teoria de Aprendizado Estatístico (TAE), que consiste em estabelecer condições matemáticas que permitem escolher um classificador, com bom desempenho, para o conjunto de dados disponíveis para treinamento e teste (VAPNIK, 1995).

A SVM se baseia em aprendizado supervisionado, assim como o kNN, e originalmente desenvolvida para classificação binária, busca a construção de um hiperplano como superfície de decisão, de tal forma que a separação entre os exemplos seja máxima. Isso com padrões linearmente separáveis, ou seja, onde é possível separar os dados com um hiperplano. Para os padrões não-linearmente separáveis, utiliza-se uma função de mapeamento (kernel) apropriada para tornar o conjunto mapeado em linearmente separável.

De acordo com (ROCHA NETO, 2011), as funções de mapeamento mais utilizadas são a linear, polinomial e gaussiano (RBF), apresentados na tabela 1.

Tabela 1 - Resumo de kernels mais populares

Kernel	Descrição
Linear	$K(x, x_i) = x_i^T \cdot x$
Polinomial	$K(x, x_i) = (x_i^T \cdot x + 1)^d$, em que d é o grau do polinômio
Gaussiano (RBF)	$K(x, x_i) = \exp \left\{ -\frac{\ x - x_i\ ^2}{\sigma^2} \right\}$, em que σ é uma constante

Fonte: (ROCHA NETO, 2011)

2.3.3. LSTM BIDIRECIONAL

A rede LSTM (Long short-term memory ou memória de longo prazo) é uma rede neural recorrente citada originalmente por Sepp Hochreiter (1997), que utiliza um mecanismo chamado *gate* (portas) para aprender sobre as dependências de longas sequências. As redes neurais recorrentes são comumente utilizadas na solução de problemas que envolvem a análise de séries temporais: reconhecimento de fala, tradução, legenda de vídeos e etc, pois tomam decisões baseadas não só na informação atual de entrada, levando em consideração também as informações aprendidas das entradas recebidas anteriormente. O que diferencia uma rede LSTM da arquitetura padrão de uma rede recorrente é que o modelo LSTM tem uma memória

maior, conseguindo trabalhar com séries de dados mais longas sem esbarrar em um problema conhecido como *vanishing gradient*.

A arquitetura LSTM possui um mecanismo conhecido como três portas: entrada, esquecimento e saída, descritos da seguinte forma no trabalho de Siامي-Namini, Tavakoli e Namin (2019):

- I. Porta de Esquecimento: Toma a decisão sobre qual informação precisa ser removida da memória LSTM, utilizando uma função não linear σ , que pode ser uma sigmoid, tangente hiperbólica ou ReLU (Unidade Linear Retificadora). A função é descrita na equação (2.9) e sua saída será um valor entre 0 e 1, no qual 0 implica em descarte do valor e 1 em sua preservação.

$$f_t = \sigma(W_{fh}[h_t - 1], W_{fx}[x_t], b_f) \quad (2.9)$$

Sendo:

x : sequência dos dados de entrada de tamanho T;

h_t : memória da rede no ponto T;

W: matriz de pesos;

b_f : constante de bias.

- II. Porta de Entrada: Possui duas camadas, uma sigmoid e uma tanh, e decide se a nova informação será ou não armazenada na memória LSTM. A função sigmoid decide qual valor será atualizado e a tanh cria um vetor de valores candidatos a serem armazenados na memória LSTM. A equação (2.10) apresenta a representação matemática desse portão:

$$c_t = f_t * c_{t-1} + i_t * c_t \quad (2.10)$$

Onde:

$$i_t = \sigma(W_{ih}[h_{t-1}], W_{ix}[x_t], b_i) \quad (2.11)$$

$$c_t = \tanh(W_{ch}[h_{t-1}], W_{cx}[x_t], b_c) \quad (2.12)$$

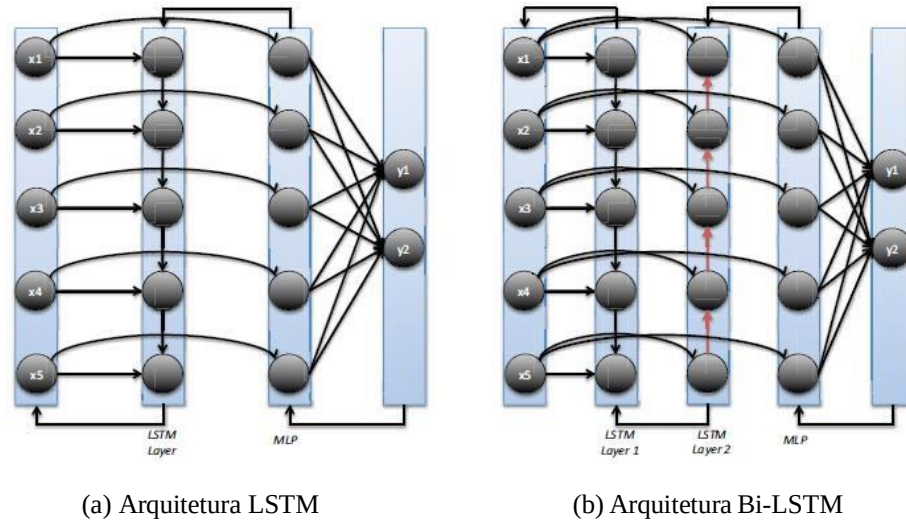
- III. Porta de Saída: Inicialmente usa uma camada *sigmoid* para decidir qual parte da memória LSTM contribui para a saída. Em seguida, executa uma função *tanh* não linear para mapear os valores entre -1 e 1. O resultado é multiplicado pela saída de uma camada *sigmoid*, conforme equações (2.13) e (2.14):

$$o_t = \sigma(W_{oh}[h_{t-1}], W_{ox}[x_t], b_o) \quad (2.13)$$

$$h_t = o_t * \tanh(c_t) \quad (2.14)$$

No qual o_t é o valor da saída e h_t é representado por um valor entre -1 e 1.

Na Bi-LSTM, duas LSTMs são aplicadas as informações de entrada, uma vai analisar os dados em sua ordem temporal convencional, e a outra na ordem reversa, conforme apresentado na figura 7.

Figura 7 – Arquitetura das redes recorrentes

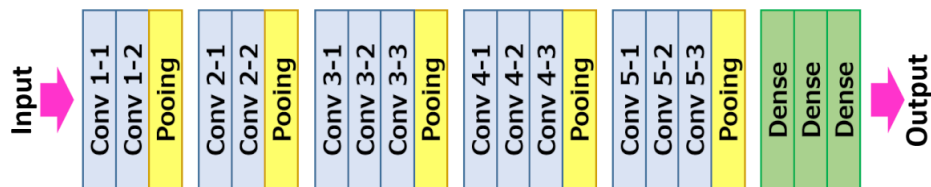
Fonte: (SIAMI-NAMINI; TAVAKOLI; NAMIN, 2019)

2.3.4. VGG16

Proposta originalmente por Simonyan e Zisserman (2015), a VGG16 é uma rede neural convolucional que se apresenta como uma melhoria da rede AlexNet e está no top 5 de acurácia em testes de classificação da Imagenet, que é um dataset composto de aproximadamente 14 milhões de imagens pertencentes a 1000 classes.

A VGG16 trabalha, por padrão, com imagens de 224x224 *pixels*, podendo aceitar imagens com, no mínimo, 32x32 *pixels*. A imagem entra inicialmente em uma pilha de camadas convolucionais com filtros de dimensão 3x3, sendo utilizado também filtros 1x1, que podem ser vistos como uma transformação linear dos canais de entrada. Na camada de convolução o *stride* é fixado em 1 para preservar a resolução espacial após a convolução.

A figura 8 apresenta a arquitetura da rede VGG16. A rede possui três camadas totalmente conectadas, sendo as duas primeiras com 4096 canais cada e a terceira com 1000 canais, uma para cada classe da Imagenet. Todas as camadas ocultas usam ReLU.

Figura 8 - Arquitetura do modelo VGG16

Fonte: (PATEL; KRISHNAN, 2020)

3. TRABALHOS RELACIONADOS

O estudo realizado por Uddin, Khaksar e Torresen (2018) apresenta um comparativo de diversas soluções baseada no uso de sensores para monitoramento de idosos. Em relação aos sensores vestíveis o estudo destaca sensores que medem parâmetros como aceleração, velocidade, força magnética, frequência cardíaca, temperatura corporal, saturação de oxigênio, frequência respiratória, eletrocardiograma e etc, e aponta como principais desvantagens de tais sensores a dificuldade de instalação, que muitas vezes requer a ajuda de um profissional. Por outro lado, os sensores externos são instalados no ambiente, não afetando a rotina do usuário, e coletam vários tipos de dados para modelar os eventos ou atividades, como quedas, localização, maneira de andar, conseguindo antecipar as necessidades do usuário e melhorar a qualidade de vida.

Outro fator importante levantado por este trabalho está relacionado a privacidade do usuário. Soluções que realizam o monitoramento das atividades diárias e aspectos de saúde dos usuários precisam manter os dados capturados sob sigilo. Os serviços de saúde exigem que os dados sejam armazenados em segurança, podendo ser disponibilizados apenas para profissionais de saúde, parentes ou o próprio usuário. Câmeras de monitoramento são utilizadas em algumas situações e se destacam por apresentarem uma visão mais clara da situação, mas acabam não sendo utilizadas por conta da violação de privacidade, e se tornam inviável para uso em ambientes privados, como quarto e banheiro (NOURY *et al.*, 2007).

Como forma de preservar a segurança e privacidade dos usuários, soluções que utilizam sensores térmicos de baixa resolução, detectando postura e movimentação dos usuários no ambiente são cada vez mais propostas, conforme apresentado na tabela 2.

Tabela 2 - Citações com foco em detecção de quedas e uso de câmeras térmicas

Autor	Sensor IR	Nº Pixels	Nº Sensores	Método	Acurácia
(SIXSMITH; JOHNSON, 2004)	IRISYS	16x16	1	MLP	30.0%
(MASHIYAMA; HONG; OHTSUKI, 2015)	AMG8831	8X8	1	SVM	100.0%
(WEI-HAN CHEN; HSI-PIN MA, 2015)	MLX90620	16x4	2	Classificador k-NN	93.0%
(HAYASHIDA; MOSHNYAGA; HASHIMOTO, 2017)	AMG8831	8X8	1	Limiar de distância	94.0%
(RIQUELME <i>et al.</i> , 2019)	D6T-8L-06/ MLX90640	1x8 / 32x24	4	Bi-LSTM	93.0%
(LIU <i>et al.</i> , 2020)	AMG8853	8x8	1	Random Forest	94.0%
(TATENO <i>et al.</i> , 2020)	MLX90640	32x24	1	LSTM / 3D CNN	82.5% / 94.2%

Fonte: Autor

3.1. ESTADO DA ARTE UTILIZANDO SENSORES COM MATRIZ QUADRADA

Uma das primeiras soluções encontradas foi proposta por Sixsmith e Johnson (2004), usando o IRISYS (InfraRed Integrated Systems), que gera uma imagem de 16x16 *pixels*. A figura 9 apresenta uma comparação de uma imagem extraída de uma câmera de monitoramento e a imagem térmica gerada pelo sensor IRISYS. Apesar da baixa qualidade da imagem obtida através deste sensor, conforme observado na figura 10, ainda é possível realizar a localização e o rastreamento de um alvo térmico no campo de visão do sensor, fornecendo informações de tamanho, localização e velocidade.

Figura 9 - Saída do sensor IRISYS



(a) Imagem de câmera convencional (b) Imagem térmica obtida do sensor

Fonte: (SIXSMITH; JOHNSON, 2004)

O estudo propõe a solução embarcada denominada SIMBAD (*Smart Inactivity Monitor using Array-Based Detectors*), que identifica a queda a partir da velocidade vertical calculada baseada nos dados coletados pelo sensor. Primeiro, o movimento do alvo é analisado para detectar a dinâmica característica da queda. Posteriormente, o sistema monitora a inatividade do alvo e compara com um mapa de períodos aceitáveis de inatividade em diferentes locais do campo de visão.

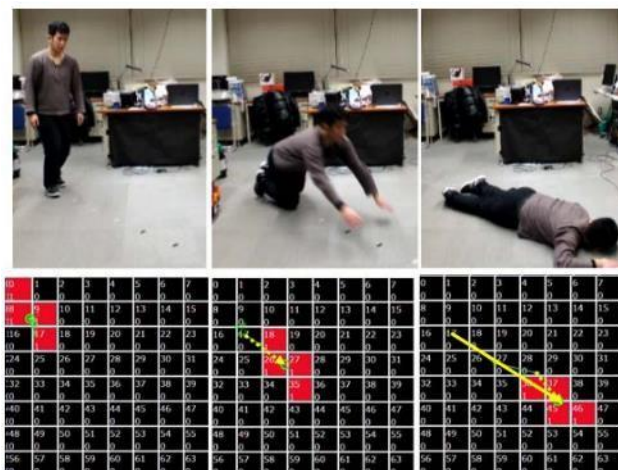
As imagens utilizadas para treinamento são obtidas através da simulação de queda por uma atriz e classificadas através de um modelo de rede perceptron multicamada. Os dados gerados cobrem uma variedade de tipos de queda, pontos de vista e graus de obscuridade, gerando aproximadamente 10000 vetores de amostras para treinamento da rede, cada vetor classificado como uma queda ou não queda.

Em testes realizados em laboratório, a detecção da queda só ocorreu em 30% dos cenários de queda simulados. O mau resultado é atrelado pelo autor a uma má generalização dos dados de treinamento, já que nos testes o sensor foi posicionado de forma a capturar um ângulo diferente do ângulo das imagens obtidas para treinamento do modelo. Assim como, foi observado que as estimativas de velocidade vertical que serviram como as entradas para o classificador se mostraram bastante inviáveis. Outro ponto levantado pelo autor é que o sistema pode exigir mais informações, além da velocidade vertical, para obter um bom desempenho.

Outro sensor encontrado na literatura é a série de sensores de matriz infravermelho Grid-EYE AMG88, fabricado pela empresa Panasonic, que produz uma matriz de 64 *pixels* dispostos em 8x8 e consegue detectar a temperatura absoluta em apenas uma mudança de temperatura. Isso permite que o Grid-EYE colete mais dados sobre os corpos quentes em seu campo de visão,

assim como é capaz de ver objetos imóveis e em que direção os objetos estão se movendo, sem invadir a privacidade das pessoas (PANASONIC, 2016). A figura 10 apresenta um exemplo de frames de imagens obtidos através de um dos sensores da série AMG88 durante a simulação de uma queda.

Figura 10 - Frames obtidos durante a simulação de uma queda.



Fonte: (HAYASHIDA; MOSHNYAGA; HASHIMOTO, 2017)

Utilizando o sensor AMG8831, Mashiyama, Hong e Ohtsuki (2015) propõem um sistema para reconhecimento de atividades diárias nas residências de idosos, composto por 5 classes: sem evento, parado, andando, sentado e caindo. Nesse estudo, o cenário de teste considera o sensor instalado no teto e a presença de apenas uma pessoa no ambiente a ser observado. Os autores criam um algoritmo seguindo o seguinte fluxo: detecção de corpo humano – de forma a classificar os *pixels* em pessoa e plano de fundo, extração de características – quatro características baseadas nos movimentos do corpo humano são extraídas de acordo com a distribuição de temperatura, e classificação – as 4 características obtidas são comparadas com os dados obtidos durante o treinamento, de forma a serem classificados como “andando”, “sentado” ou “caindo”.

Como técnica para classificação, os autores sugerem o uso de SVM (Máquinas de Vetores de Suporte), K-NN (k-Vizinhos mais próximos) e redes neurais, mas adota SVM nos seus experimentos e aponta o percentual de acurácia encontrado para cada classe analisada, sendo a detecção de queda atingindo os 100% e o menor percentual de acertos para sentado (78.6%). Como forma de melhorar os resultados, os autores sugerem a análise de outras características da imagem e a alteração do posicionamento do sensor como forma de melhorar a acurácia das classes que não atingiram 100%.

O trabalho proposto por Hayashida, Moshnyaga e Hashimoto (2017) utiliza, além da câmera térmica AMG8831, o sensor de temperatura ADT7410 e sugere o uso do protocolo de comunicação Xbee para conectar vários nós de sensores como forma de cobrir uma área maior da residência. Neste trabalho a detecção da queda é feita em 4 passos principais: aquisição da imagem térmica, extração do primeiro plano da imagem – subtraindo o plano de fundo da imagem original obtida, extração das características para detecção de queda – quatro características são selecionadas: diferença máxima de temperatura, distância máxima percorrida durante o movimento, duração do movimento e a variância máxima da diferença térmica entre o primeiro plano e o plano de fundo, e, por fim, a detecção da queda.

A queda é detectada baseada na análise do vetor de características obtido da imagem extraída. Este trabalho apresenta um diferencial na análise de seus resultados, mostrando que a variação de temperatura do ambiente, ruídos oriundos de equipamentos eletrônicos e as formas de luminância do ambiente podem afetar a acurácia do sistema. Foi observado, por exemplo, que quanto mais quente estava o ambiente, menor foi a taxa de acertos, chegando a 83% com 28°C, enquanto atingiu 97% com 18°C. Apesar de propor uma solução utilizando vários nós de sensores, os experimentos foram realizados utilizando um sistema de apenas um nó em laboratório, não tendo uma base de resultados obtidos em outros ambientes ou com o sensor capturando diferentes ângulos do recinto.

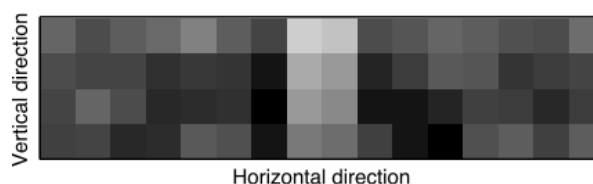
Em seu estudo, Liu *et al.* (2020) utiliza o sensor AMG8853, destacando como vantagens desse tipo de solução o baixo custo, a preservação da privacidade, a baixa complexidade, a detecção em tempo real e a alta acurácia obtida. Seu sistema considera a captura dos dados em tempo real e transferência para um host. Primeiro é utilizada interpolação bicúbica e subtração de fundo para separar o plano de fundo do objeto detectado, removendo assim ruídos da imagem. Em seguida, foi utilizado o método do componente conectado para selecionar o bloco de maior área e identificar a sua posição centróide, detectando assim ponto estacionário, ou possível ponto de queda, da pessoa detectada. A partir daí, algumas características são extraídas para determinar o movimento da pessoa, como a hora da chegada estacionária e as áreas estacionárias, antes e atuais, já que cair não é uma ação instantânea e detectar uma queda pode exigir vários quadros.

Os testes do sistema foram realizados em laboratório, com o sensor posicionado no teto e conectado a um computador via USB para transferência dos dados capturados. 8 voluntários com altura entre 1,60m e 1,80m participaram do processo de captura das imagens. Foi realizado o treinamento do algoritmo Random Forest para classificação dos dados. Para avaliar o desempenho do classificador, a curva ROC (Características de Operação do Receptor), que ilustra a taxa de falsos positivos e a taxa de verdadeiros positivos para cada categoria, foi apresentada. Os autores apresentaram uma acurácia de 94% em um sistema de classificação binária, queda e não queda. Como trabalho futuro, o autor sugere usar uma faixa de detecção maior, utilizando vários sensores para estender a área de cobertura.

3.2. ESTADO DA ARTE UTILIZANDO SENSORES COM MATRIZ RETANGULAR

Um dos sensores muito utilizado em trabalhos similares, é a série de sensores MLX906, da Melexis. Uma das soluções encontradas é proposta por Wei-Han Chen e Hsi-Pin Ma (2015), utilizando o MLX90620, que gera uma imagem com resolução de 16x4 pixel. Um exemplo de imagem extraída do sensor pode ser visto na figura 11. Nesse estudo, os autores propõem o uso de 2 sensores no ambiente, posicionados a 3.3m de distância horizontal entre eles, como forma de obter uma visão mais clara do ambiente, gerando uma imagem de 32x8 pixel.

Figura 11 - Imagem extraída do sensor MLX90620



Fonte: (WEI-HAN CHEN; HSI-PIN MA, 2015)

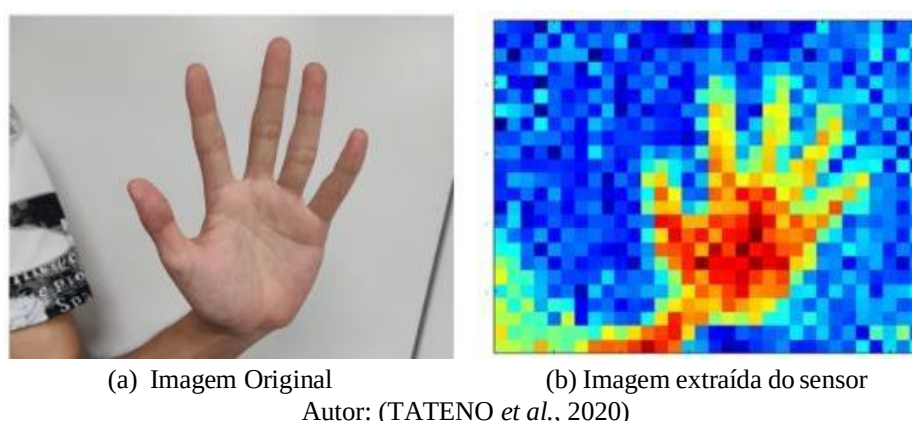
Após obtidas, as imagens são pré-processadas para remoção de ruído e extração do primeiro plano. Para estimar a localização da pessoa detectada, é usado o algoritmo de posicionamento AOA (ângulo de chegada). O AOA de cada sensor é determinado pelo cálculo da soma ponderada das regiões de primeiro plano. Com o resultado da AOA de cada sensor, a posição pode ser detectada calculando o ângulo de intercepção.

5 sujeitos participaram do estudo, simulando ações como sentar, curvar, agachar, andar e permanecer de pé, performando 16 vezes cada ação, gerando 80 ações de queda e 80 atividades normais. Para reduzir a complexidade computacional, apenas a imagem de um dos sensores era analisada para determinar a queda, sendo essa escolhida baseada na imagem que apresentasse uma maior área de primeiro plano, caracterizando uma maior proximidade do alvo para o sensor.

A detecção da queda é realizada utilizando o algoritmo classificador K-NN. Para avaliar a performance do sistema, além da acurácia, foi levado em consideração os valores de sensibilidade e especificidade. Os resultados apontaram que o sistema consegue distinguir um evento de queda com sensibilidade de 95,25%, especificidade de 90,75% e acurácia total de 93%.

Outra solução encontrada é proposta por Tateno *et al.* (2020), utilizando o sensor MLX90640, que gera uma imagem com resolução de 32x24 pixel. A figura 12 apresenta a imagem obtida através deste sensor. Conforme esperado, devido a maior resolução, a imagem apresenta uma melhor qualidade, na qual já é possível obter uma imagem com mais detalhes.

Figura 12 - Imagem extraída do sensor MLX90640



Assim como os demais, para realizar a detecção do movimento, o autor realiza um pré-processamento dos dados, removendo ruído e plano de fundo da imagem, em seguida efetua a localização do alvo, determinada pelos *pixels* que apresentam maior temperatura e, por fim,

realiza o processo de classificação. Para efetuar a classificação, neste trabalho, o autor utiliza dois métodos como forma de realizar um comparativo dos resultados.

O primeiro método de classificação utilizado é chamado de 3D-CNN (rede neural convolucional de três dimensões). A 3D-CNN é composta por uma camada de convolução, uma camada de pooling, uma camada totalmente conectada e uma camada de softmax. O segundo método utilizado foi o LSTM (Long short-term memory ou memória de longo prazo), que possui uma estrutura usada para processar sequências de dados que mudam com o tempo, espaço ou outros fatores, conforme explicado no capítulo anterior. Para realizar o experimento, 16 participantes simularam 8 tipos de movimentos, dentre eles, cair, sentar, levantar e andar.

Os resultados mostraram valores melhores de acurácia do modelo 3D-CNN se comparado a LSTM, uma média 94.2% e 82,5%, respectivamente, considerando todos os tipos de movimento, porém os tempos de processamento para treino e teste das redes LSTM, tanto com GPU (unidade de processamento gráfica) quanto sem GPU são menores, podendo chegar a uma diferença de até 2 horas, se comparado a 3D-CNN, sem o uso de GPU. Um ponto importante levantado pelos autores é que a taxa de acurácia do sistema cai ao ser detectado mais de um alvo na imagem, chegando a 72% no modelo 3D-CNN, considerando 3 pessoas no ambiente. O autor também sugere que, o número de características extraídas das imagens obtidas para o modelo LSTM pode não ter sido suficiente para a correta distinção dos 8 movimentos, resultando uma acurácia abaixo do esperado.

3.3. EXTRATORES DE ATRIBUTOS GLCM E LBP EMBARCADOS

Várias pesquisas publicadas apresentam o uso de matriz de coocorrência para a extração de características e posterior classificação de imagens, porém não foi encontrado um estudo que apresente a viabilidade para embarcar essa técnica junto com um classificador.

No trabalho de Sieler, Tanougast e Bouridane (2010), o autor afirma que a principal desvantagem do uso da GLCM é que ele exige um longo tempo de computação, especialmente para imagens muito grandes. Sendo assim, o autor propôs uma arquitetura baseada em FPGA para calcular eficientemente a matriz de coocorrência e extrair 6 atributos (ASM, Média, Variância, Correlação, Contraste e Dissimilaridade).

A simplicidade computacional e poder discriminativo do LBP o fez se tornar popular em várias aplicações com visão computacional (PIETIKÄINEN *et al.*, 2011). No trabalho de Kryjak, Komorkiewicz e Gorgon (2012) um sistema implementado em FPGA para detecção de ombros-cabeça de pedestres é apresentado. Foram utilizados o extrator de atributos LBP e SVM para o processo de classificação. O sistema final foi implementado em um Xilinx Virtex 6 FPGA, capaz de processar um fluxo de vídeo de resolução 640x480@60 fps em tempo real.

Este trabalho propõe apresentar soluções que possam vir a ser viáveis para uso na construção de classificadores embarcados em microcontroladores focado em aplicações de visão computacional, visando reduzir os custos de desenvolvimento e implantação. Portanto, serão avaliadas técnicas de extração de atributos combinada a algoritmos classificadores em comparação ao uso de algoritmos de aprendizado profundo.

3.4. INFERÊNCIA DE MODELOS DE REDES NEURAIS NA BORDA DA REDE

O aprendizado de máquina tem sido cada vez mais utilizado em soluções relacionadas a IoT e dispositivos móveis. O processamento de dados local apresenta vantagens em relação a consumo de energia, largura de banda da rede e tempo de resposta (WU *et al.*, 2019).

Segundo Wang, X. *et al.* (2020), recentemente duas plataformas ganharam destaque no que diz respeito a inferência de modelos de redes neurais na borda com baixo consumo de energia: TensorFlow Lite for Microcontrollers e STM32CubeAI. É possível encontrar também o desenvolvimento de ferramentas open-source que estimulam e facilitam o desenvolvimento de soluções que utilizam computação na borda, como é o caso dos trabalhos de Capodonti *et al.* (2020), Lai, Suda e Chandra (2018) e Wang, X. *et al.* (2020), que propõe uma ferramenta para implantação de otimização de MLPs em processadores da família ARM Cortex-M e PULP, uma família de processadores de ponta baseada na plataforma RISC-V ISA.

Em paralelo, Lai, Suda e Chandra (2018) apresentam um eficiente kernel desenvolvido para maximizar o desempenho e diminuir o uso de memória usada em aplicações que requerem o uso de redes neurais em processadores ARM Cortex-M. Essa técnica de processar modelos de redes neurais em dispositivos que possuem o consumo de energia em torno de 1mW é conhecida atualmente como TinyML, e está tornando novas aplicações possíveis (WARDEN; SITUNAYAKE, 2020).

3.5. CONSIDERAÇÕES

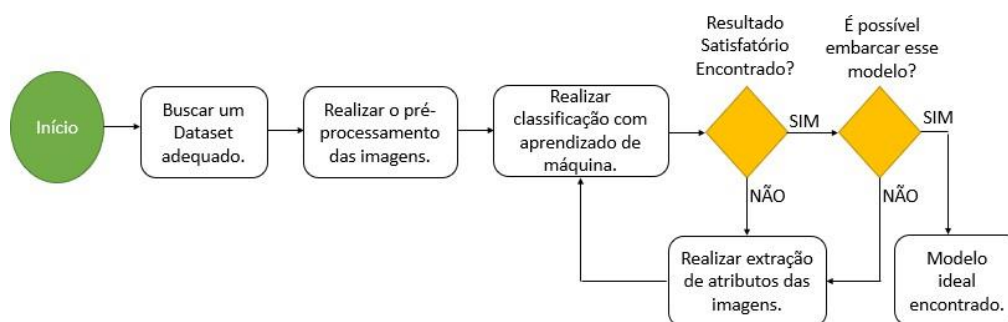
Esta dissertação focou em analisar as possibilidades de desenvolvimento de um sistema inteligente embarcado que detecte o movimento de queda e mantenha a privacidade do usuário. Foram analisadas imagens obtidas de diferentes tipos de sensores térmicos como forma de avaliar as limitações decorrentes da baixa resolução de imagem obtida, de forma a analisar a viabilidade do desenvolvimento de um sistema eficaz para esta finalidade. Assim como, pretende-se realizar o treinamento de quatro modelos de aprendizado de máquina, KNN, SVM, Bi-LSTM e VGG16, como forma de avaliar o melhor desempenho, prós e contras de cada modelo, considerando a possibilidade de ser construída uma solução embarcada.

Como forma de aproveitar as vantagens das novas tecnologias disponíveis atualmente, este trabalho avalia também a possibilidade de inferência das imagens localmente, de forma a evitar a transferência dos dados capturados para um servidor em nuvem que realizará tal ação.

4. MATERIAIS E MÉTODOS

Neste capítulo será descrito o processo adotado para obtenção de um modelo que realize a classificação das imagens térmicas, de modo a identificar um evento de queda. O processo possui a etapa de escolha do dataset, pré-processamento das imagens e posterior fluxo de aprimoramento do algoritmo de classificação, com o intuito de encontrar um modelo mais adequado a finalidade proposta, conforme fluxo apresentado na figura 13.

Figura 13 - Fluxo de procedimentos adotado na pesquisa.



Fonte: Autor

Baseado nos resultados já encontrados na literatura, as primeiras classificações serão realizadas com algoritmos de aprendizado profundo, que apresentam uma boa performance na classificação de imagens, e posteriormente será avaliado o uso de um classificador mais simples combinado a técnica de extração de atributos da imagem, visando a possibilidade de embarcar o modelo encontrado.

A implementação dos procedimentos e validação dos testes executados neste trabalho foi realizado utilizando a ferramenta web Google Colab, executado sobre uma estação de trabalho constituída de um computador Intel®Core™ i5 de 6ª geração com 2.30GHz e 8GB de RAM conectada ao Back-End do Google Compute Engine em Python 3 com 12GB de RAM.

4.1. ESCOLHA DO DATASET

A primeira fase do trabalho consistiu na pesquisa por imagens térmicas, similares a que serão capturadas pelo sensor que será utilizado em um protótipo, como forma de avaliar a viabilização do trabalho. Tais imagens foram classificadas utilizando algumas arquiteturas de classificação conhecidas, como forma de identificar a melhor estratégia a ser seguida no desenvolvimento do protótipo. A seguir será detalhado o processo realizado.

4.1.1. DATASET UTILIZADO

Baseado na dificuldade de encontrar um *dataset* apropriado para o desenvolvimento descrito em Taramasco *et al.* (2018), no qual consiste em apresentar um sistema para detecção de quedas em pessoas idosas, Riquelme *et al.* (2019) apresenta um trabalho voltado especificadamente para a montagem de 2 *datasets* consistentes que pudessem vir a ser utilizados em trabalhos futuros com abordagem similar. Tal dificuldade se dá, principalmente, por consequência de razões éticas, devido aos riscos de se submeter pessoas de idade avançada a tal experimento.

Os *datasets* disponibilizados por Riquelme *et al.* (2019) apresentam imagens consistentes devido as suas características de construção. Foram utilizados dois modelos de sensores térmicos, conforme descrito na tabela 3. O sensor MLX90640, da Melexis, com resolução de imagem de 32x24 *pixels*, e o sensor D6T-8L-06, da Omron Eletronics, com resolução de 1x8 *pixels*. Ambos sensores foram posicionados na parede, sendo o sensor MLX90640 a uma altura de 1,2m, enquanto os sensores D6T-8L-06 foram posicionados dois a 10cm do chão e outros dois a 1m, formando uma imagem de 2x16 pixel.

Tabela 3 - Descrição dos sensores utilizados para criação do dataset

Sensor	Resolução (pixel)	Frame Rate	Faixa de Temperatura de Operação	Faixa de Temperatura de Detecção	Quantidade de sensores usados
MLX90640	32x24	16 fps	-40°C a 85°C	-40° C a 300°C	1
D6T-8L-06	1x8	5fps	0° C a 50°C	-10°C a 60°C	4

Fonte: Autor

Dois grupos de voluntários participaram da captura das imagens, o grupo 01 formados por artistas performáticos que receberam instruções de como cair e o grupo 02 que não recebeu nenhuma instrução, sendo estes de gênero, idade, peso e altura distintas, conforme apresentado na tabela 4.

Tabela 4 - Características gerais dos voluntários.

Grupo 01 formado por artistas performáticos e o grupo 02 formado por pessoas normais.

	Gênero	Idade	Peso	Altura
Grupo 01	F	37	59	1.64
	F	34	51	1.62
	M	35	62	1.80
Grupo 02	F	27	49	1.52
	M	28	89	1.73
	M	29	66	1.65

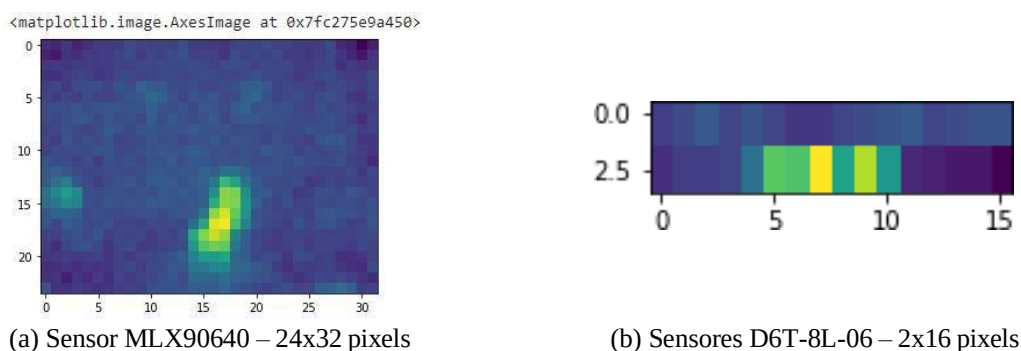
Fonte: (RIQUELME *et al.*, 2019)

Foi disponibilizado os dados brutos capturados pelos dois tipos de sensores, classificados em 15 tipos distintos de queda. O diferencial desses dados é que os mesmos foram construídos baseado em dois grupos de voluntários: um contendo pessoas normais e outro contendo artistas performáticos orientados por fisioterapeutas para simularem idosos. O dataset é formado por

180 arquivos em formato .csv, um para cada tipo de queda. Foram considerados 15 classes de movimento e 6 voluntários. A captura dos dados é realizada com o uso de 4 sensores, 2 posicionados a 10cm do chão e 2 posicionados a 1m do chão, dando uma visão lateral do ambiente.

Neste ponto da nossa pesquisa, temos o foco em conseguir desenvolver um modelo que consiga identificar a situação de queda, portanto, apenas algumas imagens foram extraídas do dataset original e separadas em duas classes: queda e não queda. A figura 14 apresenta um exemplo de imagem de cada sensor.

Figura 14 - Amostra de imagem original do dataset utilizado



Fonte: Autor

Como forma de avaliar as limitações do sistema com diferentes resoluções de imagens, foi separado uma amostra de imagens do *dataset* de cada sensor. Tais imagens passaram por um processo de geração e redimensionamento para um tamanho adequado a arquitetura que será treinada, conforme citado abaixo. Os modelos devem ser treinados utilizando os dois *datasets*, de forma distinta.

As imagens originais, geradas a partir dos dados brutos disponibilizados pelo dataset e apresentadas na figura 15, são obtidas realizando os dois primeiros passos apresentados na figura 14. Os dados obtidos através do sensor são normalizados utilizando a função `MaxAbsScaler` da biblioteca `Scikit Learn`, que normaliza os dados entre 1 e 0 dividindo-os pelo valor máximo das imagens geradas. Em seguida, utilizamos a função `get_cmap` da biblioteca `Matplotlib` para gerar os canais de cores a partir da matriz normalizada. A função retorna uma matriz de três colunas com o mesmo número de linhas que o mapa de cores da imagem original.

Figura 15 - Processo para gerar as imagens extraídas dos sensores



Fonte: Autor

Por conta da limitação do tamanho de imagem solicitada pelos modelos de classificação utilizados, as imagens passaram por processo de ampliação, tanto usando o algoritmo de `Resize`, da própria biblioteca `OpenCV`, quanto usando o algoritmo `HQX` (Filtro de ampliação de alta

qualidade). As figuras 16 e 17 apresentam os resultados obtidos através da ampliação das imagens capturadas.

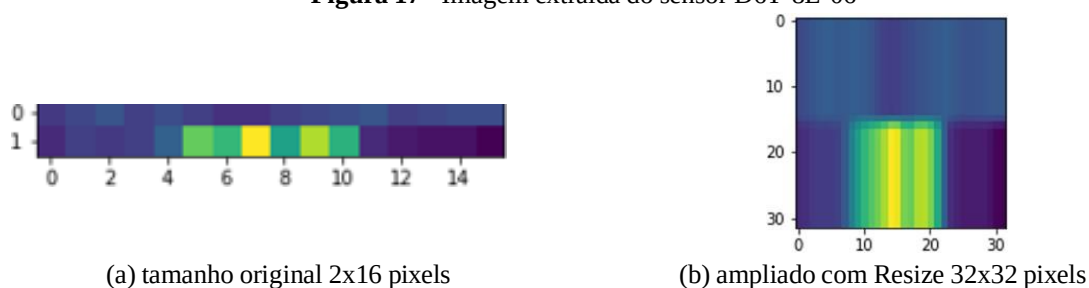
Figura 16 - Imagem extraída do sensor MLX90640



Fonte: Autor

No caso das imagens obtidas pelo sensor MLX90640, por apresentarem mais detalhes de contorno do objeto detectado, observamos que ao ser utilizado a função Resize, ocorria a perda de detalhes da imagem, principalmente do contorno do objeto detectado. Portanto, para evitar a perda de tais detalhes, visando inclusive uma melhoria dos contornos do objeto, foi utilizado o algoritmo HQX para ampliação dessas imagens.

Figura 17 - Imagem extraída do sensor D6T-8L-06

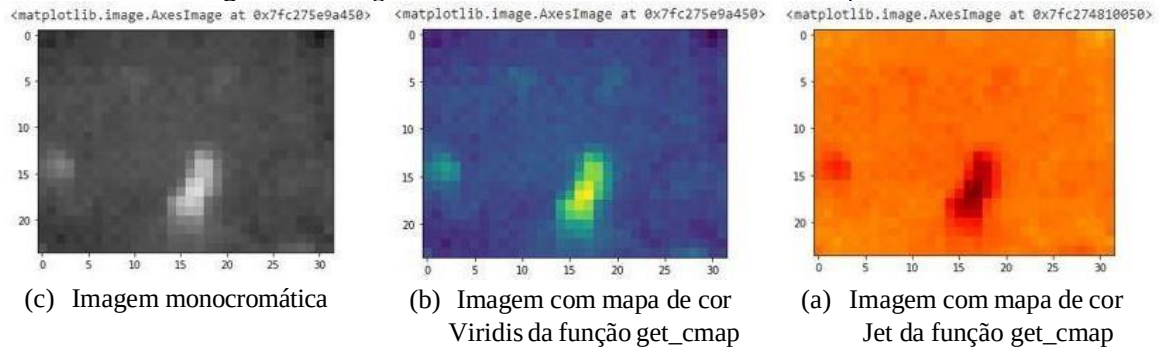


Fonte: Autor

No caso das imagens obtidas pelo sensor D6T-8L-06, observamos que as mesmas apresentam apenas 2 linhas de informação, não apresentando detalhes claros do objeto detectado, portanto, foi adotado um processo mais simples de ampliação. As imagens desse sensor tiveram suas linhas duplicadas para melhorar a simetria da imagem e, em seguida, passaram pela função Resize, da biblioteca OpenCV.

4.1.2. PRÉ PROCESSAMENTO DOS DADOS

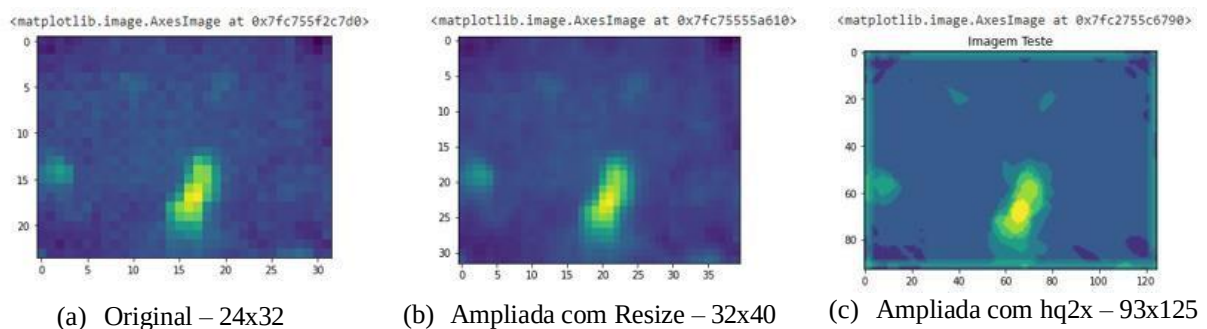
Conforme apresentado na figura 15, as imagens foram geradas a partir da normalização dos dados e geração dos canais de cores. Os canais de cores foram obtidos através da função `get_cmap` da biblioteca Matplotlib, sendo que alguns mapas de cores foram testados para avaliação das diferenças e escolha do padrão que melhor representasse os detalhes da imagem. Na figura 18 temos uma amostra de alguns mapas utilizados com as imagens originais, tamanho 24x32 pixels, do sensor MLX90640.

Figura 18 - Imagens do sensor MLX90640 em diferentes mapas de cores

Fonte: Autor

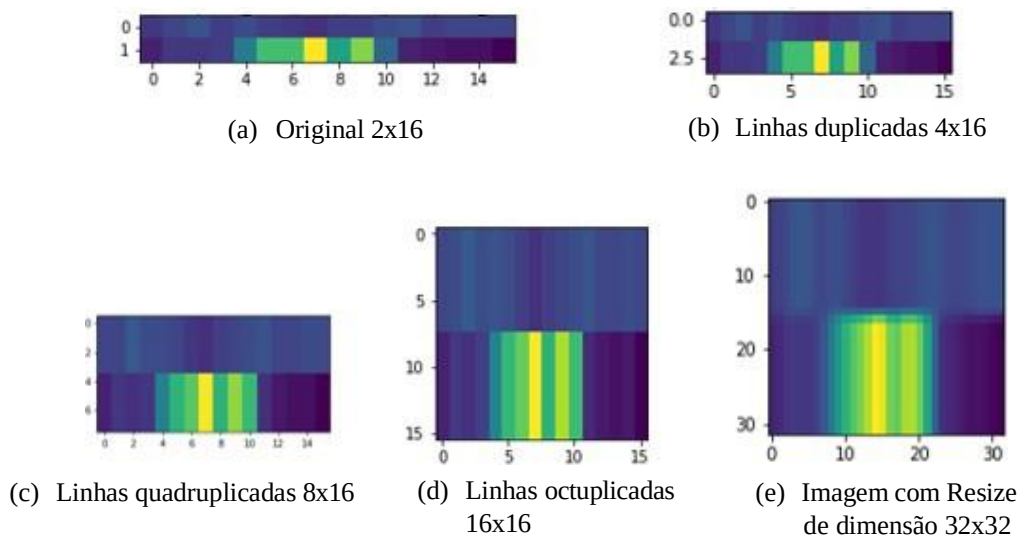
Em uma paleta de cores, as intensidades de temperaturas medidas em cada ponto da matriz passam a ser representada por uma cor específica. Com relação à interpretação da informação térmica, as paletas monocromáticas possuem a vantagem de enfatizar a geometria da informação enquanto as paletas baseadas em matiz (como a arco-íris) enfatizam a quantidade ou intensidade da informação. Essas duas paletas são conhecidas como sequenciais e podem ser vistas na figura 19 (a) e 19 (b). A figura 19 (c) apresenta a paleta Jet, que é uma paleta conhecida como divergente, e sua efetividade se baseia no fato de que nosso sistema visual pode facilmente discriminar entre diferentes cores.

Em relação ao redimensionamento, testamos a utilização de um algoritmo mais simples, que é o caso da função `Resize` da biblioteca `OpenCV`. No caso deste algoritmo, observamos que, conforme ocorria a ampliação da imagem, o contorno do elemento detectado ficava mais distorcido. Por este motivo, as imagens extraídas do sensor MLX90640, que possuem uma melhor resolução, foram ampliadas utilizando o algoritmo `hq2x`, variação do `HQX` (Filtro de ampliação de alta qualidade). Na figura 19, podemos observar a diferença das imagens geradas em relação aos algoritmos utilizados.

Figura 19 - Comparação das imagens redimensionadas do sensor MLX90640

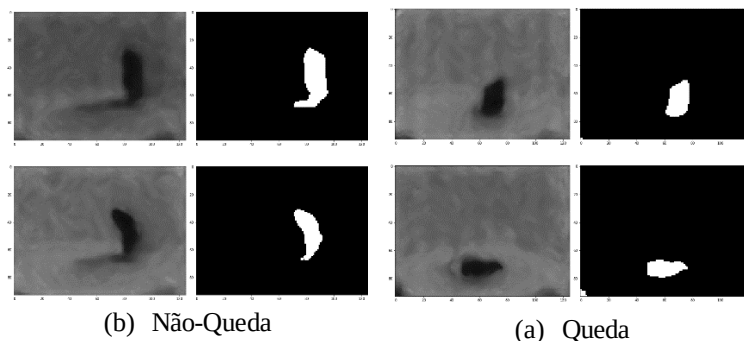
Fonte: Autor

No caso das imagens extraídas do sensor D6T-8L-06, por serem imagens com menor resolução e com um padrão mais simples, o redimensionamento foi realizado primeiramente por duplicação das suas linhas, de forma a deixar a imagem mais simétrica e, em seguida, aplicado a função `Resize` da biblioteca `OpenCV`, deixando a imagem com o tamanho mínimo recomendado para treinamento na arquitetura da VGG16, 32x32 pixels. É possível observar o resultado do processo de ampliação na figura 20.

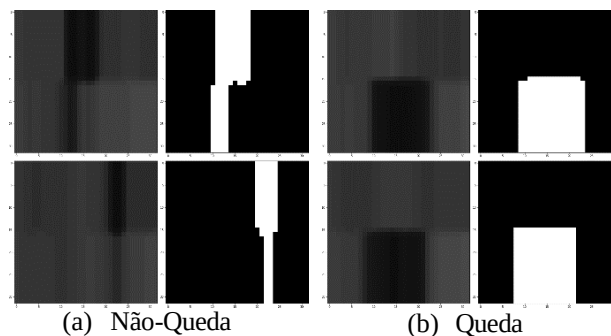
Figura 20 - Processo de ampliação das imagens extraídas do sensor D6T-8L-06

Fonte: Autor

Após do processo de redimensionamento, as imagens passaram por uma limiarização, a fim de dar destaque a região detectada com temperaturas mais elevadas, facilitando assim a classificação. Foi utilizado a função threshold com o parâmetro de binarização inversa, a fim de destacar na cor branca a região na qual foi localizada o objeto quente. As figuras 21 e 22 apresentam o resultado do processo de binarização, sendo a imagem (a) um exemplo de imagem que não apresenta queda e (b) um exemplo de imagem que apresenta a queda.

Figura 21 – Comparação de Imagens 93x125 – Original e Binarizada

Fonte: Autor

Figura 22 – Comparação de Imagens 32x32 – Original e Binarizada

Fonte: Autor

4.2. TREINAMENTO DAS ARQUITETURAS

Os primeiros experimentos de classificação foram realizados após a seleção, preparação e separação das imagens do conjunto escolhido. Por serem conhecidas por resolver problemas de classificação de imagens com altas taxas de precisão, optou por iniciar usando técnicas de aprendizado profundo.

4.2.1. APRENDIZADO PROFUNDO

Após o pré-processamento das imagens para uso, elas foram manualmente classificadas em duas categorias: ocorreu queda e não ocorreu queda. A base de dados que foi usada para treinar os dados foi composta de 2.500 imagens, distribuídas de forma equitativa entre as classes especificadas, e 20% das imagens foram usadas para validação.

A Bi-LSTM e a VGG16 foram as primeiras arquiteturas treinadas. O uso da arquitetura Bi-LSTM em trabalhos anteriores que apresentavam taxas de acurácia satisfatórias foi a motivação para realizar o treino com ela. Como a VGG16 é uma rede bem conhecida com pesos que podem ser reutilizados, otimizando o processamento da rede, foi decidido que os testes seriam realizados com a mesma técnica.

4.2.1.1. Bi-LSTM

A arquitetura do modelo conta inicialmente com uma camada de convolução, onde filtros recebem o vetor de imagens na entrada com a finalidade de extrair características hierárquicas que facilitem a classificação. Esta camada tem como resultado um vetor de 1 dimensão com tais características. Em seguida, a camada de Maxpooling recebe este vetor e reduz o número de características encontradas, deixando as mais importantes, que serão repassadas para a camada Bi-LSTM. Para finalizar, temos a camada densa, que consiste em uma rede neural do tipo perceptron usada para obter uma saída entre 0 e 1, seguida da camada de ativação. Um resumo da arquitetura do modelo é apresentado na figura 23.

Figura 23 – Resumo das camadas da rede Bi-LSTM

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, None, 32)	6176
max_pooling1d (MaxPooling1D)	(None, None, 32)	0
dropout (Dropout)	(None, None, 32)	0
bidirectional (Bidirectional)	(None, 100)	33200
dropout_1 (Dropout)	(None, 100)	0
dense (Dense)	(None, 1)	101
batch_normalization (Batch Normalization)	(None, 1)	4
activation (Activation)	(None, 1)	0

=====
Total params: 39,481
Trainable params: 39,479
Non-trainable params: 2

Fonte: Autor

A camada de convolução unidimensional conta com um kernel linear de valor 3, e extrai características dos *pixels* capturados, que são passados em forma de vetor (o tamanho do vetor de entrada é ajustado de acordo com o tamanho da imagem redimensionada). Usando esta estrutura, a próxima camada extrai informações mais precisas que serão usados para estabelecer os elementos essenciais que podem ser processados em formato temporal, reduzindo o processamento computacional que é realizado na camada recorrente. Esta camada é implementada com a função de ativação ReLU, pois a mesma possui características que lhe permitem reduzir os efeitos do *vanishing gradient*, alcançando um resultado mais rápido.

A camada de Maxpooling também conta com 1 dimensão e possui tamanho de pool 2. Como forma de obter uma melhor capacidade de generalização, esta camada reduz o número de características encontradas pela camada de convolução, deixando apenas os de interesse para a rede.

Conforme apresentado no capítulo 2, a camada Bi-LSTM possui uma estrutura similar a LSTM, porém realiza o reconhecimento das sequências temporais nas duas direções. Já a camada de ativação usa uma função *sigmoid*, considerando que, neste ponto da pesquisa, será realizado apenas uma classificação binária, identificando se houve ou não a queda.

Para compilação do modelo, como temos um problema de classificação binária, foi utilizado *binary_crossentropy* como parâmetro de perda, otimizador RMSprop e acurácia como métrica.

4.2.1.2. VGG16

Para realizar o treinamento com a arquitetura de rede da VGG16, de forma a ser possível utilizar os pesos da Imagenet, as imagens foram redimensionadas para um tamanho padrão de 64x64 *pixels* e escala de cor cinza. Com o intuito de utilizar os pesos já conhecidos da Imagenet para a VGG16, a arquitetura da rede foi importada diretamente da biblioteca do Keras.

Para possibilitar o uso do modelo pré-treinado no treinamento do modelo para o nosso problema foi preciso tornar as camadas do meio em camadas não-treináveis e remover as últimas camadas. As camadas de saída que melhor se adequem a solução do nosso problema são então inseridas no modelo ajustado. Um resumo do modelo final utilizado para treinamento é visto na figura 24.

Para compilação do modelo, como temos um problema de classificação binária, foi utilizado *binary_crossentropy* como parâmetro de perda, otimizador Adam e acurácia como métrica.

Figura 24 - Resumo da rede VGG16 adaptada

Model: "model"

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 64, 64, 3)]	0
block1_conv1 (Conv2D)	(None, 64, 64, 64)	1792
block1_conv2 (Conv2D)	(None, 64, 64, 64)	36928
block1_pool (MaxPooling2D)	(None, 32, 32, 64)	0
block2_conv1 (Conv2D)	(None, 32, 32, 128)	73856
block2_conv2 (Conv2D)	(None, 32, 32, 128)	147584
block2_pool (MaxPooling2D)	(None, 16, 16, 128)	0
block3_conv1 (Conv2D)	(None, 16, 16, 256)	295168
block3_conv2 (Conv2D)	(None, 16, 16, 256)	590080
block3_conv3 (Conv2D)	(None, 16, 16, 256)	590080
block3_pool (MaxPooling2D)	(None, 8, 8, 256)	0
block4_conv1 (Conv2D)	(None, 8, 8, 512)	1180160
block4_conv2 (Conv2D)	(None, 8, 8, 512)	2359808
block4_conv3 (Conv2D)	(None, 8, 8, 512)	2359808
block4_pool (MaxPooling2D)	(None, 4, 4, 512)	0
block5_conv1 (Conv2D)	(None, 4, 4, 512)	2359808
block5_conv2 (Conv2D)	(None, 4, 4, 512)	2359808
block5_conv3 (Conv2D)	(None, 4, 4, 512)	2359808
block5_pool (MaxPooling2D)	(None, 2, 2, 512)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 1)	2049
Total params: 14,716,737		
Trainable params: 2,049		
Non-trainable params: 14,714,688		

Fonte: Autor

4.2.2. EXTRATOR DE ATRIBUTOS COM CLASSIFICADOR

Após obtenção dos resultados de classificação dos modelos baseados em aprendizado profundo, buscou-se soluções alternativas de classificação, que demandasse menos recursos computacionais, visando a possibilidade de obter ao final do estudo uma solução embarcada.

Nesta dissertação foram usados dois descritores baseados em texturas para a extração de características. Um que usa uma técnica de extração de características de imagens digitais a partir de uma matriz de coocorrência de níveis de cinza (Gray Level Co-Occurrence Matrix, GLCM), proposta por Haralick (1979) e descrito na Seção 2.2.1 e o outro baseado em padrões binários locais (Local Binary Patterns, LBP), formulado por Ojala, Pietikainen e Harwood (1994) e apresentado na Seção 2.2.2.

Em relação aos classificadores, conforme apresentado na seção 2.3 será utilizado o algoritmo KNN (K-Nearest Neighbours) e o algoritmo SVM (Support Vector Machine). Foi adotado $k = 3$ como parâmetro do KNN e o kernel RBF como parâmetro da SVM.

4.2.2.1. PARÂMETROS GLCM

Para extrair os atributos da imagem com GLCM, foi utilizado as funções *greycomatrix* e *greycoprops* da biblioteca Skimage. Conforme descrito na seção 2.2.1., foi realizado inicialmente a extração de 6 características, sendo elas: Contraste, Dissimilaridade, Homogeneidade, ASM, Energia e Correlação. Após os testes iniciais, será utilizado apenas o Contraste e a Homogeneidade das imagens como atributos para a classificação. Os resultados serão discutidos com mais detalhes na seção 5.

Em relação a distância do pixel, foi realizado testes utilizando distancias de valor 1, 3 e 5, e observado uma variação mínima nos valores obtidos, portanto adotou-se a distância 1 como padrão. Já em relação ao ângulo, optou-se por adotar uma perspectiva circular e extrair as informações dos 4 ângulos possíveis, sendo eles: 0° , 45° , 90° e 135° , conforme apresentado na figura 4(a).

4.2.2.2. PARÂMETROS LBP

Para extrair os atributos da imagem com LBP, foi utilizado a função *local_binary_pattern* da biblioteca Skimage. Foi realizado a implementação de treino e teste com os valores 1, 2 e 3 para raio e o número de pontos observados foi considerado sempre 8x o valor do raio.

4.2.3. CLASSIFICAÇÃO E AVALIAÇÃO

Todas as arquiteturas de classificação apresentadas neste trabalho foram treinadas, testadas e validadas de acordo com os mesmos padrões, que incluem: classificação de imagens pré-processadas com Bi-LSTM e VGG16; classificar imagens pré-processadas com kNN e SVM usando as informações dos descritores GLCM e LBP.

Para o treinamento, foram usadas 2.500 imagens de cada dataset, sendo 1250 de cada classe (houve queda ou não houve queda). Essas imagens foram divididas aleatoriamente em 80% para treinamento e 20% para teste, e 50 realizações ou rodadas independentes de treinamento e teste foram realizadas.

Foi utilizada a precisão para arquiteturas de aprendizado profundo e a acurácia, precisão, recall e pontos f1 para modelos de classificação com extração de atributos como métricas de avaliação dos modelos.

5. RESULTADOS OBTIDOS

Os resultados e os experimentos realizados nesta dissertação com base nos procedimentos descritos no Capítulo 4 serão apresentados neste capítulo. Os resultados de cada modelo de classificação são discutidos separadamente, além das observações pertinentes dos escritores que os usaram.

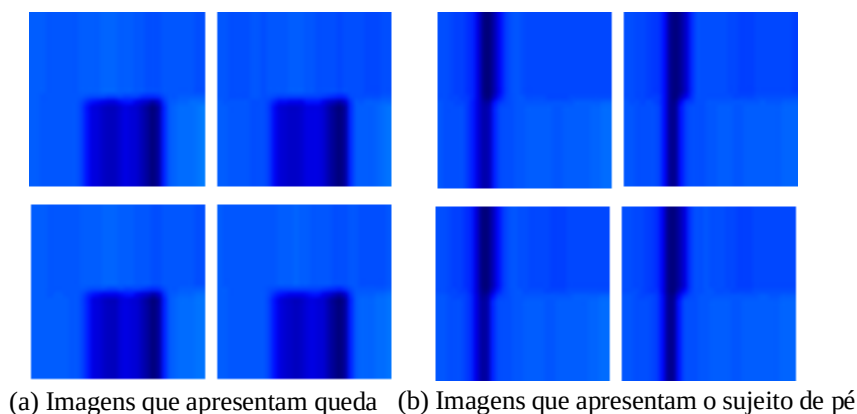
Os conjuntos de dados de imagens extraídas do sensor de maior qualidade serão apresentados como conjuntos 1 e 2, respectivamente, para facilitar o entendimento. Cada banco de dados contém 2.500 imagens, cada uma categorizada em duas categorias distintas: Queda (quando a imagem mostra movimento de queda) e Não-Queda (quando a imagem não mostra movimento de queda).

As imagens foram rotuladas como 1, se apresentam o movimento de queda, e 0, caso contrário. Portanto, a partir daqui, trataremos como classe 1 a classe de imagens que apresentam queda e classe 0 a classe de imagens que não apresentam a queda.

5.1. APRENDIZADO PROFUNDO

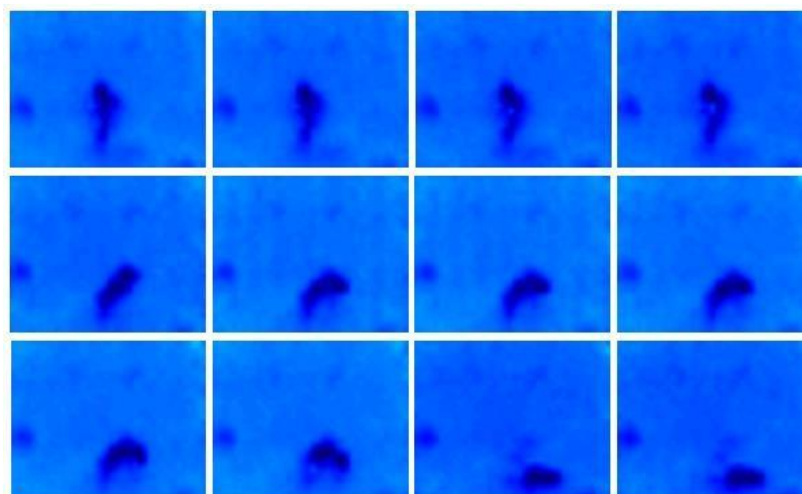
Através do processo de treinamento detalhado no capítulo anterior, foram obtidos os resultados apresentados na tabela 5. Observamos que, as imagens do dataset 2, possui uma resolução menor e formam imagens mais triviais, conforme apresentado na figura 25. É possível notar, a olho nu, que as imagens que apresentam o evento de queda apresentam, em sua maioria, o mesmo padrão: os *pixels* que preenchem a metade inferior da imagem possuem temperaturas mais elevadas do que os *pixels* que preenchem a metade superior. Dessa forma, torna-se fácil também para o modelo de aprendizado profundo detectar o fenômeno, chegando a uma acurácia média de 97,88%, quando usado a arquitetura da VGG16.

Figura 25 – Imagens extraídas do dataset 2



Fonte: Autor

No caso das imagens do dataset 1, conseguimos obter mais detalhes do ambiente, capturando detalhes do movimento e tornando as imagens mais complexas para análise, conforme visto na figura 26. Utilizando a VGG16, obtivemos a acurácia média de 87,49%. O valor mais alto de desvio padrão também aponta para uma dispersão maior dos resultados obtidos a cada época de treinamento. Acreditamos que o uso de técnicas de pré-processamento das imagens pode vir a melhorar os resultados obtidos com estas imagens.

Figura 26 - Imagens extraídas do dataset 1

Fonte: Autor

Os resultados foram observados durante o treinamento dos modelos, utilizando um dataset composto por 2500 imagens, sendo 1250 de cada classe (queda e não queda), com 20% das imagens separadas para validação. Os dados de validação foram utilizados como forma de limitar o número de épocas. Foram realizadas 50 rodadas de treinamento, sendo que a arquitetura VGG16 alcançava a convergência com um número bem menor de épocas, em torno de 100 enquanto com a Bi-LSTM ficou em torno de 500. Essa performance da VGG16 se dá em parte, devido a utilização dos pesos já treinados da Imagenet.

Tabela 5 - Resultado do treinamento das arquiteturas apresentadas

Arquitetura	Dataset Usado	Acurácia Média	Desvio Padrão	Intervalo de Acurácia
Bi-LSTM	Dataset 2	91,66%	3,95	88,45% a 97,93%
	Dataset 1	85,35%	5,98	71,75% a 98,37%
VGG16	Dataset 2	97,88%	3,11	91,65% a 100%
	Dataset 1	87,49%	6,47	74,55% a 100%

Fonte: Autor

5.2. EXTRATORES DE ATRIBUTOS

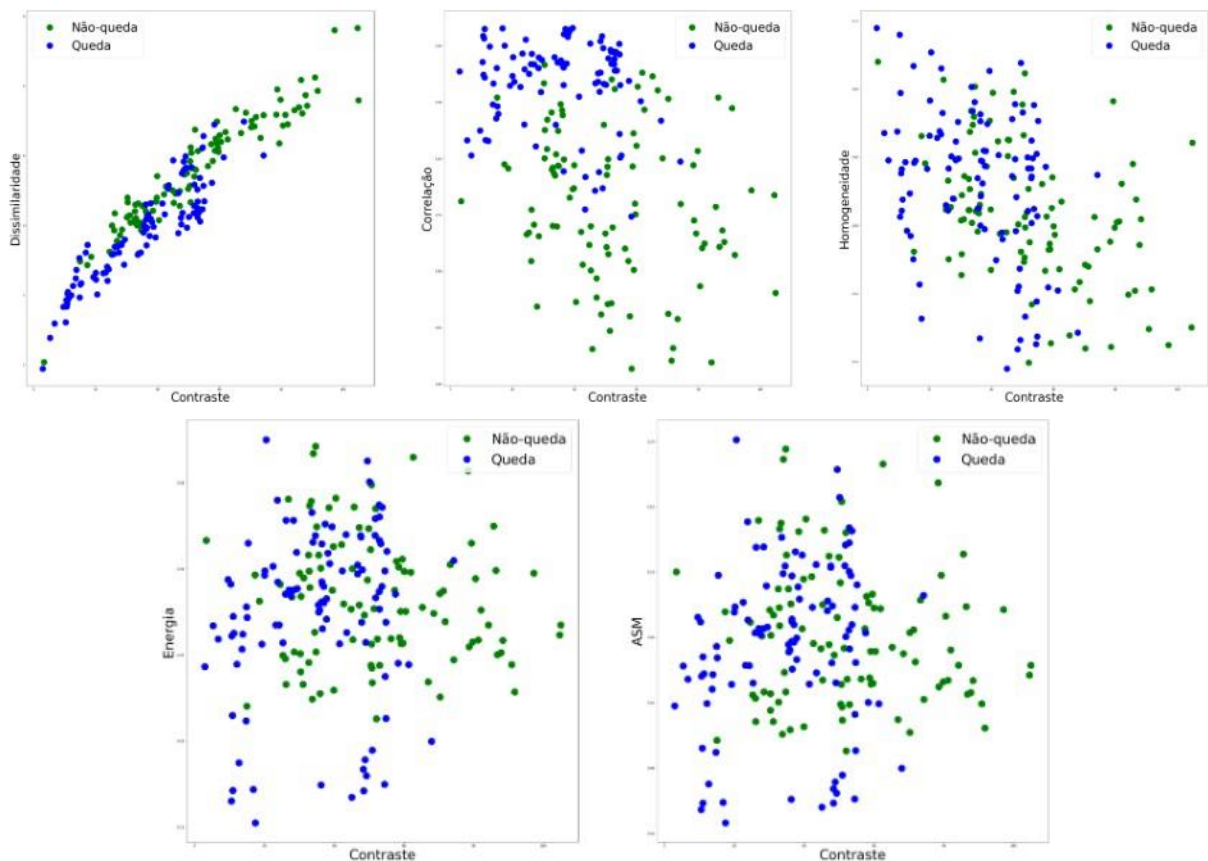
Apesar de os modelos de aprendizado profundo, que usam apenas imagens redimensionadas, apresentarem taxas de precisão elevadas, os algoritmos de extração de atributos foram usados para refinar as informações de cada banco de dados e avaliar a utilização de classificadores mais simples. Como resultado, o custo computacional do modelo final foi reduzido sem perder a qualidade dos resultados finais.

5.2.1. GLCM

Inicialmente, optou-se por obter a matriz de coocorrência com distância do pixel igual a 1, os quatro ângulos como base (0° , 45° , 90° e 135°) e 6 atributos a serem extraídos, sendo eles: Contraste, Correlação, Dissimilaridade, Homogeneidade, ASM e Energia.

Utilizando a imagem original redimensionada, observou-se que os valores extraídos através dos atributos não apresentavam uma grande distinção, conforme apresentado na figura 27, o que poderia tornar a classificação mais complexa.

Figura 27 – Distribuição de valores dos atributos extraídos da GLCM com imagem original

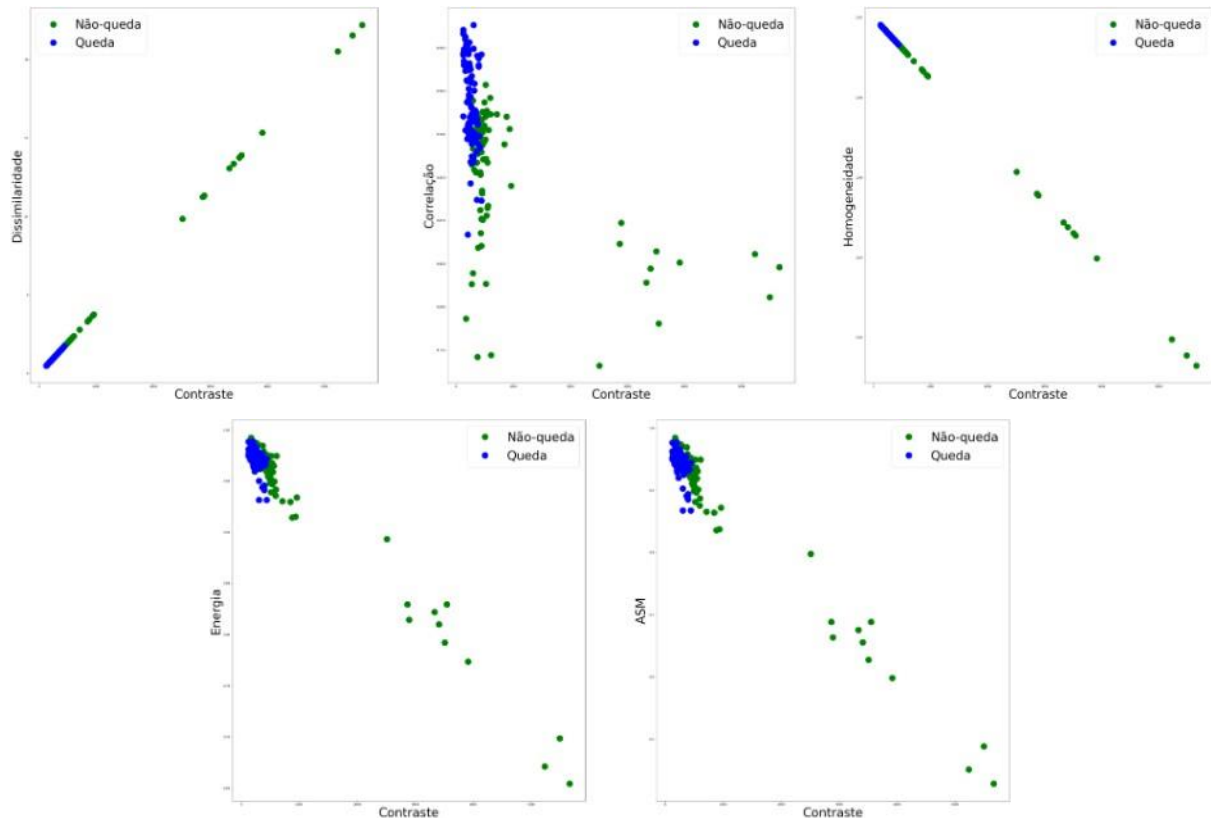


Fonte: Autor

Na figura 27 temos a distribuição dos valores da GLCM obtidas a partir de uma amostra de 100 imagens do dataset 1, sendo os pontos azuis caracterizando as imagens da classe 1 e os verdes caracterizando as imagens da classe 0. Os valores de cada atributo são apresentados em comparação ao atributo de contraste e foi observado o mesmo padrão para todas as comparações com os demais atributos.

Para tentar contornar o problema da distribuição dos valores, foi aplicado o recurso de binarização nas imagens e o cálculo da matriz GLCM foi realizado novamente. Conforme apresentado na figura 28, agora é possível verificar uma melhor distinção entre os valores obtidos com cada classe de imagem.

Figura 28 - Distribuição de valores dos atributos extraídos da GLCM com imagem binarizada

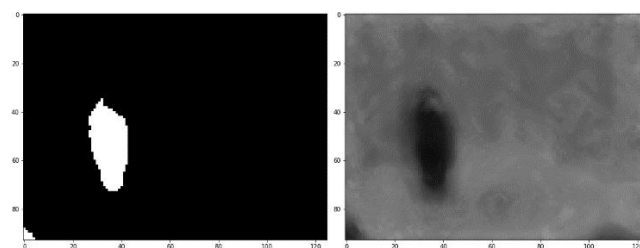


Fonte: Autor

O método de binarização inversa foi usado; os pixels com maior quantidade de calor foram selecionados para a cor branca e os outros para o preto, tornando o objeto capturado mais visível na imagem. Isso foi feito para destacar o objeto na imagem, como mostrado na Figura 29.

Com base nos resultados da distribuição dos valores da matriz de coocorrência dos seis atributos, os únicos valores usados para realizar a classificação de todas as imagens serão os de contraste e homogeneidade.

Figura 29 – Imagem binarizada em comparação com a imagem em escala de cinza



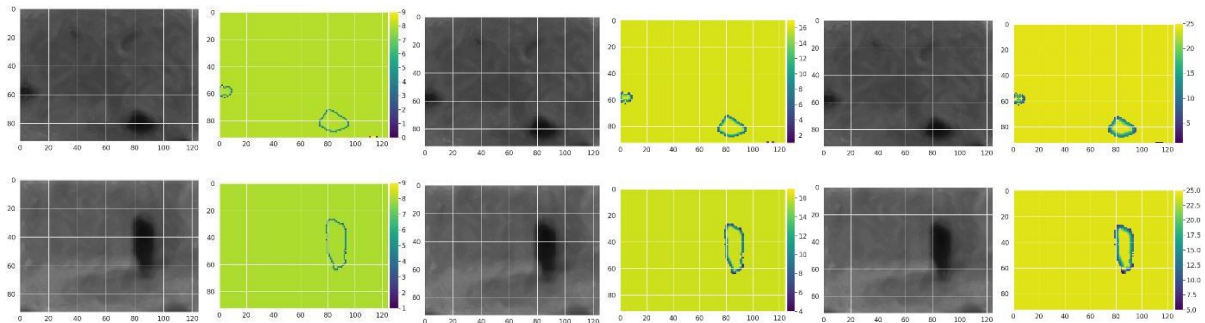
Fonte: Autor

5.2.2. LBP

Para o extrator LBP, foi realizado a extração de atributos usando raio igual a 1, 2 e 3. O resultado de cada extração foi repassado, separadamente para o classificar, a fim de analisarmos o impacto do valor do raio no resultado final. Diante disso, foi observado que quanto maior o

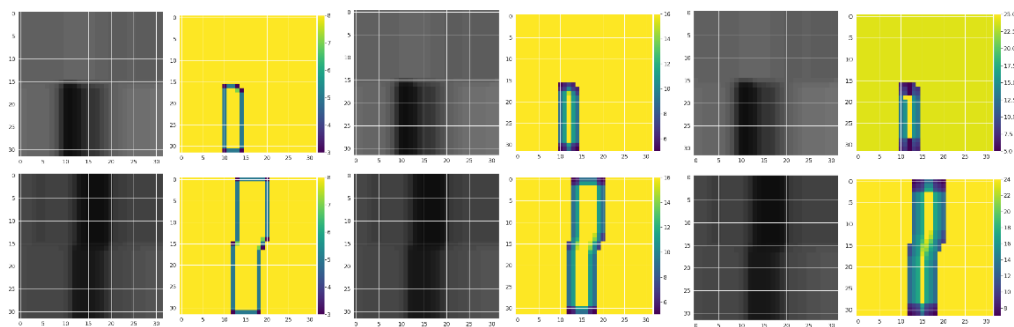
raio, maior o contorno que destaca o objeto detectado na imagem. Essa característica é observada nas figuras 30 e 31.

Figura 30 – Imagem obtida da extração LBP em comparação com a original do dataset 1 para raio 1, 2 e 3



Fonte: Autor

Figura 31 - Imagem obtida da extração LBP em comparação com a original do dataset 2 para raio 1, 2 e 3



Fonte: Autor

5.3. CLASSIFICAÇÃO COM KNN E SVM

Após a extração dos atributos, a classificação das imagens foi realizada, e os resultados encontrados nas tabelas 6 e 7 são apresentados. Para reduzir o tamanho da matriz a ser classificada, os únicos atributos utilizados na extração da GLCM foram contraste e homogeneidade. Em relação ao LPB, os raios iguais a 1, 2 e 3 foram usados, e o número de pontos foi fixado como 8 vezes o valor do raio.

Tabela 6 – Resultados Obtidos para o dataset 1

Classificador	Extrator	Raio	Acurácia Média	Desvio Padrão
KNN	GLCM	-	89,73%	1,39%
		1	93,88%	1,57%
	LBP	2	94,99%	1,24%
		3	94,84%	0,78%
SVM	GLCM	-	91,23%	1,17%
		1	97,69%	0,65%
	LBP	2	97,97%	0,50%
		3	98,20%	0,65%

Fonte: Autor

Tabela 7 – Resultados obtidos para o dataset 2

Classificador	Extrator	Raio	Acurácia Média	Desvio Padrão
KNN	GLCM	-	77,00%	4,17%
		1	95,16%	1,45%
	LBP	2	96,00%	1,62%
		3	96,58%	1,48%
SVM	GLCM	-	79,24%	3,08%
		1	96,33%	1,23%
	LBP	2	97,42%	1,29%
		3	97,03%	1,37%

Fonte: Autor

Outra métrica a ser levada em consideração para a escolha do melhor modelo são os tempos de execução de cada arquitetura. As tabelas 8 e 9 apresentam os tempos médios medidos durante a extração e classificação de cada modelo.

Tabela 8 – Tempo Médio para Extração e Classificação do dataset 1

Classificador	Extrator	Raio	Tempo Extração (ms)	Tempo Classificação (ms)
KNN	GLCM	-	15,12	3,23
		1	3,55	3,23
	LBP	2	10,62	3,23
		3	11,48	3,23
SVM	GLCM	-	12,13	13,70
		1	3,77	20,21
	LBP	2	10,07	18,65
		3	10,44	18,20

Fonte: Autor

Tabela 9 - Tempo Médio para Extração e Classificação do dataset 2

Classificador	Extrator	Raio	Tempo Extração (ms)	Tempo Classificação (ms)
KNN	GLCM	-	9,72	1,66
		1	0,52	0,32
	LBP	2	0,45	1,08
		3	0,64	1,41
SVM	GLCM	-	9,42	1,41
		1	0,46	0,59
	LBP	2	0,42	0,35
		3	0,64	0,38

Fonte: Autor

5.4. ANÁLISE DOS RESULTADOS

Os resultados dos testes realizados usando os modelos de classificação sugeridos nesta dissertação são apresentados neste capítulo. A análise incluiu dois modelos de aprendizado profundo (Bi-LSTM e VGG16) e dois modelos de extrator de característica que foram combinados com classificadores baseados em aprendizado de máquina (KNN e SVM).

As métricas de precisão e tempo de processamento foram usadas para avaliar todas as arquiteturas sugeridas. É evidente que o uso de extratores de atributos melhora a capacidade do classificador. Isso se deve ao fato de que nos modelos de aprendizado profundo, que apresentam mais camadas de abstração e, portanto, requerem arquiteturas mais robustas para executar o processo, os valores de acurácia observados são comparáveis aos das arquiteturas que usam classificadores mais simples vinculados a extratores de atributos.

A importância do pré-processamento dos dados também deve ser considerada. De acordo com os gráficos 27 e 28, é evidente a diferença que um método simples de limiarização faz no resultado final de uma classificação; isso enfatiza os pontos que realmente precisam ser observados.

Em relação aos classificadores, o uso do dataset 2 melhora o desempenho, alcançando uma taxa média de quase 98% com a arquitetura VGG16, mesmo sem realizar processos de tratamento de dados significativos. Os resultados para as arquiteturas KNN e SVM são mais semelhantes para os dois datasets, exceto quando o extrator GLCM foi usado para aplicar imagens do dataset 2. Esse extrator mostrou uma acurácia média de 77% com KNN e 79,24% com SVM, valores significativamente abaixo da média encontrada nas outras arquiteturas avaliadas.

De forma geral, os modelos de LBP com raio 3 e SVM obtiveram os valores mais precisos; os conjuntos 1 apresentaram uma taxa de precisão média de 98,20%, enquanto os conjuntos 2 apresentaram uma taxa de precisão média de 97,42% quando combinados com o LBM com raio 2.

Em relação aos tempos de execução de cada modelo, foi observado que os modelos aplicados ao conjunto de dados 2 eram mais eficientes, o que era esperado, considerando que o conjunto de dados tem imagens com resolução menor, o que significa que menos informações precisam ser processadas. Usando a método de classificação KNN combinada com o extrator LBP de raio 1 obtivemos um tempo total de cerca de 7ms para o dataset 1 e cerca de 0,8ms para o dataset 2.

6. CONSIDERAÇÕES FINAIS

A importância de projetos de automação residencial focados em melhorar a qualidade de vida, oferecendo segurança sem comprometer a privacidade e o conforto do usuário, em conjunto com a Internet das Coisas, está se tornando cada vez mais evidente com o aumento da população idosa. Esta dissertação examina a possibilidade de criar uma solução inteligente que inclua ideias de sistemas embarcados, processamento de imagem e aprendizado de máquina, que são considerados avanços no setor de cuidados de casa. Os resultados apresentados no Capítulo 5 demonstram que, com um tratamento adequado das imagens, é possível detectar quedas usando imagens térmicas de baixa resolução usando modelos de aprendizagem de máquina que requerem menos capacidade computacional. Isso abre as portas para uma solução final embarcada.

De acordo com os trabalhos apresentados no capítulo 3, observamos que vários estudos anteriores já avaliaram soluções para a identificação de queda e posterior alarme; no entanto, esses estudos incluíram questões que prejudicam o produto, como o uso de sensores vestíveis, que tornam o produto inútil quando não está fixado no paciente, ou a inclusão de câmeras que comprometem a privacidade do usuário.

Para avaliar o peso da qualidade da imagem no resultado final da solução, esta dissertação utilizou dois conjuntos de imagens térmicas com resoluções diferentes para treinar e avaliar os modelos de classificação. Com base nos resultados, pode-se concluir que os sensores de baixa resolução podem proteger a privacidade do usuário e funcionar corretamente se forem posicionados em um local estratégico para fornecer um melhor ângulo do ambiente e produzir imagens fáceis de classificação.

Os resultados foram obtidos por meio do uso de aprendizado de máquina e aprendizado profundo. A classificação com aprendizado profundo mostra uma acurácia média de 97,88%, com valores mais altos para o conjunto de dados 2. Se comparado com as imagens do dataset 1, que fornecem uma imagem mais detalhada e podem confundir o modelo, esse valor resulta da estrutura da imagem mais trivial. Por outro lado, obtivemos valores de acurácia mais próximos nas várias variações apresentadas com as técnicas de aprendizado de máquina KNN e SVM combinadas com o extrator de atributo LBP. No dataset 1, obtivemos 98,20% de acurácia com SVM e LBP de raio 3 e no dataset 2, obtivemos 97,42% de acurácia com SVM e LBP de raio 2.

Os tempos de processamento para os grupos 1 e 2 são de 7 ms e 0,8 ms, respectivamente, quando usamos LPB combinado a SVM. O desenvolvimento de um produto final inovador, simples de usar e instalar pode ajudar as pessoas mais velhas detectando quedas sem violar a privacidade dos usuários.

Referências

- A.C.SCHEFFER. Fear of falling: measurement strategy, prevalence, risk factors and consequences among older persons. **Age Ageing**, [s. l.], v. 37, n. 1, p. 19–24, 2008.
- FRIEDMAN, S.M. Falls and fear of falling: wich comes first? A longitudinal prediction model suggests strategies for primary and secondary prevention. **J. Am. Geriatr. Soc.**, [s. l.], v. 50, n. 8, p. 1329–1335, 2002.
- HAYASHIDA, Akira; MOSHNYAGA, Vasily; HASHIMOTO, Koji. New approach for indoor fall detection by infrared thermal array sensor. **Midwest Symposium on Circuits and Systems**, [s. l.], v. 2017-Augus, p. 1410–1413, 2017.
- HILDEBRANDT, Carolin; RASCHNER, Christian; AMMER, Kurt. An overview of recent application of medical infrared thermography in sports medicine in Austria. **Sensors**, [s. l.], v. 10, n. 5, p. 4700–4715, 2010.
- HRVATSKO PRIRODOSLOVNO DRUŠTVO. BIOLOŠKA SEKCIJA., KAROLJ *et al.* Periodicum biologorum. **Periodicum biologorum**, [s. l.], v. 113, n. 4, p. 407–416, 2011.
- KOPF, Johannes; LISCHINSKI, Dani. Depixelizing Pixel Art. [s. l.], v. 1, n. 212, p. 1–8, 2011.
- LIU, Zhixin *et al.* Fall Detection and Personnel Tracking System Using Infrared Array Sensors. **IEEE Sensors Journal**, [s. l.], v. 20, n. 16, p. 9558–9566, 2020.
- LOQUERCIO, Antonio *et al.* DroNet: Learning to Fly by Driving. **IEEE Robotics and Automation Letters**, [s. l.], v. 3, n. 2, p. 1088–1095, 2018.
- MARQUES FILHO, O.; VIEIRA NETO, H. **Processamento Digital de Imagens**. Brasported. São Paulo: [s. n.], 1999.
- MASHIYAMA, Shota; HONG, Jihoon; OHTSUKI, Tomoaki. Activity recognition using low resolution infrared array sensor. **IEEE International Conference on Communications**, [s. l.], v. 2015-Septe, p. 495–500, 2015.
- MOHANAIHAH, P.; SATHYANARAYANA, P.; GURUKUMAR, L. Image Texture Feature Extraction Using GLCM Approach. **International Journal of Scientific and Research Publications**, [s. l.], v. 3, n. 5, p. 1–8, 2013. Disponível em: <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e3c9c7d735c1fd8319e34fd3db37973c44802a68#page=291>.
- PANASONIC. Infrared Array Sensor Grid-EYE (AMG88). [s. l.], 2016.
- PAŠAGIĆ, Vladimir; MUŽEVIĆ, Marijan. Infrared Thermography in. [s. l.], v. 59, p. 123–130, 2008.
- PATEL, Eisha; KRISHNAN, Sridhar. Generating Stylistic Images by Extending Neural Style Transfer Method. **ACM International Conference Proceeding Series**, [s. l.], p. 19–24, 2020.
- PIETIKÄINEN, Matti *et al.* **Computer Vision Using Local Binary Patterns**. London: Springer London, 2011. (Computational Imaging and Vision). v. 40 *E-book*. Disponível em: <http://link.springer.com/10.1007/978-0-85729-748-8>.
- PULLI, By Kari; BAKSHEEV, Anatoly. OpenCV_CACM_p61-pulli. [s. l.], p. 61–69, 2012.
- RIQUELME, Fabián *et al.* Ehomeseniors dataset: An infrared thermal sensor dataset for automatic fall detection research. **Sensors (Switzerland)**, [s. l.], v. 19, n. 20, p. 1–16, 2019.
- SIAMI-NAMINI, Sima; TAVAKOLI, Neda; NAMIN, Akbar Siami. The Performance of LSTM and BiLSTM in Forecasting Time Series. **Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019**, [s. l.], p. 3285–3292, 2019.

- SIXSMITH, Andrew; JOHNSON, Neil. A smart sensor to detect the falls of the elderly. **IEEE Pervasive Computing**, [s. l.], v. 3, n. 2, p. 42–47, 2004.
- STEPIN, Maxim. **Demos & Docs – hq2x/hq3x/hq4x Magnification Filter**. [S. l.], 2003.
- TATENO, Shigeyuki *et al.* Three-Dimensional Convolutional Neural Network. [s. l.], 2020.
- UDDIN, Md Zia; KHAKSAR, Weria; TORRESEN, Jim. Ambient sensors for elderly care and independent living: A survey. **Sensors (Switzerland)**, [s. l.], v. 18, n. 7, p. 1–31, 2018.
- UNITED NATIONS, Department of Economic and Social Affairs. Population Division (2015). **World Population Ageing**, [s. l.], 2015.
- WANG, Jin *et al.* An enhanced fall detection system for elderly person monitoring using consumer home networks. **IEEE Transactions on Consumer Electronics**, [s. l.], v. 60, n. 1, p. 23–29, 2014.
- WARDEN, Pete; SITUNAYAKE, Daniel. **TinyML: machine learning with TensorFlow Lite on Arduino and ultra-low-power microcontrollers**. [S. l.: s. n.], 2019.
- WARDEN, Pete; SITUNAYAKE, Daniel. **TinyML**. [S. l.: s. n.], [s. d.].
- WEI-HAN CHEN; HSI-PIN MA. A fall detection system based on infrared array sensors with tracking capability for the elderly at home. **2015 17th International Conference on E-Health Networking, Application and Services, HealthCom 2015**, [s. l.], p. 428–434, 2015.
- WU, Carole Jean *et al.* Machine learning at facebook: Understanding inference at the edge. **Proceedings - 25th IEEE International Symposium on High Performance Computer Architecture, HPCA 2019**, [s. l.], p. 331–344, 2019.