



Universidade Federal do Piauí
Centro de Ciências da Natureza
Programa de Pós-Graduação em Ciência da Computação

Classificação de acidentes de trânsito terrestre envolvendo motocicletas através de uma rede neural local binária convolucional

Pedro Antonio Fernandes da Silva

Número de Ordem PPGCC: M001

Teresina-PI, Outubro de 2025

Pedro Antonio Fernandes da Silva

Classificação de acidentes de trânsito terrestre envolvendo motocicletas através de uma rede neural local binária convolucional

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFPI (área de concentração: Sistemas de Computação), como parte dos requisitos necessários para a obtenção do Título de Mestre em Ciência da Computação.

Universidade Federal do Piauí – UFPI

Centro de Ciências da Natureza

Programa de Pós-Graduação em Ciência da Computação

Orientador: Vinícius Ponte Machado

Coorientador: Ivan Saraiva Silva

Teresina-PI

Outubro de 2025

FICHA CATALOGRÁFICA
Universidade Federal do Piauí
Sistema de Bibliotecas UFPI - SIBi/UFPI
Biblioteca Setorial do CCN

S586c Silva, Pedro Antonio Fernandes da.
Classificação de acidentes de trânsito terrestre envolvendo motocicletas através de uma rede neural local binária convolucional / Pedro Antonio Fernandes da Silva. -- 2025.
40 f. : il.

Dissertação (Mestrado) - Universidade Federal do Piauí. Centro de Ciências da Natureza. Programa de Pós-Graduação em Ciências da Computação, Teresina, 2025.
“Orientador: Prof. Dr. Vinícius Ponte Machado, Coorientador: Prof. Dr. Ivan Saraiva Silva”.

1. Aprendizado de Máquina. 2. Rede Neural Local Binária Convolucional - LBCNN. 3. Acidente de trânsito. I. Machado, Vinícius Ponte. II. Silva, Ivan Saraiva. III. Título.

CDD 005.43


Bibliotecária: Caryne Maria da Silva Gomes - CRB3/1461

Pedro Antonio Fernandes da Silva


Classificação de acidentes de trânsito terrestre envolvendo motocicletas através de uma rede neural local binária convolucional

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFPI (área de concentração: Sistemas de Computação), como parte dos requisitos necessários para a obtenção do Título de Mestre em Ciência da Computação.


Trabalho aprovado. Teresina-PI, 10 de Outubro de 2025:

Documento assinado digitalmente
 VINICIUS PONTE MACHADO
Data: 09/12/2025 16:49:04-0300
Verifique em <https://validar.iti.gov.br>


Vinicius Ponte Machado
Orientador

Documento assinado digitalmente
 IVAN SARAIVA SILVA
Data: 09/12/2025 17:47:09-0300
Verifique em <https://validar.iti.gov.br>


Ivan Saraiva Silva
Co-Orientador

Documento assinado digitalmente
 KELSON ROMULO TEIXEIRA AIRES
Data: 11/12/2025 10:28:22-0300
Verifique em <https://validar.iti.gov.br>

Kelson Romulo Teixeira Aires
Convidado 1

Documento assinado digitalmente
 RAIMUNDO SANTOS MOURA
Data: 11/12/2025 15:30:19-0300
Verifique em <https://validar.iti.gov.br>

Raimundo Santos Moura
Convidado 2

Documento assinado digitalmente
 GERALDO BRAZ JUNIOR
Data: 14/12/2025 07:29:12-0300
Verifique em <https://validar.iti.gov.br>

Geraldo Braz Júnior
Convidado 3

Teresina-PI
Outubro de 2025

*Dedico a Nossa Senhora, a quem sou consagrado.
A ela pertence todos os meus méritos.*

Agradecimentos

Agradeço aos meus pais, Maria Rita Pereira Fernandes e Francisco Fernandes da Silva, por todo o esforço que tiveram para me educar.

Aos meus irmãos, por serem exemplo de estudo e dedicação para mim.

Agradeço ao meu orientador, Vinícius Ponte Machado, por todos os conselhos, pela paciência e pela ajuda neste período.

Assim como agradeço ao meu coorientador, Ivan,

Aos meus amigos, por todo o apoio dado.

Resumo

Observando as estatísticas de mortalidade no trânsito, a quantidade de acidentes envolvendo motocicletas e o destaque dado pela Organização Mundial da Saúde a esse tema, percebe-se que se trata de uma questão de saúde pública. Ao analisar a literatura sobre aplicações de aprendizado de máquina nesse contexto, identifica-se uma lacuna na qual este trabalho se insere.

O objetivo deste estudo é classificar imagens de acidentes envolvendo motocicletas por meio de algoritmos de aprendizado de máquina supervisionados em ambientes de computação em borda. Como a utilização de conexão pela internet nesses casos aumenta os custos de implantação e manutenção do sistema, é preferível executar os algoritmos localmente e, portanto, buscar a otimização dos recursos computacionais.

Nesse sentido, este trabalho se propôs a utilizar uma rede neural binária local (LBCNN), uma vez que essa arquitetura, além de reduzir a quantidade de parâmetros empregados em uma convolução, também apresenta propostas de otimização em hardware.

Nos experimentos realizados, destacaram-se dois modelos: a configuração LBCNN16-Bernoulli30-ResNet18, que obteve a melhor acurácia (88%), e a LBCNN16-Bernoulli50-ResNet18, com 84% de acurácia, mas com maior potencial de otimização em hardware.

Palavras-chaves: Rede Neural Local Binária Convolutiva. Acidente de trânsito terrestre. ResNet.

Abstract

Considering traffic mortality statistics, the number of motorcycle accidents, and the emphasis placed on this issue by the World Health Organization, it is clear that motorcycle accidents constitute a public health issue. An analysis of the literature on machine learning applications in this context reveals a gap that this study aims to address.

The objective of this study is to classify images of motorcycle accidents using supervised machine learning algorithms in edge computing environments. Because the use of an internet connection in such cases increases system implementation and maintenance costs, executing the algorithms locally is preferable; thus, optimizing computational resources becomes a priority.

Therefore, this study proposes the use of a Local Binary Convolutional Neural Network (LBCNN), as this architecture not only reduces the number of parameters used in convolutions but also offers hardware optimization advantages.

In the experiments conducted, two models stood out: the LBCNN16-Bernoulli30-ResNet18 configuration, which achieved the highest accuracy (88%), and the LBCNN16-Bernoulli50-ResNet18, which attained 84% accuracy but demonstrated greater potential for hardware optimization.

Keywords: Local binary convolutional neural network. Road traffic accident. ResNet.

Lista de ilustrações

Figura 1 – TLU (GÉRON, 2019).	8
Figura 2 – Problema da classificação de XOR e uma MLP que resolve. (GÉRON, 2019).	9
Figura 3 – Arquitetura de uma rede neural multicamadas de perceptrons com duas entradas, uma camada oculta de quatro neurônios e três neurônios de saída (o neurônio de viés são mostrados aqui, mas usualmente eles estão implícitos). (GÉRON, 2019).	10
Figura 4 – Exemplo de operação de convolução unidimensional com filtro de tamanho $l = 3$ e stride $s = 1$. (RUSSELL; NORVIG, 2020).	11
Figura 5 – Módulo básico de uma rede convolucional tradicional e de uma rede local binária convolucional. W_l e V_l são os pesos treináveis em cada módulo (JUEFEI-XU; BODDETI; SAVVIDES, 2017).	13
Figura 6 – Exemplo de arquiteturas de redes neurais para a base de dados ImageNet. À esquerda, o modelo VGG-19 (SIMONYAN; ZISSERMAN, 2015) para referência. No meio, uma rede convolucional simples com 34 camadas. À direita, uma rede neural residual de 34 camadas. Os atalhos pontilhados aumentam as dimensões. A Tabela 1 detalha outras variantes da ResNet.	14
Figura 7 – À esquerda, um bloco de construção para ResNet-18/34 como na Figura 6. À direita, um bloco de construção para ResNet-50/101/152.	15
Figura 8 – Atalho nas conexões entre camadas quando há alteração no tamanho e na profundidade do mapa de atributos.	15
Figura 9 – Curva ROC do identificador de acidente de trânsito proposto (SADEKY et al., 2010).	19
Figura 10 – Metodologia aplicada.	23
Figura 11 – Exemplos CADP: (a) a (i).	24
Figura 12 – Exemplos de imagens após o pré-processamento: (a), (b) e (c).	25
Figura 13 – Exemplos de imagens transformadas: (a) a (i).	26
Figura 14 – Acurácia vs Quantidade de parâmetros.	31

Lista de tabelas

Tabela 1 – Arquiteturas variantes da ResNet, com diferentes tamanhos, para a base de dados ImageNet. (HE et al., 2016).	15
Tabela 2 – Taxa de erro (%) dos conjuntos. O erro top-5 nas amostras de teste da ImageNet (HE et al., 2016).	15
Tabela 3 – Comparação das métricas de desempenho por classe para cada classificador (WAHAB; JIANG, 2019).	18
Tabela 4 – Desempenho do modelo proposto na segmentação comparado com o estado da arte para a base de dados ISIC 2018 (SALIH; VIRIRI, 2020).	20
Tabela 5 – Validação cruzada usando Faster R-CNN. Os autores usaram a métrica mAP@0.5.	21
Tabela 6 – Comparativo do estado da arte apresentado.	22
Tabela 7 – Metricas por rótulo da CNN-ResNet18.	32
Tabela 8 – Metricas por rótulo da LBCNN8-ResNet18.	32
Tabela 9 – Metricas por rótulo da LBCNN16-Bernoulli30-Resnet18.	32
Tabela 10 – Metricas por rótulo da LBCNN16-Bernoulli50-Resnet18.	32
Tabela 11 – Matriz de confusão da LBCNN16-Bernoulli50-ResNet18 (rótulo real na vertical e rótulo previsto na horizontal).	33
Tabela 12 – Matriz de confusão da CNN-ResNet18 com subconjuntos separados por vídeo (rótulo real na vertical e rótulo previsto na horizontal).	34

Lista de abreviaturas e siglas

IoT	Internet of Things
ATT	Acidente de Trânsito Terrestre
LBCNN	Local Binary Convolutional Neural Network
TLU	Threshold Logic Unit
LTU	Linear Threshold Unit
XOR	Exclusive OR
MLP	Multi-Layer Perceptron
DNN	Deep Neural Network
ReLU	Rectified Linear Unit
CNN	Convolutional Neural Network
DR	Detection Rate
FAR	False-Alarm Rate
ISIC	International Skin Imaging Collaboration
LBC	Local Binary Convolution

Sumário

Introdução	1
Motivação	1
Objetivos Gerais	2
Objetivos Específicos	2
Estrutura do Trabalho	2
I PREPARAÇÃO DA PESQUISA	5
1 REFERENCIAL TEÓRICO	7
1.1 Redes neurais	7
1.1.1 Rede neural convolucional	10
1.1.2 Rede neural local binária convolucional	12
1.1.3 Deep Residual Learning	13
1.1.4 Em Resumo	16
2 ESTADO DA ARTE	17
2.1 Aprendizado de máquina supervisionado a acidentes de trânsito terreste	17
2.2 Aprendizado de máquina supervisionado aplicado à atributos extraí- dos de imagem	18
2.3 Classificação de acidentes de trânsito terrestre por imagem	19
2.4 Aplicações de redes neurais locais binárias convolucionais	19
2.5 Deteção de objetos em acidentes de trânsito	20
2.6 Trabalhos em comparação	21
3 MATERIAIS E MÉTODOS	23
3.1 O ambiente de desenvolvimento	23
3.2 Metodologia	23
3.2.1 Seleção	24
3.2.2 Pré-processamento	25
3.2.3 Transformação	26
3.2.4 Treinamento	27
3.2.5 Avaliação	27

II	PARTE FINAL	29
4	RESULTADOS E DISCUSSÃO	31
4.1	Resultados	31
4.2	Discussão	33
	Conclusão e Trabalhos Futuros	35
4.3	Conclusão	35
	REFERÊNCIAS	37

Introdução

Como destacado pela Organização Mundial da Saúde, aproximadamente 1,19 milhões de pessoas morrem todos os anos em acidentes de trânsito ([World Health Organization, 2023](#)). A segurança no trânsito é tão importante para a saúde pública que as Nações Unidas estabeleceram metas ambiciosas para 2030: reduzir, no mínimo, em 50% as mortes e os ferimentos em acidentes ([United Nations, 2020](#)).

Entre 2015 e 2016, no Brasil, cerca de 40 mil pessoas morriam em acidentes de trânsito e, durante a pandemia, foram registradas cerca de 30 mil mortes por ano ([CARVALHO, 2021](#)). Em uma projeção realizada por [Carvalho \(2021\)](#), estima-se que um terço das mortes em acidentes de trânsito terrestre (ATT), em 2021, envolveu motocicletas.

Analisando esses dados, surge o questionamento: como os avanços tecnológicos podem ajudar a reduzir a mortalidade em acidentes de trânsito? É nesse questionamento que este trabalho se posiciona.

Motivação

Algumas contribuições já foram realizadas nesse tópico; por exemplo, [Wahab e Jiang \(2019\)](#) propuseram um algoritmo que classifica acidentes por gravidade de ferimentos; [Sadeky et al. \(2010\)](#), um framework que, em tempo real, calcula a probabilidade de ocorrer um acidente; e [Pashaei, Ghatee e Sajedi \(2020\)](#), um método para detectar acidentes e classificar sua gravidade.

Dadas suas diferentes propostas, os autores chegaram a resultados e conclusões distintos. [Wahab e Jiang \(2019\)](#) conseguiram interpretar fatores que ocasionam acidentes e ainda sugeriram políticas públicas para mitigar a gravidade da situação dos motociclistas feridos. O framework de [Sadeky et al. \(2010\)](#) mostrou-se capaz de prever acidentes; em seu artigo, os autores também sugerem seu uso na vigilância do trânsito em tempo real. [Pashaei, Ghatee e Sajedi \(2020\)](#) obtiveram um robusto e eficaz sistema de análise do trânsito que utiliza, em conjunto, diversos algoritmos para detectar acidentes, identificar gravidade e distinguir os tipos de veículos envolvidos.

Apesar de esses trabalhos já tratarem do reconhecimento de acidentes de trânsito, não há pesquisa que busque resolver esse problema em dispositivos inteligentes acoplados a câmeras, executando localmente o algoritmo sem a necessidade de delegar a tarefa a outra máquina com mais recursos computacionais. Esse é um aspecto importante, pois reduz falhas causadas pela comunicação via internet, diminui a complexidade da solução e reduz os custos de implantação e manutenção do sistema.

Como citado por [Silva \(2022\)](#), com o crescimento da quantidade de dispositivos de internet das coisas (IoT), que possuem recursos computacionais e energéticos limitados, surgiu o paradigma de computação de borda. Quando há necessidade de aplicar redes neurais artificiais em dispositivos de borda, a quantidade de recursos requeridos por esses algoritmos torna-se um problema.

Para auxiliar vítimas de ATT com motocicletas, um dispositivo de borda voltado a identificar esse cenário necessitaria de um algoritmo de aprendizado de máquina que, além de eficaz para reconhecer esse padrão, fosse eficiente no uso de memória e tempo de processamento.

Objetivo Geral

Analisando a literatura, encontrou-se o modelo de rede neural local binária convolucional (em inglês, LBCNN), proposto por [Juefei-Xu, Boddeti e Savvides \(2017\)](#). Além disso, [Lin et al. \(2021\)](#) descrevem a arquitetura de um acelerador de camada neural local binária convolucional (LBC).

Dessa forma, dada a limitação de recursos no cenário apontado e devido à existência de literatura suficiente com propostas tanto de hardware quanto de software, o objetivo deste artigo é propor a aplicação de uma LBCNN para identificar acidentes de trânsito terrestre envolvendo motocicletas. O escopo dos dados utilizados será restrito a imagens extraídas de vídeos de circuito fechado de televisão (sigla em inglês: CCTV).

Objetivos Específicos

De forma mais específica, pretende-se calcular o desempenho de uma rede neural com camadas LBC utilizando as métricas de precisão, sensibilidade e F1. Deve-se encontrar a melhor arquitetura para a LBCNN e os melhores parâmetros, além de compará-la a uma rede neural convolucional convencional.

Estrutura do Trabalho

Após esta introdução, o trabalho está estruturado da seguinte forma:

- [Capítulo 1](#) - Apresenta a pesquisa bibliográfica inicial realizada sobre redes neurais;
- [Capítulo 2](#) - Aborda o estado da arte referente à aplicação de redes neurais binárias e à identificação de ATTs;
- [Capítulo 3](#) - Descreve os algoritmos empregados e a metodologia adotada para a avaliação dos modelos de redes neurais;

- [Capítulo 4](#) - Apresenta os resultados dos testes aplicados à LBCNN, comparando seu desempenho com o de uma rede neural convolucional convencional;
- [Conclusões e Trabalhos Futuros](#) - Reúne as principais conclusões obtidas ao longo do estudo, bem como sugestões para pesquisas futuras.

Parte I

Preparação da pesquisa

1 Referencial Teórico

Este trabalho implementa uma LBCNN baseada na proposta de [Juefei-Xu, Bodeti e Savvides \(2017\)](#), conforme detalhado na [subseção 1.1.2](#), e adota uma arquitetura semelhante à proposta por [He et al. \(2016\)](#), conforme descrita na [subseção 1.1.4](#). Além dessas abordagens, outros conceitos também se mostram essenciais para a compreensão da pesquisa.

Portanto, este capítulo pretende explicar alguns conceitos importantes para o trabalho. Na [seção 1.1](#), é explicada a evolução, ao longo do tempo, das arquiteturas de redes neurais e como isso resulta no modelo implementado. São detalhadas as arquiteturas de uma rede neural profunda, de uma rede neural convolucional e de uma rede neural local binária convolucional. Além disso, ao final, apresenta-se uma arquitetura de rede neural convolucional considerada estado da arte.

1.1 Redes neurais

Para a identificação dos ATTs mencionados, o modelo proposto utiliza camadas LBC que, juntamente com camadas de outras arquiteturas, compõem uma rede neural local binária convolucional.

O componente mais básico de uma LBCNN, assim como de outras arquiteturas, é o chamado neurônio artificial. O desenvolvimento de neurônios artificiais começou com [McCulloch e Pitts \(1943\)](#), que, inspirados pelo funcionamento do neurônio biológico, propuseram, matematicamente, o primeiro modelo de um neurônio.

Continuando o trabalho desses autores, [Rosenblatt \(1957\)](#) propôs uma arquitetura de rede neural artificial chamada Perceptron. Como explicado por [Géron \(2019\)](#), essa proposta é composta por unidades denominadas TLU (do inglês *threshold logic unit*) ou LTU (do inglês *linear threshold unit*).

Como mostrado na [Figura 1](#), essa unidade calcula uma soma ponderada das entradas e aplica uma função de ativação — neste caso, a função degrau, detalhada na [Equação 1.2](#) ([GÉRON, 2019](#)). O valor a ser aplicado à função de ativação é calculado conforme declarado na [Equação 1.1](#). Dado que existem n entradas, $X_i \in \mathbf{x}$ representa a entrada i , $W_i \in \mathbf{w}$ é o peso a ser multiplicado por essa entrada, e W_0 é um peso que não necessita de entrada correspondente (denominado viés).

$$z = W_0 + \sum_{i=1}^n X_i \cdot W_i \quad (1.1)$$

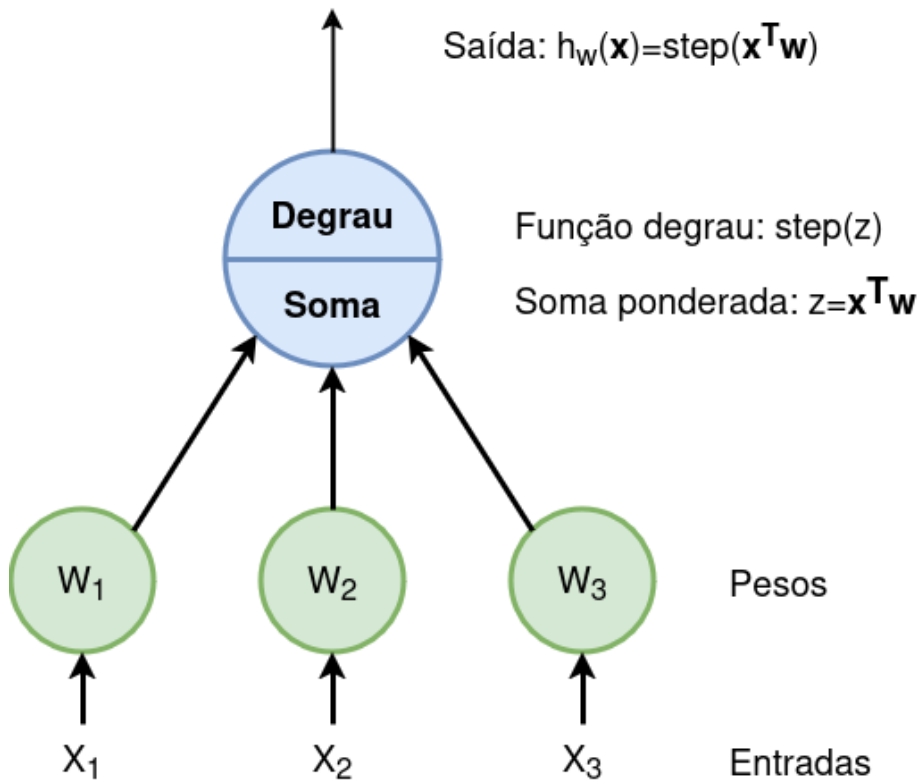


Figura 1 – TLU (Géron, 2019).

$$h_{\mathbf{w}}(\mathbf{x}) = \begin{cases} 1 & \text{if } z \geq \theta \\ 0 & \text{if } z < \theta \end{cases} \quad (1.2)$$

Entretanto, o Perceptron é capaz de construir soluções apenas para problemas linearmente separáveis (MINSKY; PAPERT, 1969). Como exemplificado na Figura 2, com o problema da porta lógica XOR, essas soluções só são encontradas com um Perceptron de múltiplas camadas, conhecido como MLP (do inglês multi-layer perceptron).

A arquitetura de uma MLP resume-se a uma camada de entrada da rede neural, uma ou mais camadas ocultas compostas de unidades TLU e uma camada final de TLUs que produzem a saída (Géron, 2019). Quando uma MLP possui muitas camadas ocultas, ela é frequentemente chamada de rede neural profunda (DNN, do inglês Deep Neural Network) (Géron, 2019).

A Figura 3 exemplifica a arquitetura de uma MLP denominada Feedforward. Como o nome sugere, uma rede neural feedforward é um modelo composto por um grafo acíclico, com conexões em uma única direção, que se iniciam nos nós de entrada e terminam nos nós de saída Figura 3.

Rumelhart, Hinton e Williams (1986) propuseram a retropropagação do erro como método de aprendizagem que ajusta os pesos das conexões em uma rede neural, chamado Backpropagation, pois essa é “a forma como o erro na saída é transmitido de volta pela

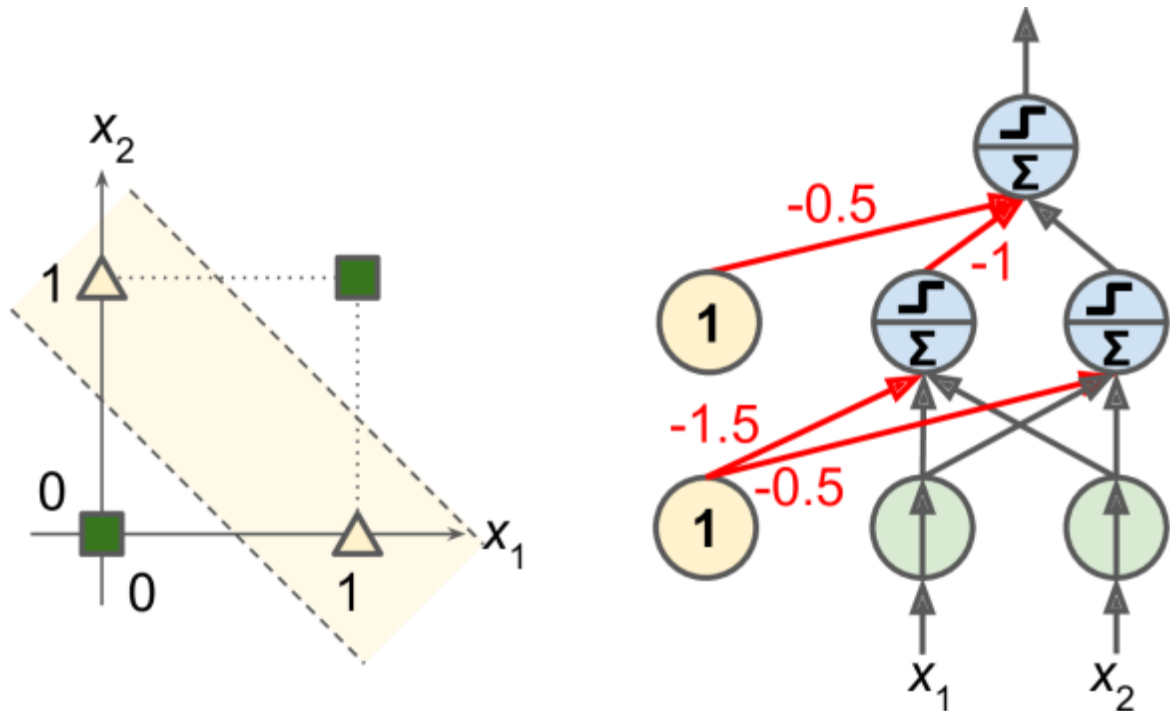


Figura 2 – Problema da classificação de XOR e uma MLP que resolve. (GÉRON, 2019).

rede” Figura 3. Esse método é utilizado atualmente em redes neurais profundas e em outros modelos.

Atualmente, várias funções de ativação além da função degrau foram propostas; destacam-se as funções ReLU (do inglês Rectified Linear Unit) e Softmax, detalhadas, respectivamente, nas Equação 1.3 e Equação 1.4. A função ReLU tornou-se bastante popular devido à sua capacidade de evitar o problema do desvanecimento dos gradientes (RUSSELL; NORVIG, 2020). Em contrapartida, a função Softmax ganhou popularidade por sua melhor adaptação a problemas de classificação envolvendo múltiplos rótulos, pois é capaz de gerar d saídas cujas probabilidades somam 1 (RUSSELL; NORVIG, 2020). Essas funções de ativação são utilizadas nos modelos implementados neste trabalho, pois são utilizadas na LBCNN proposta.

$$ReLU(z) = \max(0, z) \quad (1.3)$$

$$softmax(in)_k = \frac{e^{in_k}}{\sum_{k'=1}^d e^{in_{k'}}} \quad (1.4)$$

Assim como ocorre nas DNNs, as LBCNNs e outras propostas possuem camadas semelhantes à arquitetura de um Perceptron. Essas camadas também são chamadas de camadas densas, pois todos os neurônios de uma camada têm conexão com todas as saídas da camada anterior ou com as entradas do modelo. Os algoritmos Backpropagation e Feedforward são comuns a essas arquiteturas, assim como as funções de ativação.

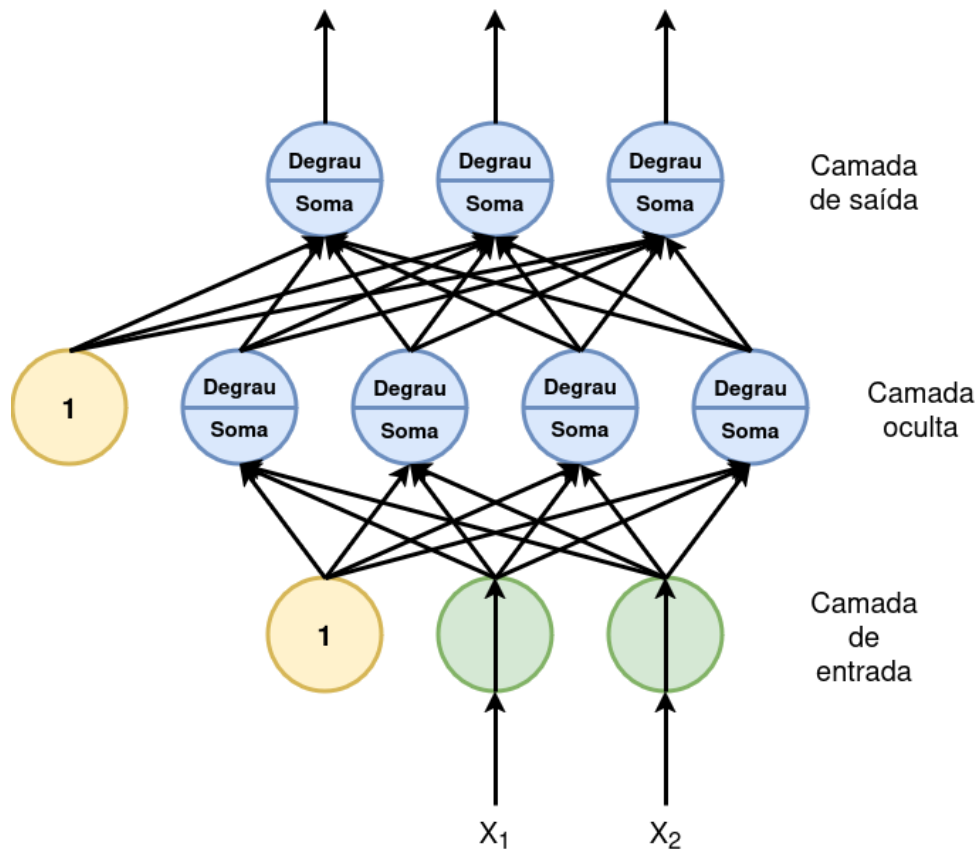


Figura 3 – Arquitetura de uma rede neural multicamadas de perceptrons com duas entradas, uma camada oculta de quatro neurônios e três neurônios de saída (o neurônio de viés são mostrados aqui, mas usualmente eles estão implícitos). (GÉRON, 2019).

1.1.1 Rede neural convolucional

O funcionamento de uma LBCNN é uma evolução das arquiteturas de redes neurais convolucionais. Por sua vez, uma rede neural convolucional (CNN, do inglês Convolutional Neural Network) é aquela que possui camadas de convolução em sua composição. Elas se caracterizam por apresentar conexões espaciais locais; isto é, um neurônio não se conecta a todas as entradas na camada inicial nem a todos os neurônios da camada anterior (GÉRON, 2019). As conexões são delimitadas por um padrão de pesos replicados entre as unidades de uma camada e deslocados sobre a entrada, alterando-se a região de interesse em cada neurônio. Esse padrão de pesos é chamado kernel (ou filtro); o deslocamento na entrada é denominado stride; e esse processo é chamado convolução (RUSSELL; NORVIG, 2020).

Para um problema unidimensional, com um vetor de entrada \mathbf{x} de tamanho n e um filtro \mathbf{k} de tamanho l , a operação de convolução, escrita como $z = x * w$, é formalizada por Russell e Norvig (2020) na Equação 1.5. A saída z na posição i corresponde ao produto escalar do filtro com um subconjunto de \mathbf{x} , centrado em x_i e de largura l .

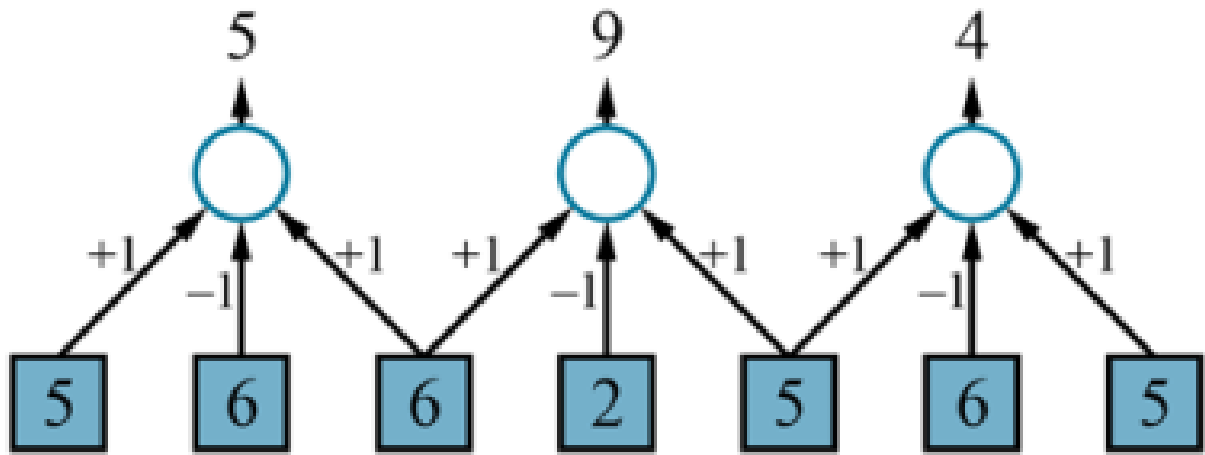


Figura 4 – Exemplo de operação de convolução unidimensional com filtro de tamanho $l = 3$ e stride $s = 1$. (RUSSELL; NORVIG, 2020).

$$z_i = \sum_{j=1}^l k_j x_{j+i-\frac{l+1}{2}} \quad (1.5)$$

Visualmente, a operação ocorre conforme ilustrado na Figura 4. Com stride $s = 2$, observa-se certa redução na largura da saída em relação à entrada. Esse efeito pode ser mitigado adicionando-se valores às bordas da entrada, por exemplo, inserindo zeros nas extremidades; essa técnica é chamada zero padding (GÉRON, 2019). Existem ainda outras estratégias possíveis.

Géron (2019) resume todo o cálculo de um neurônio de uma camada convolucional bidimensional l na Equação 1.6, considerando:

- $z_{i,j,k}$ é valor do mapa de atributos no índice i, j gerado pelo filtro k ;
- s_h e s_w são strides vertical e horizontal, respectivamente;
- f_h e f_w são as dimensões do subconjunto de interesse na entrada e $f_{n'}$ a quantidade de canais de entrada;
- $x_{i',j',k'}$ são valores localizados na camada $l - 1$, no índice i', j' e canal de entrada k' ;
- b_k é o viés do mapa de atributos k da camada l ;
- $w_{u,v,k',k}$ é o peso da conexão de um neurônio qualquer para o mapa de atributos k , camada l , entradas de índice u, v e canal de entrada k' .
- $z_{i,j,k}$: valor do mapa de atributos no índice i, j gerado pelo filtro k ;
- s_h e s_w : strides vertical e horizontal, respectivamente;

- f_h e f_w : dimensões do subconjunto de interesse na entrada e $f_{n'}$ a quantidade de canais de entrada;
- $x_{i',j',k'}$: valor localizado na camada $l - 1$, no índice i', j' e canal de entrada k' ;
- b_k : viés do mapa de atributos k da camada l ;
- $w_{u,v,k',k}$: peso da conexão entre qualquer neurônio da camada l , nos índices de entrada (u, v) e canal k' , e o mapa de atributos k .

$$z_{i,j,k} = b_k + \sum_{u=0}^{f_h-1} \sum_{v=0}^{f_w-1} \sum_{k'=0}^{f_{n'}-1} x_{i',j',k'} \times w_{u,v,k',k} \quad \text{onde} \quad \begin{cases} i' = i \times s_h + u \\ j' = j \times s_w + v \end{cases} \quad (1.6)$$

1.1.2 Rede neural local binária convolucional

A camada LBC implementada é descrita por [Juefei-Xu, Boddeti e Savvides \(2017\)](#), conforme exposto na [Figura 5](#). O módulo consiste em m filtros convolucionais fixos e pré-definidos, chamados de pesos âncora. Mapas intermediários de atributos são obtidos por meio da aplicação desses filtros sobre a entrada; em seguida, tais mapas são submetidos a uma função de ativação não linear. Ao final, os m mapas intermediários são linearmente combinados, por meio de pesos treináveis, para gerar a saída.

Como os pesos âncora são parâmetros fixos e não treináveis pela retropropagação do erro, eles devem ser definidos previamente, de forma determinística ou estocástica. No [Capítulo 3](#), explica-se como esses pesos serão definidos.

Os autores também expressaram essa operação por meio de uma fórmula matemática, conforme exposto na [Equação 1.7](#). Nessa função, t representa o canal de saída; s , o canal de entrada; $b_i, i \in [m]$ são os pesos âncora; x_l é a entrada da camada t ; e $v_{l,i}, i \in [m]$ são os pesos treináveis da camada l .

Podemos simplificar a camada representando-a com duas subcamadas, separadas por uma função de ativação. A primeira subcamada executa uma convolução com m filtros cujos valores são ternários (-1, 0 e 1). Pesos iguais a zero não têm impacto na operação de acumulação que ocorre durante a convolução, pois o produto desse valor é sempre nulo; por isso, ela é considerada uma rede neural binária. A segunda subcamada executa uma convolução adicional com filtros de dimensão 1×1 e valores em ponto flutuante.

$$x_l^t = \sum_{i=1}^m \sigma \left(\sum_s b_i^s * x_l^s \right) \cdot v_{l,i}^t \quad (1.7)$$

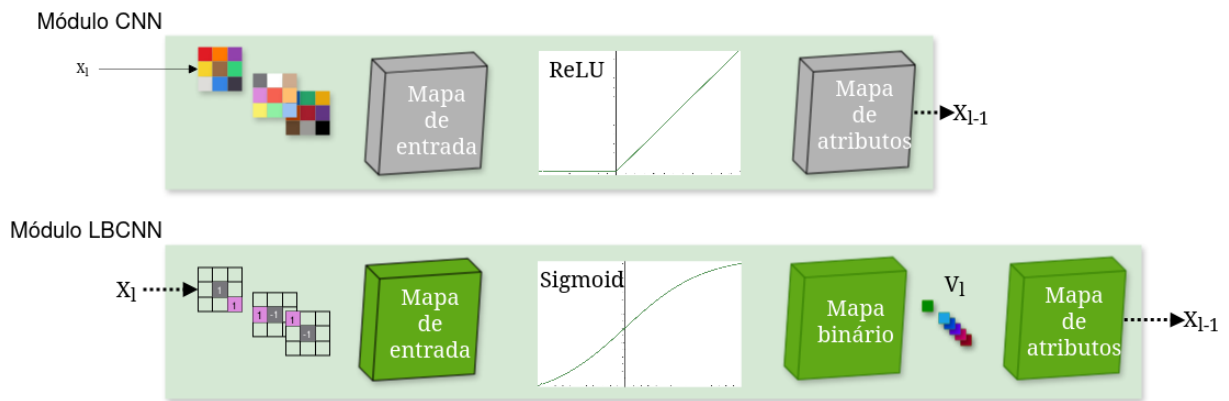


Figura 5 – Módulo básico de uma rede convolucional tradicional e de uma rede local binária convolucional. W_i e V_i são os pesos treináveis em cada módulo (JUEFEI-XU; BODDETI; SAVVIDES, 2017).

1.1.3 Deep Residual Learning

Proposta por He et al. (2016), a rede residual (ResNet, do inglês residual network) é um modelo de rede neural profunda que reformula as camadas para aprender funções residuais, em vez de funções não referenciadas. Essa estratégia torna a rede neural mais fácil de otimizar e permite alcançar maior precisão com o aumento da profundidade. Um conjunto desses modelos alcançou 3,57% de erro nos testes da base de dados ImageNet. Dessa forma, os autores ficaram na primeira posição da competição ILSVRC 2015 na tarefa de classificação de imagens.

A Figura 6 exemplifica a arquitetura de uma rede residual de 34 camadas em comparação com outras arquiteturas; as camadas entre as setas constituem os blocos de construção que compõem a função residual, detalhada na Figura 7. Quando a entrada e a saída desse bloco têm as mesmas dimensões, o atalho pode ser ligado diretamente entre os dois pontos; caso contrário, como nas setas pontilhadas da Figura 6, é necessária uma camada para ajustar as dimensões, conforme detalhado na Figura 8. As variantes da ResNet apresentam diversos tamanhos, definidos pela quantidade de unidades residuais, como detalhado na Tabela 1.

A Tabela 2 mostra o erro entre o rótulo correto das imagens e os cinco rótulos mais prováveis definidos por um conjunto de modelos formados por redes ResNet, em comparação com o estado da arte de redes neurais. Os valores foram coletados no ILSVRC'15 e foram usadas as amostras de teste da base de dados ImageNet. A arquitetura proposta por He et al. (2016) mostrou-se superior aos demais competidores e aos campeões das edições anteriores, com um erro de 3,57%.

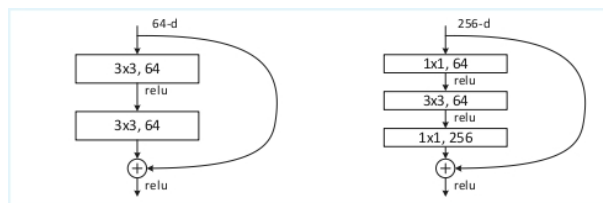


Figura 7 – À esquerda, um bloco de construção para ResNet-18/34 como na Figura 6. À direita, um bloco de construção para ResNet-50/101/152.

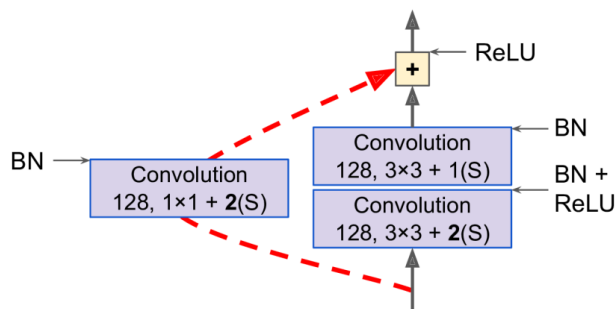


Figura 8 – Atalho nas conexões entre camadas quando há alteração no tamanho e na profundidade do mapa de atributos.

Tabela 1 – Arquiteturas variantes da ResNet, com diferentes tamanhos, para a base de dados ImageNet. (HE et al., 2016).

Camada	Tamanho da Saída	18-layer	34-layer	50-layer	101-layer	152-layer
conv1_x	112×112	$7 \times 7, 64, \text{stride } 2$				
conv2_x	conv2_x	$3 \times 3 \text{ max pool, stride } 2$				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$

Tabela 2 – Taxa de erro (%) dos conjuntos. O erro top-5 nas amostras de teste da ImageNet (HE et al., 2016).

Método	top-5 err.
VGG (SIMONYAN; ZISSERMAN, 2015) (ILSVRC'14)	7.32
GoogLeNet (SZEGEDY et al., 2015) (ILSVRC'14)	6.66
VGG (SZEGEDY et al., 2015) (v5)	6.8
PReLU-net (HE et al., 2015)	4.94
BN-inception (IOFFE; SZEGEDY, 2015)	4.82
ResNet (ILSVRC'15)	3.57

1.1.4 Em Resumo

O modelo proposto, detalhado no [Capítulo 3](#), consiste em uma ResNet-18 conforme descrita na [Tabela 1](#), mas utiliza camadas LBCNN nas convoluções. O funcionamento das camadas densas é o mesmo apresentado na [seção 1.1](#), assim como o das funções Softmax e ReLU. As camadas LBC foram implementadas conforme a proposta da [subseção 1.1.2](#) e constituem uma variação das camadas convolucionais explicadas na [subseção 1.1.1](#).

2 Estado da Arte

Este trabalho propõe uma LBCNN para identificar, a partir de imagens, ATTs envolvendo motocicletas. Muitos trabalhos anteriores se propuseram a resolver questões relacionadas a esse tópico, como exposto nas seções a seguir. São apresentadas diversas abordagens para identificação de acidentes de trânsito terrestre, assim como aplicações de redes neurais binárias em outros problemas e desenvolvimentos recentes e relevantes de aceleradores de camadas neurais.

Para estudar o estado da arte, foi realizada uma pesquisa sistemática nos seguintes tópicos: aprendizado de máquina supervisionado aplicado a acidentes de trânsito terrestre; aprendizado de máquina supervisionado aplicado a atributos extraídos de imagens; classificação de acidentes de trânsito terrestre por imagem; aplicações de redes neurais convolucionais locais binárias; e detecção de objetos em acidentes de trânsito.

Nos fichamentos, os trabalhos foram descritos com base nas características dos dados utilizados, nas propostas metodológicas apresentadas e nas formas de avaliação dos resultados. Para que um artigo fosse considerado relevante, os critérios incluíram: detalhamento da base de dados; descrição da metodologia de tratamento dos dados; especificação do modelo utilizado (arquitetura, parâmetros, hiperparâmetros etc.); avaliação dos resultados por meio de mais de uma métrica; e apresentação de resultados minimamente relevantes (por exemplo, acurácia superior a 60%).

Os trabalhos foram classificados segundo os seguintes critérios, em ordem de relevância: número de referências ao artigo; ano de publicação (com prioridade para os mais recentes); resultados obtidos; diversidade nas formas de avaliação; diversidade dos modelos comparados; e destaque específico para acidentes envolvendo motocicletas.

2.1 Aprendizado de máquina supervisionado a acidentes de trânsito terrestre

A pesquisa sistemática do estado da arte de aplicações de aprendizado de máquina supervisionado aplicadas a ATT revelou inúmeras propostas. Entre elas, [Santos et al. \(2021\)](#) propuseram identificar fatores de risco para acidentes de trânsito utilizando distintos algoritmos de aprendizado; [Santos et al. \(2024\)](#), [Zhang et al. \(2022\)](#), [Rezapour, Mehrara Molan e Ksaibati \(2020\)](#) e [Wahab e Jiang \(2019\)](#) compararam diversos algoritmos na classificação da gravidade de acidentes; e [Kuo et al. \(2018\)](#) utilizaram algoritmos supervisionados para prever a mortalidade em acidentes com motocicletas.

Tabela 3 – Comparação das métricas de desempenho por classe para cada classificador (WAHAB; JIANG, 2019).

Classificador	Precisão	Sensibilidade	Área sobre a curva	Acurácia
Árvore de Decisão	0.755	0.736	0.896	73.64
Floresta aleatória	0.757	0.739	0.902	73.91
k-ésimo vizinho mais próximo	0.755	0.739	0.902	73.91
Regressão logística multinomial	0.755	0.737	0.902	73.71

Dentre esses trabalhos, destaca-se o trabalho de Wahab e Jiang (2019). Os autores utilizaram uma base de dados com relatórios de acidentes classificados em quatro níveis de gravidade de ferimentos em motociclistas: fatal, hospitalizado, com ferimentos mas sem hospitalização e sem ferimentos. Como mostrado na Tabela 3, foram comparados os desempenhos dos algoritmos árvore de decisão, floresta aleatória, k-ésimo vizinho mais próximo e regressão logística multinomial.

2.2 Aprendizado de máquina supervisionado aplicado à atributos extraídos de imagem

De forma semelhante a este trabalho, Sadeky et al. (2010) propuseram a criação de um framework para reconhecimento de acidentes de trânsito em tempo real. Eles aplicaram regressão logística sobre atributos extraídos de imagens de acidentes de trânsito terrestre. Os autores alcançaram bons resultados, como mostrado na curva ROC (Receiver Operating Characteristic) da Figura 9. A curva ROC mostra a relação entre a taxa de detecção de acidentes e a proporção de falsos alarmes, explicada pelos autores nas Equações 2.1 e 2.2. A curva indica 99,6% de reconhecimento de acidentes com 5,2% de falsos alarmes.

$$DR = \frac{\#acidentes\ corretamente\ identificados}{total\ \# de\ acidentes} * 100\% \quad (2.1)$$

$$FAR = \frac{\#acidentes\ identificados\ erroneamente}{total\ \# de\ acidentes} * 100\% \quad (2.2)$$

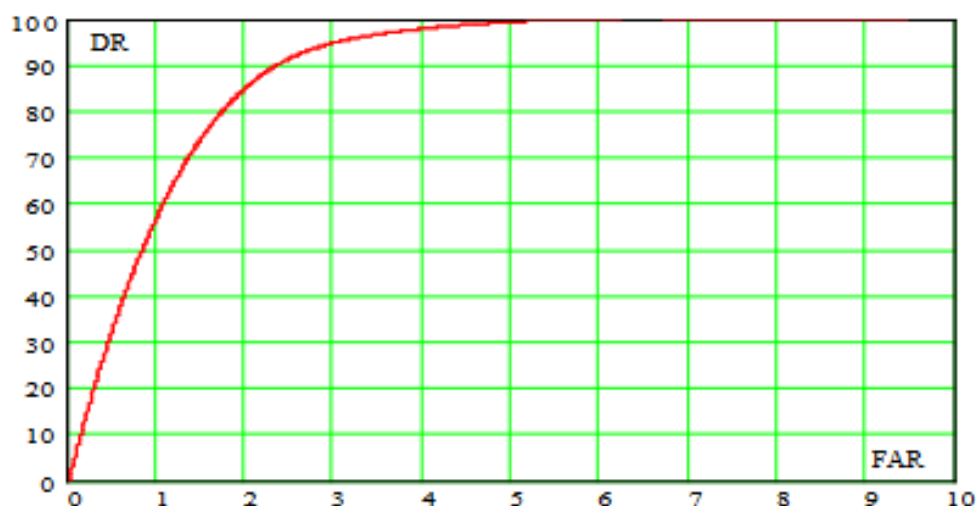


Figura 9 – Curva ROC do identificador de acidente de trânsito proposto (SADEKY et al., 2010).

2.3 Classificação de acidentes de trânsito terrestre por imagem

Utilizando imagens para classificação de acidentes, há trabalhos como o de [Tmagusko et al. \(2022\)](#), que desenvolveram duas opções de modelos de rede neural para distinguir cenários com e sem acidentes a partir de imagens sintetizadas. Já [Ravindran, Viswanathan e Rangaswamy \(2016\)](#) utilizaram máquinas de vetores de suporte para classificar a gravidade dos danos em carros.

Em destaque, [Pashaei, Ghatee e Sajedi \(2020\)](#) testaram e compararam diversos métodos para extração de atributos em imagens e diversos modelos de aprendizado de máquina para classificar imagens de acidentes. Ao final, o sistema composto por uma CNN para extração de atributos e uma composição de diversos algoritmos de aprendizado de máquina conseguiu 99.31% de precisão na detecção de acidentes, 90.27% na classificação de gravidade e 92.73% na classificação de veículos.

2.4 Aplicações de redes neurais locais binárias convolucionais

Dentre as aplicações de redes neurais binárias, destaca-se a proposta de [Salih e Viriri \(2020\)](#). Os autores também utilizaram camadas LBC, porém para a detecção de melanoma. O estudo compara o desempenho da rede neural implementada com diversas outras arquiteturas na segmentação de lesões cutâneas. Entre os testes realizados com diversas bases de dados, apresentam-se os resultados para a base de dados ISIC 2018 ([TSCHANDL; ROSENDAHL; KITTLER, 2018](#)) ([CODELLA et al., 2019](#)), exemplificados na [Tabela 4](#).

Tabela 4 – Desempenho do modelo proposto na segmentação comparado com o estado da arte para a base de dados ISIC 2018 (SALIH; VIRIRI, 2020).

Método	Acurácia	Coefficiente Jac-card	Coefficiente DICE
U-net (RONNEBERGER; FISCHER; BROX, 2015)	90.14	61.64	76.27
FCN (LONG; SELHAMER; DARRELL, 2015)	92.72	72.17	83.83
CDNN (YUAN; CHAO; LO, 2017)	93.40	76.50	84.90
mFCN-PI (BI et al., 2017)	95.51	84.64	91.18
FrCN (AL-MASNI et al., 2018)	94.03	77.11	87.08
Ensemble (GOYAL et al., 2020)	94.10	79.30	87.10
MS-Unet (TANG et al., 2019)	95.87	85.3	91.47
LBCDN (SALIH; VIRIRI, 2020)	97.09	88.43	92.53

2.5 Detecção de objetos em acidentes de trânsito

Shah et al. (2018) tiveram o propósito de “cobrir” a lacuna no acesso a bases de dados públicas de acidentes de trânsito com anotações temporais e espaciais. Para isso, criaram uma nova base de dados para análise, denominada CADP (sigla em inglês para Car Accident Detection and Prediction).

Os autores propuseram um processo para coletar os dados. O motor de busca do YouTube foi explorado para extrair vídeos com acidentes de trânsito publicados na plataforma por usuários de todo o mundo; esses vídeos foram anotados manualmente por membros da equipe de pesquisa e colaboradores contratados para a construção da base. A base foi desenvolvida com os propósitos de segmentação temporal, detecção de objetos, rastreamento, detecção de colisão de veículos e previsão de acidentes.

A base de dados é composta por 230 vídeos, cada um contendo, no mínimo, um acidente capturado por câmera fixa, totalizando 1416 segmentos de acidentes de trânsito. Os autores selecionaram 205 desses segmentos para anotação espacial e temporal nas tarefas de detecção de objetos, rastreamento e detecção de colisões, escolhendo apenas exemplos em alta definição.

Para ilustrar os resultados, a Tabela 5 apresenta os números de uma validação cruzada com k repetições usando o modelo Faster R-CNN. A tabela mostra um desempenho estável em quase todas as categorias; “pessoa” é a única exceção.

Tabela 5 – Validação cruzada usando Faster R-CNN. Os autores usaram a métrica mAP@0.5.

Repetição	Todos	Pessoa	Carro	Ônibus	Duas-rodas	Três-rodas	Outros
1	82.32	36.89	81.04	97.24	76.00	98.71	94.58
2	84.39	52.26	89.39	97.27	77.56	98.88	91.00
3	84.33	47.22	85.40	98.57	82.32	98.30	94.58
Mean	83.68	45.46	85.28	97.69	78.63	98.63	93.39

2.6 Trabalhos em comparação

Percebe-se que muitos pesquisadores aplicaram diferentes técnicas de aprendizado de máquina supervisionado a acidentes de trânsito terrestre, e alguns se dedicaram especificamente aos que envolvem motocicletas. Diversos algoritmos foram testados em dados estruturados e em imagens; entretanto, nenhum utilizou uma rede neural binária para classificar acidentes.

A [Tabela 6](#) apresenta um panorama geral do estado da arte, com os principais autores e suas contribuições. Em comparação com este trabalho, [Wahab e Jiang \(2019\)](#) e [Sadeky et al. \(2010\)](#) utilizam uma fonte de dados diferente para identificar acidentes. [Wahab e Jiang \(2019\)](#), [Sadeky et al. \(2010\)](#) e [Pashaei, Ghatee e Sajedi \(2020\)](#) não utilizam algoritmos que se adequem à proposta de equilibrar eficácia nos resultados e eficiência no uso de recursos. [Salih e Viriri \(2020\)](#) também usam LBCNN, mas para detecção de melanoma.

Outra contribuição relevante utilizada como insumo para este trabalho foram os dados coletados por [Shah et al. \(2018\)](#) para a construção do conjunto de dados CADP, conforme será detalhado no [Capítulo 3](#).

Tabela 6 – Comparativo do estado da arte apresentado.

Autores	Objeto de pesquisa	Objetivo	Modelo	Limitações
Wahab e Jiang (2019)	Relatórios de acidentes	Classificar gravidade de ferimentos	Árvore de decisão, floresta aleatória, k-ésimo vizinho mais próximo e regressão logística multinomial	Não utiliza imagens; utiliza algoritmos mais simples e apresenta resultados limitados
Sadeky et al. (2010)	Imagens de acidentes de trânsito	Reconhecimento de acidentes de trânsito em tempo real	Regressão logística	Utiliza imagens, mas apenas extrai atributos; recorre a algoritmo simples e apresenta resultados limitados
Pashaei, Ghaatee e Sajedi (2020)	Imagens de acidentes de trânsito	Detectar acidentes, classificar gravidade e veículos	Combinação de vários algoritmos de aprendizado de máquina	Obtém ótimos resultados, porém com composição muito complexa que exige muitos recursos computacionais
Salih e Viriri (2020)	Imagens dermatoscópicas de lesões de pele (ISIC 2018)	Detectar melanoma	LBCNN	Não aborda acidentes de trânsito
Shah et al. (2018)	Vídeos com acidentes de trânsito	Construir base de dados para detecção de objetos em acidentes de trânsito	Faster R-CNN	Foca apenas na detecção de objetos; objetivo é propor a base de dados, não otimizar resultados

3 Materiais e Métodos

Este capítulo apresenta os algoritmos utilizados e a metodologia para treinamento e avaliação da rede neural. Como resumido na metodologia da [Figura 10](#), a [subseção 3.2.1](#) parte da CADP para selecionar os vídeos, que são tratados e limpos na [subseção 3.2.2](#). Na [subseção 3.2.3](#), obtêm-se imagens e rótulos adequados para os próximos passos. Nesse ponto, as imagens são utilizadas por algoritmos de aprendizado de máquina na [subseção 3.2.4](#), e sua eficácia é medida na [subseção 3.2.5](#).

3.1 O ambiente de desenvolvimento

Todo o código foi desenvolvido em Python 3.12.3, executado em um computador com 16 GB de memória RAM, processador AMD Ryzen 5 4600G e placa de vídeo NVIDIA GeForce RTX 3070. Foram utilizadas diversas bibliotecas Python, como Label Studio 1.0.8 para a seleção de dados, NumPy 1.26.4, Pillow 10.4.0 e MoviePy 2.1.1 para o pré-processamento e a transformação. Para a construção das redes neurais na etapa de treinamento, empregou-se o framework TensorFlow 2.17.0; em conjunto com o Scikit-Learn 1.5.2, ele também foi usado para realizar os testes e calcular as métricas de avaliação. A interface NVIDIA CUDA 12.0.140 foi essencial para executar os cálculos das redes neurais na GPU.

3.2 Metodologia

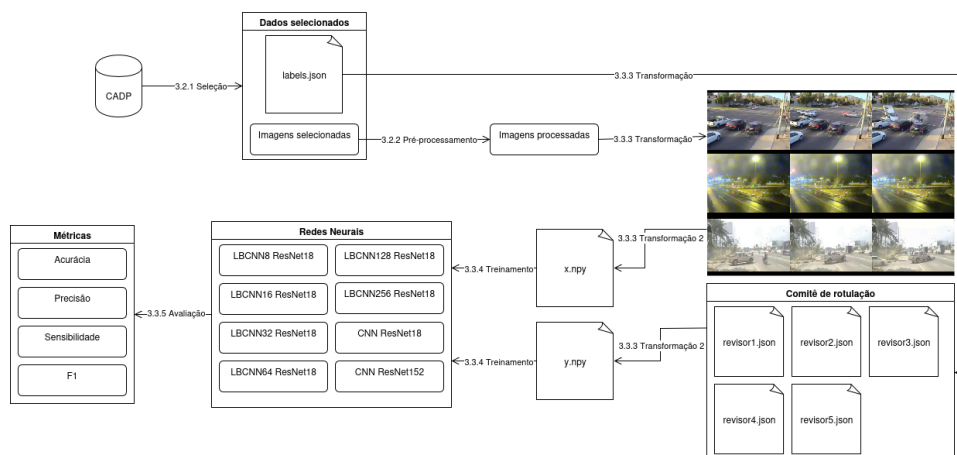


Figura 10 – Metodologia aplicada.

3.2.1 Seleção

Para a construção da base de dados de acidentes de trânsito terrestre, utilizaram-se como ponto inicial os vídeos coletados por [Shah et al. \(2018\)](#), que formam o CADP. Embora os autores tenham adotado uma metodologia própria para a seleção e o processamento desses dados, neste trabalho aproveitou-se apenas a etapa de exploração do motor de busca. A seleção, o processamento e a transformação foram realizados por meio de um método próprio.

Os dados coletados são, de fato, bastante úteis para este trabalho; entretanto, o escopo aqui é um pouco diferente. Consideram-se apenas vídeos de circuitos fechados de televisão. Algumas amostras encontram-se fora desse escopo: gravações de tela, filmagens com câmeras embarcadas em veículos e registros feitos por pedestres com celular ou outros aparelhos. Também foram detectados muitos ruídos e elementos visuais introduzidos na edição (que igualmente classificamos como ruído), como vinhetas, transições, baixo FPS, marcas-d'água, censuras, círculos, setas, marcações etc.

A [Figura 11](#) apresenta exemplos de vídeos da base de dados: os itens (a), (b), (d), (e), (f) e (g) foram descartados, enquanto os demais foram selecionados e processados nas etapas subsequentes.

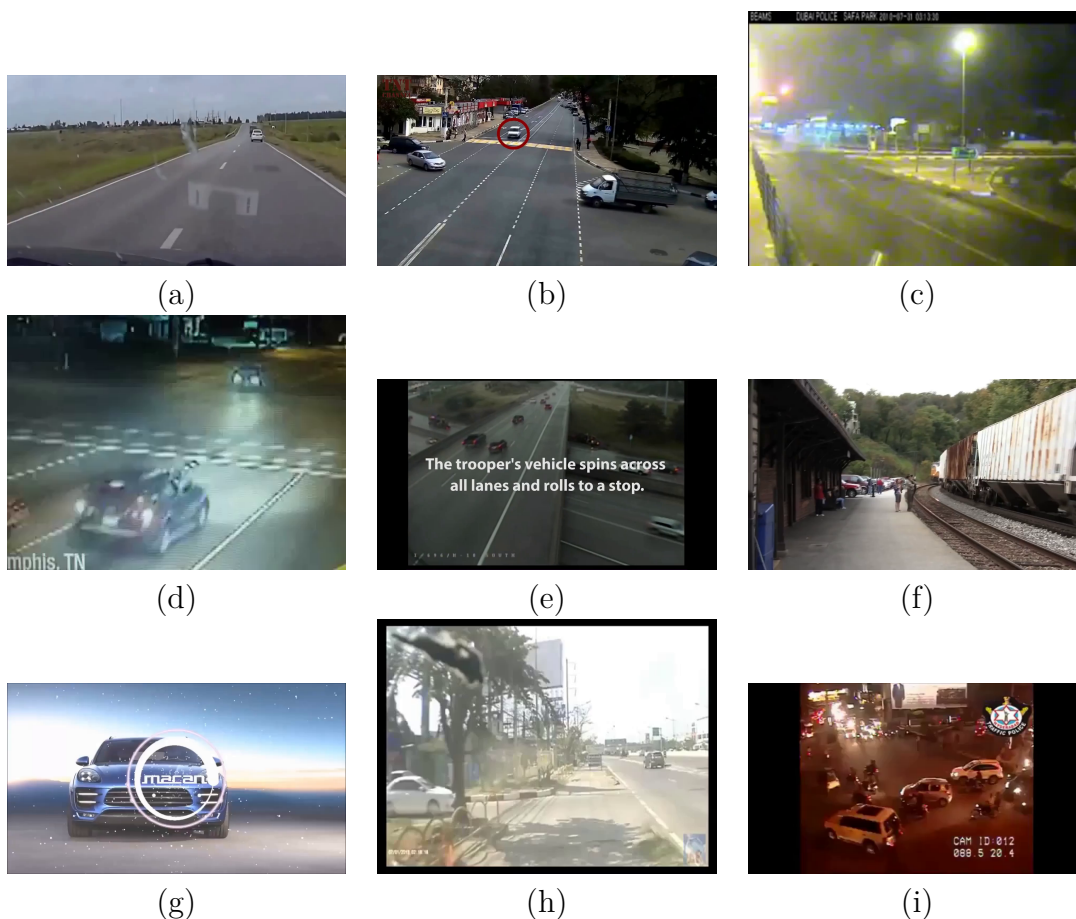


Figura 11 – Exemplos CADP: (a) a (i).

Os vídeos foram rotulados temporalmente em três classes possíveis, utilizando a ferramenta Label Studio:

- sem acidente;
- acidente sem motocicleta;
- acidente com motocicleta.

Tudo o que estava fora do escopo foi marcado para descarte, bem como os exemplos cujo ruído não pôde ser tratado no pré-processamento. Mais adiante, na etapa de transformação, um comitê valida os rótulos; quando não há consenso na anotação, o vídeo também é descartado, a fim de evitar imprecisões na resposta do modelo. Ao fim, os dados selecionados são cortados e separados por classe.

3.2.2 Pré-processamento

Em boa parte dos vídeos, foi possível cortar a imagem utilizando Label Studio e Pillow, mantendo apenas a região de interesse onde ocorre o acidente. Como consequência, obtivemos resoluções de imagem bastante variadas; portanto, o passo seguinte foi padronizá-las. Para isso, todos os frames foram redimensionados proporcionalmente para dimensões menores ou iguais a 640×480 e, em seguida, acrescentou-se uma borda para atingir exatamente essa resolução.

Outro problema foi a diversidade de fps; bastou acelerar ou desacelerar os vídeos para manter um padrão de 30 fps.

Devido às várias revisões dessa metodologia, muitos rótulos precisaram ser atualizados, e foi necessário remapear periodicamente os vídeos ao arquivo de rótulos durante o pré-processamento. Isso, porém, não impactou a base de dados: tratou-se apenas de localizar os rótulos corretos para cada arquivo, algo resolvido facilmente nessa etapa.

O resultado de todas essas rotinas está ilustrado na [Figura 12](#): trata-se de vídeos com dimensões diferentes que, anteriormente, continham ruídos visuais. Após o processamento, os dados foram limpos e padronizados.



Figura 12 – Exemplos de imagens após o pré-processamento: (a), (b) e (c).

3.2.3 Transformação

Para assegurar a confiabilidade das classes, um comitê de cinco revisores humanos reavaliou todas as amostras sem ter acesso aos rótulos originais. Na maioria dos casos, a decisão do comitê coincidiu com o rótulo atribuído na seleção. Sempre que surgiram divergências, adotou-se o rótulo indicado pela maioria dos revisores.

Houve também situações de empate. Alguns conflitos foram resolvidos incluindo-se o rótulo inicial como voto de desempate; em outros, foi necessário refinar o escopo da classe ou aplicar um pré-processamento adicional às imagens. Quando todas as alternativas se mostraram insuficientes, o vídeo foi descartado, conforme estabelecido na etapa de seleção.



Figura 13 – Exemplos de imagens transformadas: (a) a (i).

Concluída a etapa de revisão dos rótulos, antes do treinamento propriamente dito, foram executadas as seguintes rotinas de formatação dos dados:

1. Extração de frames – foi capturado um frame por segundo de cada vídeo, como evidenciado na [Figura 13](#);
2. Conversão para NumPy – tanto as imagens quanto os rótulos foram transformados em arrays NumPy;

3. Gravação binária com memory-map – a estrutura resultante foi salva em um arquivo binário e associada a um memory-map, otimizando o uso de memória ao carregar os dados em lotes durante o treinamento.

3.2.4 Treinamento

A arquitetura ResNet foi escolhida para compor, em conjunto com camadas LBC, uma LBCNN com o máximo de variações de pesos âncora possíveis dentro do tempo hábil desta pesquisa. Por isso, optou-se pelas seguintes configurações: a versão básica com 8 kernels e versões com 16 a 256 kernels, utilizando pesos definidos a partir da distribuição de Bernoulli, com esparsidade de 50% e 30%.

Originalmente, a ResNet é composta por camadas convolucionais baseadas no algoritmo tradicional de convolução. Nesta implementação, cada camada convolucional foi substituída por camadas LBC, construídas sobre a mesma interface das camadas tradicionais.

Para fins de comparação, também foram criados dois modelos baseados em convolução tradicional: ResNet-18 e ResNet-152. A menor versão proposta por [He et al. \(2016\)](#), a ResNet-18 (detalhada na [subseção 1.1.4](#)), foi utilizada por minimizar a quantidade de parâmetros.

3.2.5 Avaliação

Foram calculadas precisão, sensibilidade e F_1 para cada modelo, usando 70% dos dados para treinamento e 30% para teste. Considerando os termos da matriz de confusão da [Equação 3.1](#), a [Equação 3.2](#) e [Equação 3.3](#) empregam os valores de verdadeiro positivo, falso negativo, falso positivo e verdadeiro negativo para calcular essas três métricas, conforme detalhado por [Flach e Kull \(2015\)](#). A [Equação 3.4](#) mostra, ainda, como combinar precisão e sensibilidade em uma única métrica ([FLACH; KULL, 2015](#)).

$$\text{matriz de confusão} = \begin{bmatrix} vp & fn \\ fp & vn \end{bmatrix} \quad (3.1)$$

$$\text{precisão} = \frac{tp}{tp + fp} \quad (3.2)$$

$$\text{sensibilidade} = \frac{tp}{tp + fn} \quad (3.3)$$

$$f_1 = \frac{2 * \text{precisão} * \text{sensibilidade}}{\text{precisão} + \text{sensibilidade}} \quad (3.4)$$

Parte II

Parte Final

4 Resultados e Discussão

4.1 Resultados

A partir do CADP, montou-se uma base de dados com 15752 imagens e três rótulos: sem acidente, acidente sem moto e acidente com moto. Utilizou-se uma amostra de 30% para testar os modelos. Havia 4680 exemplos sem acidente, 8918 com acidente sem moto e 2154 com acidente com moto. A proporção no subconjunto de teste, conforme descrito nas Tabelas 7, 8 e 9, foi: 940 imagens sem acidente, 441 de acidente com moto e 1769 acidentes sem moto.

A Figura 14 apresenta os resultados das diversas redes neurais implementadas. Foram utilizadas duas redes neurais com convolução tradicional, com as arquiteturas ResNet18 e ResNet152, para fins de comparação, além de diversas variações da LBCNN baseadas na ResNet18. Entre os modelos binários, variou-se a quantidade de pesos-âncora, com redes utilizando 8, 16, 32, 64, 152 e 256 filtros, e com esparsidade de 50% ou 30%. Ao todo, foram 11 modelos: LBCNN8-ResNet18; LBCNN16-Bernoulli30-ResNet18 até LBCNN256-Bernoulli30-ResNet18; e LBCNN16-Bernoulli50-ResNet18 até LBCNN256-Bernoulli30-ResNet18.

Na Figura 14, observa-se primeiramente a CNN-ResNet18 e a CNN-ResNet152, utilizadas como base de comparação. Todos os modelos binários possuem menos parâmetros que as CNNs, porém, quanto mais filtros LBC forem utilizados, mais próxima fica a

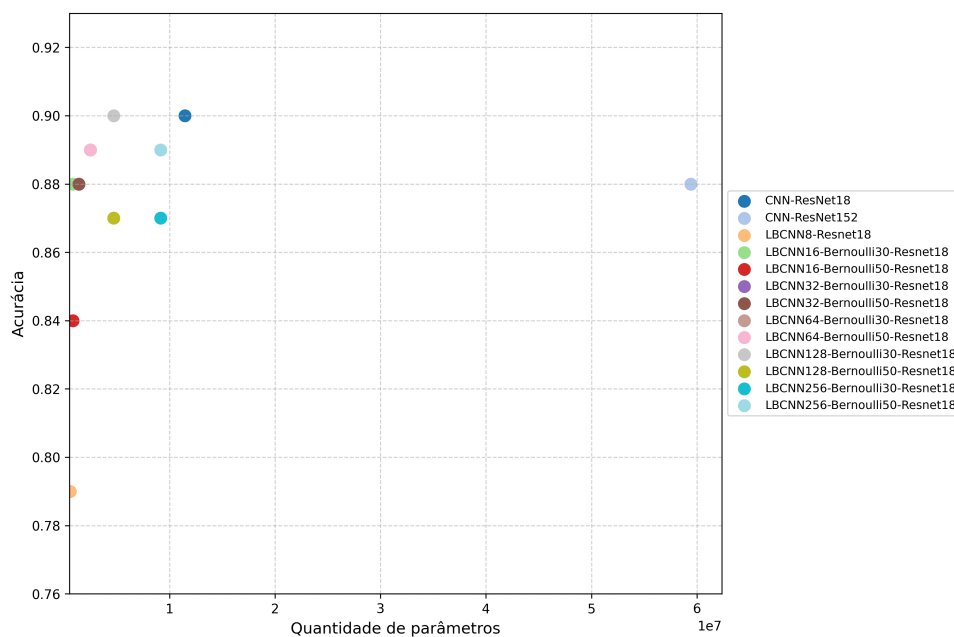


Figura 14 – Acurácia vs Quantidade de parâmetros.

Tabela 7 – Metricas por rótulo da CNN-ResNet18.

Rótulo	precisão	sensibilidade	f1-score	Quantidade de amostras
sem acidente	0.83	0.86	0.85	940
acidente com moto	0.93	0.88	0.90	441
acidente sem moto	0.93	0.93	0.93	1769

Tabela 8 – Metricas por rótulo da LBCNN8-ResNet18.

Rótulo	precisão	sensibilidade	f1-score	Quantidade de amostras
sem acidente	0.66	0.74	0.70	940
acidente com moto	0.64	0.73	0.90	441
acidente sem moto	0.86	0.86	0.93	1769

Tabela 9 – Metricas por rótulo da LBCNN16-Bernoulli30-Resnet18.

Rótulo	precisão	sensibilidade	f1-score	Quantidade de amostras
sem acidente	0.87	0.73	0.79	940
acidente com moto	0.94	0.87	0.90	441
acidente sem moto	0.94	0.92	0.93	1769

quantidade de parâmetros da CNN-ResNet18. Quanto à acurácia, todos os modelos permanecem próximos a partir da LBCNN16-ResNet18; na maioria deles, a métrica é levemente inferior (variando de 1% a 3% a menos). Apesar disso, em todas as LBCNNs, a menor esparsidade resulta em maior acurácia — os modelos com 30% são sempre superiores aos com 50%. Além disso, o modelo LBCNN8-ResNet18 utiliza os 8 filtros padrão propostos por [Juefei-Xu, Boddeti e Savvides \(2017\)](#), que possuem 90% de esparsidade, e por isso apresenta a pior acurácia (78%).

As Tabelas 7, 8, 9 e 10 detalham melhor os resultados apresentados no gráfico para alguns dos modelos, além de resumirem o resultado geral. Da CNN para a LBCNN8, observa-se uma queda considerável nas métricas para todos os rótulos, sendo mais acentuada nos acidentes sem moto (a precisão caiu de 93% para 64%).

Entretanto, ao se aumentar para 16 kernels e aplicar 30% de esparsidade, como na LBCNN16, a avaliação se aproxima dos valores originais. A precisão se mostra levemente superior, mas não de forma significativa (no rótulo "sem acidente", está 4% superior; nos

Tabela 10 – Metricas por rótulo da LBCNN16-Bernoulli50-Resnet18.

Rótulo	precisão	sensibilidade	f1-score	Quantidade de amostras
sem acidente	0.92	0.55	0.69	940
acidente com moto	0.71	0.94	0.81	441
acidente sem moto	0.85	0.96	0.90	1769

Tabela 11 – Matriz de confusão da LBCNN16-Bernoulli50-ResNet18 (rótulo real na vertical e rótulo previsto na horizontal).

	sem acidente	acidente com moto	acidente sem moto
sem acidente	516	139	285
acidente com moto	12	416	13
acidente sem moto	31	34	1704

demais, 1%).

Quanto à sensibilidade, a diferença diminui, mas ainda é relevante quando não há acidente (13% a menos); nos demais rótulos, a diferença é de apenas 1%.

Ao utilizar 50% de esparsidade, observa-se uma queda no desempenho do modelo, especialmente na sensibilidade para os casos "sem acidente". Um valor de 55% de sensibilidade indica que, dentre todos os casos que realmente não apresentam acidentes, apenas 55% foram classificados corretamente; os demais 45% foram considerados como contendo algum tipo de acidente (com ou sem moto). A [Tabela 11](#) detalha que, desses casos incorretamente classificados, 15% foram rotulados como "acidente com moto" (139 imagens) e 30% como "acidente sem moto" (285 imagens). Esse comportamento também contribuiu para a redução da precisão nos rótulos "acidente com moto" e "acidente sem moto".

4.2 Discussão

Conclui-se que o uso de camadas LBC em redes neurais baseadas na ResNet permite reduzir a quantidade de parâmetros utilizados e, conseqüentemente, a memória consumida e o tempo de processamento. Em relação ao desempenho do modelo, os resultados das métricas calculadas dependem da quantidade de filtros utilizados como pesos-âncora e do grau de esparsidade desses filtros. Quanto mais filtros são utilizados, mais o desempenho se aproxima de uma CNN tradicional, embora isso também implique um aumento no número de parâmetros. Um maior grau de esparsidade pode resultar em pior desempenho. No entanto, conforme proposto por [Lin et al. \(2021\)](#), é possível implementar otimizações no uso de memória e processamento em hardware a partir da esparsidade dos filtros (quanto mais espaços, mais eficiente o uso dos recursos).

Portanto, utilizar os modelos LBCNN16-Bernoulli30-ResNet18 ou LBCNN16-Bernoulli50-ResNet18 parece ser a escolha mais razoável: caso seja necessária maior eficiência, utiliza-se 50% de esparsidade; se o objetivo for obter melhores resultados, utiliza-se 30% de esparsidade.

Entretanto, é necessário ter cautela, pois a versão com 50% classifica erroneamente, com muita frequência, casos "sem acidente" como se houvesse algum tipo de acidente —

Tabela 12 – Matriz de confusão da CNN-ResNet18 com subconjuntos separados por vídeo (rótulo real na vertical e rótulo previsto na horizontal).

	sem acidente	acidente com moto	acidente sem moto
sem acidente	234	507	211
acidente com moto	244	605	108
acidente sem moto	101	711	191

cerca de 55% desses casos, como mencionado anteriormente.

Em aplicações reais, o sistema responsável pela tomada de decisão em situações de trânsito pode gerar um aumento nos custos relacionados ao tratamento dos alertas gerados pela classificação, justamente devido à quantidade de falsos positivos.

Os testes especificados, de modo geral, mostraram ótimos resultados. Entretanto, é importante ressaltar que os subconjuntos de teste e treino foram separados apenas a nível de imagens, ou seja, frames distintos de um mesmo vídeo podem constar tanto no subconjunto de teste quanto no subconjunto de treino. Ao resolver essa questão, separando vídeos para treino e vídeos para teste, os resultados mostram que os modelos desenvolvidos passam a classificar de forma aleatória todos os casos, como mostrado na matriz de confusão da [Tabela 11](#). A rede neural não conseguiu convergir para a classificação: os rótulos previstos estão espalhados entre os rótulos reais na matriz de confusão, sem qualquer padrão observável.

Conclusões e Trabalhos Futuros

4.3 Conclusão

Ao iniciar este trabalho, pretendia-se encontrar uma forma de otimizar o uso de memória e recursos de processamento na classificação de acidentes de trânsito terrestre. Na literatura, foram encontradas propostas tanto de software quanto de otimizações em hardware para a chamada rede neural local binária convolucional (LBCNN). Dessa forma, optou-se por utilizá-la neste estudo.

O objetivo foi avaliar o desempenho da LBCNN em comparação com uma rede neural convolucional tradicional, utilizando métricas como precisão, sensibilidade e F1, buscando-se a melhor arquitetura e os melhores parâmetros para essa rede neural.

Durante a implementação, encontrou-se o trabalho de [Shah et al. \(2018\)](#), que descreve a construção da base de dados CADP. A partir desse trabalho, os dados foram adaptados à tarefa abordada neste estudo: os dados coletados pelos autores foram pré-processados e transformados de acordo com os objetivos específicos deste trabalho.

Diversas redes neurais binárias foram implementadas com a arquitetura ResNet18 e comparadas às arquiteturas tradicionais ResNet18 e ResNet152. Dentre as várias opções, destacaram-se dois modelos: LBCNN16-Bernoulli30-ResNet18 e LBCNN16-Bernoulli50-ResNet18. O primeiro demonstrou maior eficácia na classificação (88% de acurácia), enquanto o segundo mostrou-se potencialmente mais eficiente, considerando a possibilidade de otimizações em hardware baseadas na esparsidade dos pesos-âncora.

O desempenho das LBCNNs foi levemente inferior ao das CNNs tradicionais. No entanto, essa é uma compensação esperada em função da otimização no uso de recursos computacionais. Conclui-se que os objetivos foram alcançados: a LBCNN utiliza, de fato, menos parâmetros e, conseqüentemente, requer menos recursos para processar as operações de uma rede neural. Além disso, embora o desempenho seja ligeiramente inferior, permanece dentro de um nível aceitável.

Entretanto, é importante destacar os péssimos resultados obtidos ao separar os subconjuntos de treino e teste de forma que frames de um mesmo vídeo constem em apenas um dos conjuntos. Os resultados dessa conclusão foram calculados em casos em que frames distintos de um mesmo vídeo podem estar tanto no treino quanto no teste.

Trabalhos Futuros

O próximo passo que daria continuidade a este trabalho seria, a partir da implementação da LBCNN, desenvolver um acelerador de domínio específico para camadas LBC, com o objetivo de otimizar ainda mais o uso de memória e os recursos de processamento na tarefa de classificação de acidentes de trânsito.

Outro possível abordagem para trabalhos futuros é buscar outras formas otimizações de redes neurais e outras implementações de redes neurais binárias, além de comparar os resultados com resultados exposto aqui.

Por fim, também se faz necessário encontrar meios para que, sendo os subconjuntos de treino e teste conjuntos disjuntos de vídeos, obtenham-se resultados melhores do que uma simples classificação aleatória.

Referências

- AL-MASNI, M. A. et al. Skin lesion segmentation in dermoscopy images via deep full resolution convolutional networks. *Computer Methods and Programs in Biomedicine*, v. 162, p. 221–231, 2018. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0169260718304267>>. Citado na página 20.
- BI, L. et al. Dermoscopic image segmentation via multistage fully convolutional networks. *IEEE Transactions on Biomedical Engineering*, v. 64, n. 9, p. 2065–2074, 2017. Citado na página 20.
- CARVALHO, C. H. R. Mortalidade por acidentes de transporte terrestre e desigualdades interestaduais no brasil. *Boletim Radar: Tecnologia, produção e comércio exterior*, v. 1, n. 67, p. 7–10, 2021. Citado na página 1.
- CODELLA, N. et al. *Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC)*. 2019. Disponível em: <<https://arxiv.org/abs/1902.03368>>. Acesso em: 7 nov. 2024. Citado na página 19.
- FLACH, P.; KULL, M. Precision-recall-gain curves: Pr analysis done right. In: CORTES, C. et al. (Ed.). *Advances in Neural Information Processing Systems 28 (NIPS 2015)*. Curran Associates Inc, 2015. p. 838–846. Disponível em: <https://papers.nips.cc/paper_files/paper/2015/file/33e8075e9970de0cfea955afd4644bb2-Paper.pdf>. Acesso em: 19 jun. 2024. Citado na página 27.
- GOYAL, M. et al. Skin lesion segmentation in dermoscopic images with ensemble deep learning methods. *IEEE Access*, v. 8, p. 4171–4181, 2020. Citado na página 20.
- GÉRON, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. Sebastopol, CA, USA: O’Reilly Media, Inc., 2019. Citado 6 vezes nas páginas 13, 7, 8, 9, 10 e 11.
- HE, K. et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. [S.l.: s.n.], 2015. p. 1026–1034. Citado na página 15.
- HE, K. et al. Deep residual learning for image recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016. p. 770–778. Disponível em: <<https://ieeexplore.ieee.org/document/7780459>>. Acesso em: 19 jun. 2024. Citado 4 vezes nas páginas 15, 7, 13 e 27.
- IOFFE, S.; SZEGEDY, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*. [S.l.]: JMLR.org, 2015. (ICML’15), p. 448–456. Citado na página 15.
- JUEFEI-XU, F.; BODDETI, V. N.; SAVVIDES, M. Local binary convolutional neural networks. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017. p. 4284–4293. Disponível em: <<https://ieeexplore.ieee.org/>

[document/8099939](#)>. Acesso em: 19 jun. 2024. Citado 5 vezes nas páginas 13, 2, 7, 12 e 32.

KUO, P.-J. et al. Derivation and validation of different machine-learning models in mortality prediction of trauma in motorcycle riders: a cross-sectional retrospective study in southern taiwan. *BMJ Open*, British Medical Journal Publishing Group, v. 8, n. 1, 2018. Disponível em: <<https://bmjopen.bmj.com/content/8/1/e018252>>. Citado na página 17.

LIN, I.-C. et al. A novel, efficient implementation of a local binary convolutional neural network. *IEEE Transactions on Circuits and Systems II: Express Briefs*, v. 68, n. 4, p. 7–10, 2021. Citado 2 vezes nas páginas 2 e 33.

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2015. p. 3431–3440. Disponível em: <<https://ieeexplore.ieee.org/document/7298965>>. Acesso em: 7 nov. 2024. Citado na página 20.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biology*, v. 5, n. 4, p. 115–113, 1943. Citado na página 7.

MINSKY, M.; PAPERT, S. *Perceptrons*. Cambridge, MA, USA: MIT Press, 1969. Citado na página 8.

PASHAEI, A.; GHATEE, M.; SAJEDI, H. Convolution neural network joint with mixture of extreme learning machines for feature extraction and classification of accident images. *Journal of Real-Time Image Processing*, v. 17, n. 4, p. 1051–1066, 2020. Citado 4 vezes nas páginas 1, 19, 21 e 22.

RAVINDRAN, V.; VISWANATHAN, L.; RANGASWAMY, S. A novel approach to automatic road-accident detection using machine vision techniques. *International Journal of Advanced Computer Science and Applications*, Science and Information (SAI) Organization Limited, v. 7, n. 11, p. 235–242, 2016. Citado na página 19.

REZAPOUR, M.; Mehrara Molan, A.; KSAIBATI, K. Analyzing injury severity of motorcycle at-fault crashes using machine learning techniques, decision tree and logistic regression models. *International Journal of Transportation Science and Technology*, v. 9, n. 2, p. 89–99, 2020. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2046043019301133>>. Citado na página 17.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: NAVAB, N. et al. (Ed.). *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Cham: Springer International Publishing, 2015. p. 234–241. Disponível em: <https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28>. Acesso em: 7 nov. 2024. Citado na página 20.

ROSENBLATT, F. *The perceptron: A perceiving and recognizing automaton (Project PARA)*. Buffalo, NY, USA, 1957. Citado na página 7.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning internal representations by error propagation. *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, v. 1, p. 318–362, 1986. Citado na página 8.

- RUSSELL, S.; NORVIG, P. *Artificial intelligence: a modern approach*. New York, NJ, USA: Pearson Education, 2020. Citado 4 vezes nas páginas 13, 9, 10 e 11.
- SADEKY, S. et al. Real-time automatic traffic accident recognition using hfg. In: *20th International Conference on Pattern Recognition*. IEEE, 2010. p. 3348–3351. Disponível em: <<https://ieeexplore.ieee.org/document/5597511>>. Acesso em: 19 jun. 2024. Citado 6 vezes nas páginas 13, 1, 18, 19, 21 e 22.
- SALIH, O.; VIRIRI, S. Skin lesion segmentation using local binary convolution-deconvolution architecture. *Image Analysis and Stereology*, v. 39, n. 3, p. 169–185, Nov. 2020. Disponível em: <<https://www.ias-iss.org/ojs/IAS/article/view/2397>>. Citado 5 vezes nas páginas 15, 19, 20, 21 e 22.
- SANTOS, D. et al. Machine learning approaches to traffic accident analysis and hotspot prediction. *Computers*, v. 10, n. 12, 2021. Disponível em: <<https://www.mdpi.com/2073-431X/10/12/157>>. Citado na página 17.
- SANTOS, K. et al. Analysis of motorcycle accident injury severity and performance comparison of machine learning algorithms. *Transportation Research Record*, v. 2678, n. 1, p. 736–748, 2024. Disponível em: <<https://doi.org/10.1177/03611981231172507>>. Citado na página 17.
- SHAH, A. P. et al. Cadp: A novel dataset for cctv traffic camera based accident analysis. In: *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. [S.l.: s.n.], 2018. p. 1–9. Citado 5 vezes nas páginas 20, 21, 22, 24 e 35.
- SILVA, C. F. da. *Acelerador para a Convolução de Redes Neurais Binárias visando Computação de borda*. Dissertação (Dissertação de Mestrado) — Universidade Federal de Santa Catarina, 2022. Citado na página 2.
- SIMONYAN, K.; ZISSERMAN, A. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. Disponível em: <<https://arxiv.org/abs/1409.1556>>. Citado 3 vezes nas páginas 13, 14 e 15.
- SZEGEDY, C. et al. Going deeper with convolutions. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2015. p. 1–9. Citado na página 15.
- TAMAGUSKO, T. et al. Deep learning applied to road accident detection with transfer learning and synthetic images. *Transportation Research Procedia*, v. 64, p. 90–97, 2022. International Scientific Conference “The Science and Development of Transport - Znanost i razvitak prometa”. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2352146522006263>>. Citado na página 19.
- TANG, Y. et al. A multi-stage framework with context information fusion structure for skin lesion segmentation. In: *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. IEEE, 2019. p. 1407–1410. Disponível em: <<https://ieeexplore.ieee.org/document/8759535>>. Acesso em: 7 nov. 2024. Citado na página 20.

- TSCHANDL, P.; ROSENDAHL, C.; KITTLER, H. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, v. 5, n. 1, p. 2052–4463, 2018. Citado na página 19.
- United Nations. Improving global road safety. *UN General Assembly*, v. 74, n. A/RES/74/299, 2020. Disponível em: <<https://documents.un.org/doc/undoc/gen/n20/226/30/pdf/n2022630.pdf>>. Citado na página 1.
- WAHAB, L.; JIANG, H. A comparative study on machine learning based algorithms for prediction of motorcycle crash severity. *PLOS ONE*, Public Library of Science, v. 14, n. 4, p. 1–17, 04 2019. Disponível em: <<https://doi.org/10.1371/journal.pone.0214966>>. Citado 6 vezes nas páginas 15, 1, 17, 18, 21 e 22.
- World Health Organization. *Road traffic injuries*. 2023. Portal Organização Mundial da Saúde. Disponível em: <<https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries>>. Acesso em: 4 nov. 2024. Citado na página 1.
- YUAN, Y.; CHAO, M.; LO, Y.-C. Automatic skin lesion segmentation using deep fully convolutional networks with jaccard distance. *IEEE Transactions on Medical Imaging*, v. 36, n. 9, p. 1876–1886, 2017. Citado na página 20.
- ZHANG, S. et al. Hybrid feature selection-based machine learning classification system for the prediction of injury severity in single and multiple-vehicle accidents. *PLOS ONE*, Public Library of Science, v. 17, n. 2, p. 1–19, 02 2022. Disponível em: <<https://doi.org/10.1371/journal.pone.0262941>>. Citado na página 17.