Proposta de Modelos Leves para Classificação de Vagas de Estacionamentos em Cidades Inteligentes

Luan Marko Kujavski
luan.marko@ufpr.br
Departamento de Informática
Universidade Federal do Paraná
Curitiba, Paraná, Brasil

Paulo Mateus Luza Alves
paulomateus@ufpr.br
Departamento de Informática
Universidade Federal do Paraná
Curitiba, Paraná, Brasil

Paulo Lisboa de Almeida
paulorla@ufpr.br
Departamento de Informática
Universidade Federal do Paraná
Curitiba, Paraná, Brasil

ABSTRACT

In smart cities, a common problem is the parking spots classification into empty and occupied. It may seem simple, but a large number of Deep Learning approaches rely on CNNs (Convolutional Neural Networks). These solutions are commonly expensive, demanding high computational power and specialized hardware to run properly, making them unsuitable for large-scale deployments, such as in smart cities. In this work, we propose two lightweight CNN architectures, built upon existing solutions by enhancing their efficiency and robustness. We used a cross-dataset scenario, where a model is trained and validated in two datasets and tested in another, applying three robust state-of-the-art datasets: PKLot, CNRPark-EXT and PLds. This process improves generalization across different contexts and sets a more realistic scenario when compared to real urban environments. Also, we compared our models to state-of-the-art networks, such as MobileNetV3 Large and Small to ensure consistency and validate the results with well-explored models in the literature. Our results showed that our models, with up to 34× and 88× fewer parameters than the MobileNetV3 Large, reach less than 2% lower accuracy when compared to the MobileNetV3 networks. Furthermore, by using grayscale images, the results were slightly better and also decreased processing and storage costs.

PALAVRAS-CHAVE

Aprendizado Profundo, Visão Computacional, Classificação de vagas de estacionamento, Modelos leves

1 INTRODUÇÃO

A rápida evolução da tecnologia e do Aprendizado de Máquina tem propiciado avanços e melhorias em inúmeros campos de atuação, como a automação de sistemas de monitoramento em cidades inteligentes. Nesse contexto, a atividade de classificação de vagas de estacionamento se mostra relevante, podendo amenizar problemas urbanos como congestionamentos, emissões de gases, dentre outros [1]. Entretanto, a vasta maioria dos sistemas baseados em Aprendizado de Máquina faz uso de modelos de visão computacional, como Redes Neurais Convolucionais (Convolutional Neural Networks - CNN) complexas, sendo necessários recursos computacionais significativos para a realização das atividades, impossibilitando o seu uso em sistemas limitados, como câmeras inteligentes ou dispositivos de ponta (computação de borda).

Visando endereçar esta limitação, neste trabalho são propostas duas arquiteturas de CNNs leves, capazes de operar diretamente em dispositivos de ponta, possuindo performance levemente inferior a modelos do estado da arte, como a MobileNetV3 [2], porém com até 88 vezes menos parâmetros.

Para isso, nosso processo de desenvolvimento se baseou na rede proposta por Hochuli et al. [3], uma CNN composta de três camadas convolucionais, abrangendo um total de 158.914 parâmetros. Desta forma, foi aplicada uma sequência de modificações incrementais, sendo validadas empiricamente a fim de equilibrar o custo e o desempenho da arquitetura e, ao mesmo tempo, melhorar os resultados obtidos no artigo de origem.

Para o estudo, foram utilizados conjuntos de dados robustos, amplamente aplicados em trabalhos do estado da arte para a classificação de vagas de estacionamento, como PKLot [4], CNRPark-EXT [5] e PLds [6] (veja um exemplo na Figura 1) para o treinamento e teste dos modelos. Ademais, os resultados obtidos foram comparados com modelos do estado da arte, como a MobileNetV3 [2].

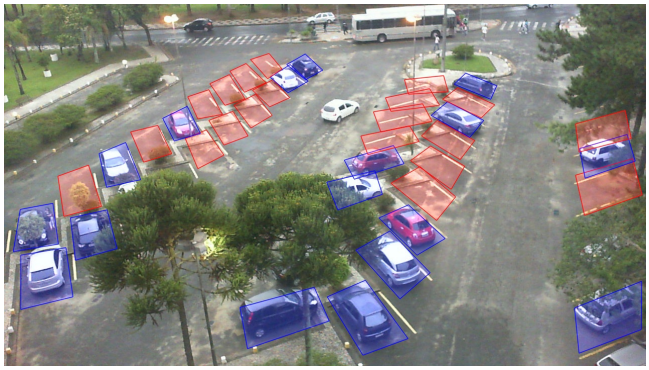

Figura 1: Imagem da PKLot contendo vagas classificadas entre ocupadas (azul) e vazias (vermelho).

Desta forma, este trabalho visou responder à seguinte pergunta de pesquisa:

- Q1: Como otimizar arquiteturas de redes neurais leves para classificação de vagas de estacionamento visando sua implementação em dispositivos de ponta (computação de borda)?

Os resultados obtidos demonstraram que as arquiteturas propostas neste trabalho, contendo 34× e 88× menos parâmetros quando comparadas a modelos complexos do estado da arte, como a MobileNetV3 Large [2], atingiram resultados médios de 93,21% e 93,04% de acurácia, respectivamente, estando apenas 1,68 e 1,85 pontos percentuais abaixo dela.

Desta maneira, este trabalho foi dividido da seguinte forma: na Seção 2 são discutidos os trabalhos relacionados e do estado da arte na questão de classificação de vagas de estacionamento. Na Seção 3 são apresentados os conjuntos de dados utilizados no processo. Na Seção 4, os experimentos e os resultados são detalhados para que,

na Seção 5, sejam discutidos. Por fim, na Seção 6, tiramos nossas conclusões sobre o estudo.

2 TRABALHOS RELACIONADOS

Esta Seção foca em apresentar trabalhos relacionados e de estado da arte que utilizam ou propõem modelos leves para a classificação de vagas de estacionamento. Estas redes normalmente são aplicadas como alternativas a modelos maiores e mais custosos, buscando diminuir o gasto computacional relacionado à inferência e possibilitando seu uso em dispositivos de ponta (computação de borda) ou que possuem restrições de *hardware*.

Por exemplo, em Amato et al. [7] é proposto o uso de uma mAlexNet [8], que é baseada na AlexNet [9], composta de três camadas convolucionais e duas camadas densas, sendo implementada diretamente em uma câmera inteligente para a realização da classificação das vagas de estacionamento, não sendo necessário o envio de imagens para um servidor. Ademais, o método foi testado utilizando unicamente a CNRPark-EXT [5] apresentando um *Overall Error Rate* de 0,4, sendo necessários 15 segundos para o processamento de uma imagem de entrada[1].

Mais recentemente, Hochuli et al. [3] propôs uma CNN de três camadas contendo 158.914 parâmetros, dimensionada especialmente para dispositivos com alta restrição computacional, como câmeras inteligentes. Entretanto, mesmo não exigindo imagens do conjunto de dados alvo para treinamento, este modelo atingiu apenas 80,9% de acurácia utilizando a PKLot [4] e a CNRPark-EXT [5]. Para comparação, a MobileNetV3, com 4.204.594 parâmetros, atingiu uma acurácia de 89,9%.

De forma semelhante, Zhang et al. [10] propõe um modelo leve para classificação de imagens de baixa resolução, buscando endereçar possíveis questões de segurança e privacidade pública relacionadas à utilização de imagens de alta resolução. Entretanto, durante os testes de construção da rede, uma coleta aleatória de imagens da PKLot [4] e CNRPark-EXT [5] foi efetuada para confecção dos conjuntos de treinamento e teste, ignorando o conceito temporal existente nestes conjuntos de dados e gerando possíveis vieses [11]. Por fim, o método proposto atingiu em média 91,68% de acurácia utilizando a CNRPark-EXT [5], separando o conjunto em câmeras ímpares e pares, alternando entre treinamento e teste.

No mesmo contexto, em Yuldashev et al. [12] é proposta uma modificação da MobileNetV3 para a classificação de vagas de estacionamento. Esta versão é composta por diversas alterações de estruturas internas da rede, como módulos e funções de ativação, buscando especialização para o problema em questão. Ademais, os autores reportaram uma acurácia de 99% em cenários onde as imagens de treinamento e teste advêm do mesmo estacionamento e 95% a 98% em cenários nos quais os dados vêm de conjuntos diferentes, utilizando a PKLot [4] e CNRPark-EXT [5]. Entretanto, a técnica de validação cruzada *k-fold*[2] foi aplicada durante o treinamento das redes, possivelmente gerando vieses nos resultados.

Já em Alves et al. [13] é proposto o uso de uma CNN de três camadas [3] como modelo estudante na aplicação da técnica de destilação de conhecimento [14]. O objetivo dos autores foi reduzir o custo computacional necessário para a classificação de vagas de estacionamento, destilando o conhecimento do modelo professor por meio de pseudo-rótulos. Por fim, os autores reportaram uma acurácia média acima de 96% quando o modelo estudante é treinado com os rótulos gerados pelos professores no mesmo estacionamento de treinamento e teste, sendo necessário em média 0,01 segundos para a classificação de uma vaga em ocupada/vazia em um dispositivo com baixo poder computacional[3].

A Tabela 1 apresenta um resumo dos resultados obtidos nos trabalhos discutidos nesta Seção. Desta forma, observa-se que modelos com melhor generalização e poucos parâmetros são necessários, visando um melhor benefício entre custo e desempenho.

Tabela 1: Resumo dos trabalhos de classificação de vagas de estacionamento.

Autores	Modelo	Parâm.[4]	Acc (%)[5]
Amato et al. [7]	mAlexNet	32.380	-
Hochuli et al. [3]	*3-Conv. Layer*	158.914	80.9
Zhang et al. [10]	Não nomeado	31.702	91,68
Yuldashev et al. [12]	MobineNetV3 mod.	> 4.000.000	95 - 98
Alves et al. [13]	*Custom* [3]	158.914	-
Média	-	-	89,69

3 CONJUNTOS DE DADOS UTILIZADOS

Foram utilizados três conjuntos de dados para a execução dos experimentos[6], sendo eles a PKLot [4], CNRPark-EXT [5] e PLds [6]. Para facilitar a compreensão do estudo, CNRPark-EXT é chamada de CNRPark. Todos os conjuntos utilizados possuem imagens retiradas de estacionamentos reais em diferentes épocas, ângulos de câmeras, estações do ano e climas diferentes. A Tabela 2 apresenta um resumo geral das características dos conjuntos de dados utilizados.

O conjunto de dados PKLot [4] contém imagens dos estacionamentos da Universidade Federal do Paraná (UFPR) e da Pontifícia Universidade Católica do Paraná (PUCPR). As imagens foram capturadas a cada cinco minutos por aproximadamente três meses sem interseção entre os dias de captura, apresentando três diferentes climas: Ensolarado, Nublado e Chuvoso. As imagens foram armazenadas em resoluções de 1280×720 pixels em formato JPEG. Desta forma, foram capturadas 12.400 imagens, compondo 1.199.857 vagas de estacionamento distribuídas entre ocupadas e vazias.

Ao todo, dois estacionamentos foram capturados, possuindo dois ângulos de câmera no estacionamento da UFPR, denominados

[1] Não fica claro se os autores cronometraram o tempo para ambas classificação e sobrecargas, como o corte das imagens.

[2] Consiste em separar os dados de treinamento em k grupos, utilizando, de forma alternada, $k-1$ grupos para treinamento e 1 para validação.

[3] Foi utilizado um *Raspberry Pi 5*, e tempo considera as sobrecargas de carregamento e corte da imagem do estacionamento.

[4] A quantidade de parâmetros foi aproximada seguindo as informações apresentadas nos artigos.

[5] Cenários de *cross-dataset*, ou seja, imagens de estacionamentos ou câmeras diferentes são dadas para treinamento e teste

[6] O numero total de imagens ultrapassa os valores originalmente publicados em [4–6], pois foram inseridas novas imagens rotuladas. Em adição, as anotações e os retângulos rotacionados foram padronizados para todos os estacionamentos.

UFPR04 e UFPR05, posicionados respectivamente no 4° e 5° andar do prédio de administração da universidade e um ângulo de câmera no estacionamento da PUCPR, sendo posicionado no 10° andar do prédio de administração, e denominado PUCPR. Um exemplo de imagem deste conjunto de dados pode ser encontrado na Figura 2.

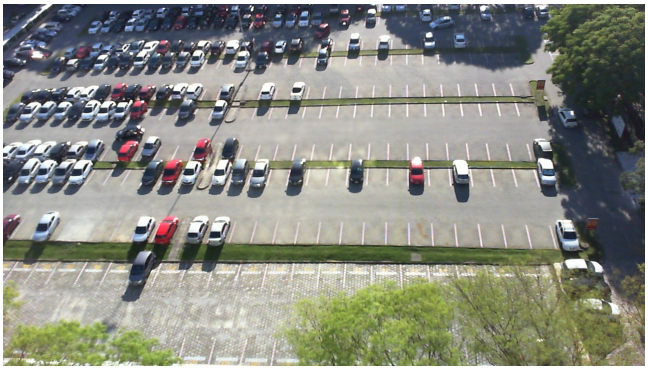

Figura 2: Imagem da câmera PUCPR do conjunto de dados PKLot [4].

O conjunto de dados CNRPark é uma extensão da CNRPark [5], contendo 4.278 imagens capturadas no campus do *National Research Concil* (CNR), em Pisa - Itália, propiciando 148.916 vagas de estacionamento. As imagens foram coletadas a cada 30 minutos de nove diferentes ângulos de câmeras simultaneamente e sobre o mesmo estacionamento, sendo armazenadas em resoluções de 1000×750 pixels. O conjunto engloba três climas: Nublado, Chuvoso e Ensolarado.

Desta forma, a CNRPark abrange uma grande quantidade de ângulos sobre o mesmo estacionamento, propiciando diferentes oclusões e desafios. Um exemplo pode ser encontrado na Figura 3.

Figura 3: Imagem da câmera CAMERA1 do conjunto de dados CNPark-EXT [5].

O último conjunto de dados utilizado foi a PLds [6]. Este conjunto contém 8.616 imagens capturadas no estacionamento do *Pittsburgh International Airport*, armazenadas com resolução de 1280 × 960, totalizando 105.811 vagas de estacionamento e quatro ângulos de câmera diferentes. As imagens foram capturadas em cinco climas: Nublado, Chuvoso, Ensolarado, Neve e Noite Clara.

Diferentemente dos demais conjuntos de dados utilizados, a PLds possui uma quantidade mais diversa de climas, sendo o único conjunto contendo imagens de estacionamento com neve. Ademais, de forma semelhante à CNRPark-EXT, também há imagens no período noturno. Um exemplo pode ser encontrado na Figura 4.

Figura 4: Imagem da câmera ISSHK do conjunto de dados PLds [6].

Tabela 2: Conjuntos de Dados utilizados

PKLot – 12.400 imagens					
# Dias	# Estac.	# Ângulos	# Ocupado	# Vazio	# Total
100	2	3	656.273	543.584	1.199.857
CNRPark – 4.278 imagens					
# Dias	# Estac.	# Ângulos	# Ocupado	# Vazio	# Total
23	1	9	66.934	81.982	148.916
PLds – 8.616 imagens					
# Dias	# Estac.	# Ângulos	# Ocupado	# Vazio	# Total
71	1	4	60.449	45.362	105.811

4 EXPERIMENTOS E RESULTADOS

Nesta Seção os experimentos e resultados obtidos para validar as arquiteturas testadas empiricamente neste trabalho serão introduzidos. Desta forma, na Seção 4.1 serão apresentadas as técnicas de treinamento, validação e teste utilizadas nos experimentos. Em sequência, a Seção 4.2 descreve a ordem de testes empíricos realizados que culminaram nas versões finais das arquiteturas apresentadas neste trabalho.

4.1 Treinamento das Redes

Utilizando os conjuntos de dados apresentados na Seção 3, a técnica de teste cruzado foi aplicada, na qual dois conjuntos são utilizados para treinamento e validação e o terceiro para teste, sendo eles alternados durante a execução dos experimentos. De tal forma que, durante os testes, o modelo recebe imagens de estacionamentos totalmente diferentes nos quais foi treinado. Desta forma, uma grande

quantidade de cenários diferentes pode ser definida, melhorando a diversificação dos resultados. A Tabela 3 apresenta os agrupamentos criados com esta técnica. Cada resultado obtido com alguma das divisões é apresentado nas tabelas referindo-se somente ao conjunto de dados de teste para facilitar a compreensão do estudo.

Tabela 3: Esquema de divisão de conjunto de dados.

Treinamento	Validação	Teste
PKLot + CNRPark	PKLot + CNRPark	PLds
PKLot + PLds	PKLot + PLds	CNRPark
CNRPark + PLds	CNRPark + PLds	PKLot

Em adição, para a construção dos conjuntos de treinamento e validação foi aplicado uma divisão 70% - 30% (em ordem cronológica dos dias) do conjunto de dados de treinamento, evitando que imagens dos mesmos dias e, consequentemente, dos mesmos carros nas mesmas posições estivessem presentes em ambos os conjuntos, removendo um possível enviesamento dos resultados obtidos. Ademais, os dados destes conjuntos foram nivelados a partir da classe menos recorrente.

A normalização dos dados de entrada foi aplicada em todos os experimentos. Para as arquiteturas definidas neste trabalho, a normalização foi feita por meio da média e desvio padrão dos dados de treinamento, já para as redes de estado da arte comparadas (mais detalhes na Seção 5), os valores referentes ao seu pré-treinamento foram utilizados.

Ademais, o otimizador Adam, com uma taxa de aprendizado de 0,001 e mini-lotes de 32 imagens foram aplicados durante o treinamento das redes, sendo que todas as camadas dos modelos testados durante o processo empírico de construção foram treinadas. Além disso, foram executadas 15 épocas de treinamento, e em todos os casos o classificador com menor erro nos dados de validação foi selecionado.

Por fim, foram aplicadas as técnicas de *Receiver Operating Characteristic* (ROC), e *Equal Error Rate* (EER), para definir o melhor liminar de classificação que minimiza a diferença entre falsos positivos e falsos negativos dos resultados de inferência dos modelos. Os resultados apresentados são uma média de três execuções.

4.2 Definição das Arquiteturas

Tomaremos como ponto de partida a arquitetura investigada em [3, 15], composta por três camadas convolucionais, duas camadas de *max-pooling* e duas camadas de classificação, levando como entrada uma imagem RGB 32×32, como demonstrado na Tabela 4. Esta arquitetura compreende 158.914 parâmetros. Uma visualização desta rede pode ser encontrada na Figura 5.

A primeira modificação aplicada na rede proposta em [3] foi a adição das camadas de Normalização em Lote (*Batch Normalization*) após cada camada convolucional e antes de cada função de ativação. Esta técnica é utilizada para normalizar os valores das ativações dos neurônios de Redes Neurais Profundas. A Equação 1 exemplifica as operações executadas nesta camada.

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{1}$$

Tabela 4: Especificação para a arquitetura proposta por [3, 15]. *s* significa *Stride* e *p* significa preenchimento.

Entrada	Operação	s	p	Saída
$32^2 \times 3$	Conv2d, 3×3	1	1	$32^2 \times 32$
$32^2 \times 32$	Max-Pool, 2×2	-	-	$16^2 \times 32$
$16^2 \times 32$	Conv2d, 3×3	1	0	$14^2 \times 64$
$14^2 \times 64$	Max-Pool, 2×2	-	-	$7^2 \times 64$
$7^2 \times 64$	Conv2d, 3×3	1	0	$5^2 \times 64$
1600	Densa	-	-	64
64	Densa	-	-	2

x_i - Ativação de entrada.
μ_B - Média do mini-lote.
σ_B^2 - Variância do mini-lote.
ϵ - Valor pequeno adicionado para evitar divisão por zero.

Com a introdução desta técnica, pode-se notar um aumento considerável nos resultados obtidos nos cenários de teste, como demonstrado na Tabela 5, evidenciando que o uso de Normalização em Lote pode melhorar o desempenho da arquitetura.

Tabela 5: Resultados sem Normalização em Lote vs. com Normalização em Lote. Acurácia ± Desvio Padrão.

	Teste - Acurácia (%)			
	CNRPark	PLds	PKLot	Média
Sem Norm.	92,3 ± 0,7	93,5 ± 0,6	88,1 ± 1,3	91,3 ± 2,3
Com Norm.	94,2 ± 0,5	94,3 ± 0,4	90,5 ± 0,6	93,0 ± 1,8

A partir desta rede, a seguinte pergunta foi levantada: a função de ativação *ReLU* é ideal para este cenário? Esta função de ativação, amplamente utilizada por diversos modelos, tem como um dos objetivos evitar o problema de desaparecimento de gradientes, presente em redes muito profundas. Porém, como nossa arquitetura compõe uma rede relativamente pequena, este problema não é evidente. Desta forma, duas diferentes funções de ativação foram testadas, sendo elas a *Hardswish* e *LeakyReLU*. Porém, nenhuma delas trouxe ganho considerável, sendo mais conveniente manter o uso da *ReLU*, dada sua alta eficiência, que otimiza o processo de inferência da rede.

Em sequência, a próxima pergunta foi levantada: a quantidade de filtros em cada camada é suficiente? Com esta questão, o estudo visou identificar se, caso a quantidade de filtros fosse alterada, os resultados obtidos seriam melhorados. Desta forma, foram testadas duas novas alternativas, uma dobrando a quantidade de filtros por camada e outra diminuindo pela metade. Os resultados obtidos podem ser encontrados na Tabela 6.

Como evidenciado, o uso do dobro de filtros não se provou eficaz, aumentando o tamanho da rede sem obter nenhum ganho de eficiência. Por outro lado, a segunda rede, possuindo metade da quantidade de filtros da rede de origem, mostrou-se interessante, tendo apenas um leve declínio de acurácia, porém com uma quantidade menor de parâmetros.

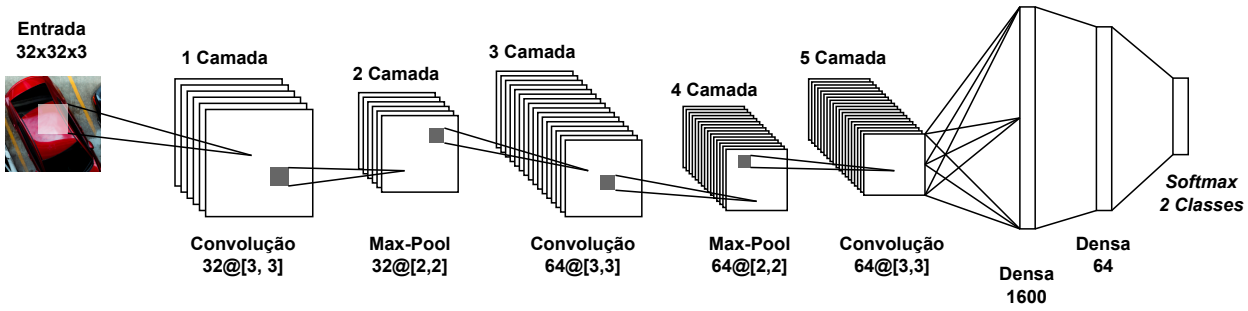

Figura 5: Arquitetura proposta por [3].

Tabela 6: Resultados do uso do dobro ou metade dos filtros da arquitetura de [3, 15] com Normalização em Lote. Acurácia ± Desvio Padrão.

		Teste - Acurácia (%)			
	Parâm.	CNRPark	PLds	PKLot	Média
Dobro	428.994	93,7 ± 0,3	94,9 ± 0,6	89,1 ± 3,0	92,6 ± 2,5
Normal	159.362	94,2 ± 0,5	94,3 ± 0,4	90,5 ± 0,6	93,0 ± 1,8
Metade	66.018	92,6 ± 0,2	93,2 ± 1,2	90,9 ± 1,0	92,3 ± 1,0

Dando continuidade, foi levantada uma nova questão: há benefícios em aumentar o tamanho da entrada de 32×32 para 64×64? Este estudo buscou investigar se o aumento do tamanho da entrada, proporcionando maior quantidade de informações, poderia resultar em melhorias no desempenho. Além disso, considerando que a redução pela metade no número de filtros resultou em redes mais leves, mas com desempenho similar ao da arquitetura normal, foi testado o impacto do aumento do tamanho da entrada em ambas as duas arquiteturas (normal e com metade de filtros). Porém, caso a composição das redes permanecesse inalterada, o número de parâmetros aumentaria significativamente devido ao crescimento das dimensões na camada de classificação, o que seria indesejável para os objetivos deste estudo. Para contornar esse problema, foi introduzida uma camada convolucional inicial em cada arquitetura, utilizando um *kernel* de tamanho cinco, *stride* de dois, *padding* de dois e saída de três filtros, seguida por uma camada de Normalização em Lote. Essa modificação permite reduzir as dimensões da entrada para $32 \times 32 \times 3$, mantendo a estrutura original das redes intacta a partir dessa camada. Os resultados obtidos estão apresentados na Tabela 7.

Tabela 7: Resultados das arquiteturas produzidas para entradas de imagens 64 x 64. Acurácia ± Desvio Padrão.

	Teste - Acurácia (%)			
	CNRPark	PLds	PKLot	Média
Normal	93,9 ± 0,6	94,2 ± 1,1	89,4 ± 0,9	92,5 ± 2,2
Metade	92,6 ± 0,8	93,6 ± 2,3	87,7 ± 2,4	91,3 ± 2,6

Como demonstrado, o ganho ao utilizar imagens de 64×64 não foi evidente. Em suma, todos os resultados obtidos foram inferiores às redes previamente testadas com entradas de 32×32, não justificando o aumento de complexidade existente ao dobrar o tamanho da imagem de entrada sob essas configurações. Porém, como não foram testadas outras abordagens, este estudo não pode afirmar que o uso de imagens 64×64 é pior. Para resultados mais concisos, uma bateria mais extensa de testes deverá ser executada, o que pode ser feito em trabalhos futuros.

Subsequentemente, foi observado que a arquitetura proposta por Hochuli et al. [3, 15] possui *padding* de tamanho um em sua primeira camada. Desta forma, o próximo teste buscou compreender se essa configuração é relevante para o resultado final. Caso não seja, sua remoção resultaria na diminuição da dimensão dos dados e, por consequência, no número de parâmetros. Os resultados na Tabela 8 mostram que, para a arquitetura normal, uma diminuição de aproximadamente 23% no número de parâmetros mostrou uma perda de acurácia de 1,02 pontos percentuais. Já para a arquitetura com metade dos filtros, houve redução de quase 28% no número de parâmetros, ao passo que apenas 0,73 pontos percentuais de acurácia foram perdidos. Desta forma, os resultados obtidos não se mostraram totalmente conclusivos sobre o uso ou não de *padding* na primeira camada convolucional. Por esse motivo, ambas as decisões de arquitetura foram avaliadas nos próximos testes.

Tabela 8: Resultados das arquiteturas sem *padding* na primeira camada. Acurácia ± Desvio Padrão.

		Teste - Acurácia (%)			
	Parâm.	CNRPark	PLds	PKLot	Média
Normal	122.498	92,3 ± 0.5	93,3 ± 0,1	90,2 ± 0.7	92,0 ± 1,3
Metade	47.586	92,3 ± 0,4	94,1 ± 0,2	89,2 ± 1,6	91,8 ± 2,0

Por fim, o último teste visou estudar se a diminuição de canais de entrada (de RGB para escala de cinza) seria vantajosa para as redes. Uma das principais vantagens em poder utilizar imagens em escala de cinza está no fato da diminuição do processamento necessário para realizar a inferência das imagens. Com ela, o número de canais é reduzido a apenas um e, portanto, o custo computacional reduz. Além disso, a capacidade de armazenamento necessária para guardar as imagens reduziria consideravelmente. Também, a utilização de imagens em escala de cinza abre espaço para o uso de dispositivos mais simples e baratos, além de possibilitar o uso de câmeras que coletam imagens em período noturno através de

visão infravermelha, que comumente coletam imagens em escala de cinza. Os resultados da Tabela 9 demonstraram que a utilização de imagens em escala de cinza pode ser benéfica para as arquiteturas. Isso pode ser explicado pois, removendo canais de cores da imagem, a rede pode ter passado a focar mais em características como bordas e texturas, de tal forma que a quantidade de canais presente nela foi suficiente para encontrar uma boa representação com essas informações. Além disso, os resultados também evidenciaram que o uso de redes sem *padding* é benéfico. Como apontado na tabela, todas as arquiteturas avaliadas apresentaram valores semelhantes de acurácia, justificando o uso das redes sem *padding*, por conta do seu menor número de parâmetros.

Tabela 9: Resultados das arquiteturas usando escala de cinza, *sp* significa sem *padding* e *cp* significa com *padding*. Acurácia ± Desvio Padrão.

	Teste - Acurácia (%)			
	CNRPark	PLds	PKLot	Média
Normal - *cp*	94,0 ± 0,4	95,0 ± 0,5	90,4 ± 0,6	93,1 ± 2,0
Normal - *sp*	93,5 ± 0,6	95,2 ± 0,6	90,9 ± 1,0	93,2 ± 1,8
Metade - *cp*	93,4 ± 0,2	94,7 ± 1,0	91,2 ± 0,1	93,1 ± 1,5
Metade - *sp*	93,2 ± 0,1	95,2 ± 0,6	90,7 ± 0,5	93,0 ± 1,8

Após a execução sequencial destes experimentos, obtivemos como resultados a definição de duas redes leves para a classificação de vagas de estacionamento denominadas *CustomNNLarge* e *CustomNNSmall*, possuindo 121.922 e 47.298 parâmetros, respectivamente. Ambas possuem três camadas convolucionais, duas camadas de *max-pooling* e duas camadas densas para classificação, tendo a Normalização em Lote aplicada após cada camada convolucional, sendo a principal diferença entre elas a quantidade de filtros aplicados por cada camada. Além disso, as entradas são dadas em escala de cinza, isto é, têm apenas um canal de cor. Uma representação visual das redes pode ser encontrada nas Figuras 6 e 7. Ademais, as Tabelas 10 e 11 detalham as configurações de ambas as redes.

Tabela 10: Especificação para a *CustomNNLarge*. Veja a Tabela 4 para notação.

Entrada	Operação	s	p	Saída
$32^2 \times 1$	Conv2d, 3×3	1	0	$30^2 \times 32$
$30^2 \times 32$	Batch Norm.	-	-	$30^2 \times 32$
$30^2 \times 32$	Max-Pool, 2×2	-	-	$15^2 \times 32$
$15^2 \times 32$	Conv2d, 3×3	1	0	$13^2 \times 64$
$13^2 \times 64$	Batch Norm.	-	-	$13^2 \times 64$
$13^2 \times 64$	Max-Pool, 2×2	-	-	$6^2 \times 64$
$6^2 \times 64$	Conv2d, 3×3	1	0	$4^2 \times 64$
$4^2 \times 64$	Batch Norm.	-	-	$4^2 \times 64$
1024	Densa	-	-	64
64	Densa	-	-	2

Tabela 11: Especificação para a *CustomNNSmall*. Veja a Tabela 4 para notação.

Entrada	Operação	s	p	Saída
$32^2 \times 1$	Conv2d, 3×3	1	0	$30^2 \times 16$
$30^2 \times 16$	Batch Norm.	-	-	$30^2 \times 16$
$30^2 \times 16$	Max-Pool, 2×2	-	-	$15^2 \times 16$
$15^2 \times 16$	Conv2d, 3×3	1	0	$13^2 \times 32$
$13^2 \times 32$	Batch Norm.	-	-	$13^2 \times 32$
$13^2 \times 32$	Max-Pool, 2×2	-	-	$6^2 \times 32$
$6^2 \times 32$	Conv2d, 3×3	1	0	$4^2 \times 32$
$4^2 \times 32$	Batch Norm.	-	-	$4^2 \times 32$
512	Densa	-	-	64
64	Densa	-	-	2

5 COMPARAÇÕES

Para validar nossos resultados, comparamos nossas arquiteturas com redes do estado da arte, em especial os modelos da família *MobileNetV3* [2], que contém duas versões: *Large* e *Small*. Esta família de modelos foi projetada com o intuito de ser utilizada em dispositivos móveis ou integrados, balanceando tamanho, velocidade de inferência e latência, enquanto mantém uma acurácia razoável para os *benchmarks* do estado da arte. Desta forma, sua arquitetura implementa uma série de técnicas avançadas para melhorar os blocos convolucionais da rede, como conexões residuais, transformações não lineares e blocos de *squeeze-and-excitation* que utilizam a relevância de cada canal da camada para computar o *feature map* de saída. Para mais detalhes dessas arquiteturas, veja Howard et al. [2]. Uma comparação dos tamanhos das redes pode ser encontrada na Tabela 12.

Ademais, as redes da família *MobileNetV3* foram pré-treinadas no conjunto de dados *ImageNet* e refinadas para o nosso caso de uso seguindo a mesma estrutura apresentada na Seção 4.1, sendo a única diferença que, para esses modelos, foi aplicado o processo de transferência de conhecimento, treinando apenas as camadas de classificação do modelo e a última camada convolucional.

Tabela 12: Comparação das arquiteturas utilizadas.

Arquitetura	# Parâmetros	Memória (IEEE 754 s.p.)
MobileNetV3 Large [2]	4.204.594	17,00 MB
MobileNetV3 Small [2]	1.519.906	6,20 MB
3-Conv.Layers [3, 15]	158.914	0,64 MB
CustomNNLarge	121.922	0,48 MB
CustomNNSmall	47.298	0,19 MB

A Tabela 13 apresenta os resultados obtidos na classificação de vagas de estacionamento utilizando todas as redes comparadas. Ambos os modelos da família *MobileNetV3* e a *3-Conv.Layers* tiveram entradas em formato RGB, sendo o tamanho de 128×128 para os primeiros e 32×32 para a segunda. Já para as arquiteturas deste estudo, foram dadas entradas em escala de cinza e de tamanho 32×32.

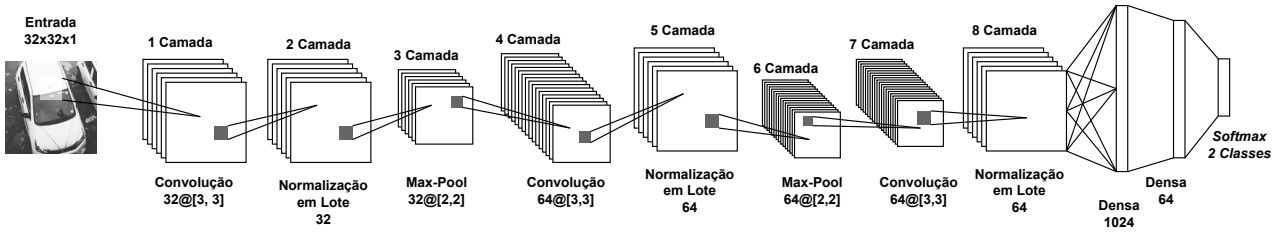

Figura 6: Arquitetura proposta da rede *CustomNNLarge*.

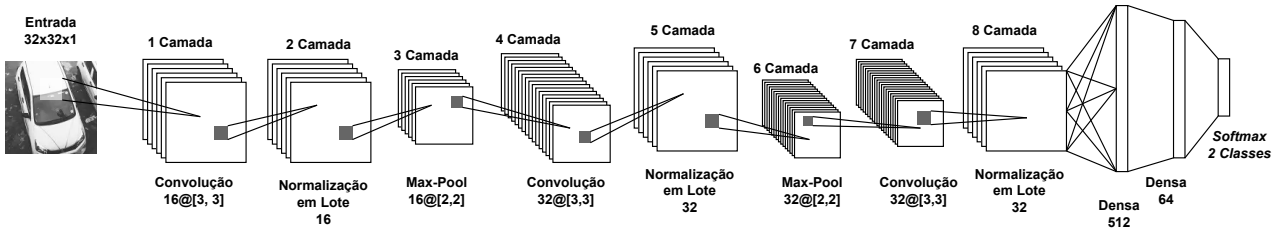

Figura 7: Arquitetura proposta da rede *CustomNNSmall*.

Tabela 13: Comparação na classificação de vagas de estacionamento. Acurácia ± Desvio Padrão.

	Teste - Acurácia (%)			
	CNRPark	PLds	PKLot	Média
MobileNetV3 L. [2]	95,4 ± 0,9	94,4 ± 1,2	94,7 ± 0,9	94,9 ± 0,3
MobileNetV3 S. [2]	95,6 ± 0,1	96,0 ± 0,3	94,3 ± 0,2	95,3 ± 0,1
3-Conv.Layers [3, 15]	92,3 ± 0,7	93,5 ± 0,6	89,3 ± 3,0	91,3 ± 0,4
CustomNNLarge	93,5 ± 0,6	95,2 ± 0,6	90,9 ± 1,0	93,2 ± 0,6
CustomNNSmall	93,2 ± 0,1	95,2 ± 0,6	90,7 ± 0,5	93,0 ± 0,1

Com os resultados, podemos validar que as nossas arquiteturas ficaram, em média, apenas 1,76% abaixo quando comparadas à *MobileNetV3 Large*, que é o maior modelo testado, porém com 34 vezes menos parâmetros no caso da *CustomNNLarge* e 88 vezes menos parâmetros no caso da *CustomNNSmall*, levando em consideração a mesma rede. Podemos notar também que os testes realizados no conjunto de dados PKLot, isto é, treinando os modelos na CNRPark e na PLds, obtiveram resultados inferiores aos demais devido à grande discrepância da quantidade de imagens para treinamento e validação.

Além disso, ambas as redes *CustomNNLarge* e *CustomNNSmall* atingiram desempenhos médios superiores em comparação com a rede proposta por Hochuli et al. [3]. Nossa primeira arquitetura atingiu resultado médio 1,92 pontos percentuais superior a esta. Já a segunda atingiu um resultado 1,75% maior, porém utilizando aproximadamente quatro vezes menos parâmetros, se mostrando uma boa alternativa que equilibra custo e eficiência.

6 CONCLUSÃO

Neste trabalho foram apresentadas duas arquiteturas de Redes Neurais Convolucionais especializadas na classificação de vagas de estacionamento. Além disso, o processo empírico de construção empregado foi detalhado, exemplificando as decisões tomadas, com base em resultados provisórios, para a construção dos modelos finais. Este trabalho chegou à seguinte conclusão em relação à pergunta de pesquisa:

Q1: Como otimizar arquiteturas de redes neurais leves para classificação de vagas de estacionamento visando sua implementação em dispositivos de ponta (computação de borda)? O processo de construção demonstrou que ao utilizar técnicas atuais, como a Normalização em Lote, em conjunto com leves modificações nas configurações das camadas, como número de filtros e valores de *padding*, além do uso de imagens em escala de cinza, melhoraram significativamente o desempenho das redes. As arquiteturas propostas apresentaram, em média, um desempenho 1,76% inferior ao de redes mais complexas e robustas, como a *MobileNetV3 Large*. No entanto, destacam-se pela significativa redução no número de parâmetros: 34 vezes menos na *CustomNNLarge* e 88 vezes menos na *CustomNNSmall*. Esses resultados evidenciam um equilíbrio notável entre custo e desempenho. Além disso, quando comparadas à arquitetura original proposta por Hochuli et al. [3], foi apresentada uma melhora significativa de desempenho, ao passo que diminuiu o número de parâmetros.

Por fim, nossos resultados demonstram que o uso de redes leves pode se equiparar a resultados obtidos por redes mais complexas, possibilitando o uso de Redes Neurais Convolucionais (CNNs) em dispositivos com alta restrição computacional, sem uma grande perda de desempenho.

AGRADECIMENTOS

Este trabalho foi financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) – Projeto 405511/2022-1.

REFERÊNCIAS

[1] Vijay Paidi, Hasan Fleyeh, Johan Håkansson, and Roger Nyberg. Smart parking sensors, technologies and applications for open parking lots: A review. *IET Intelligent Transport Systems*, 12, 04 2018. doi: 10.1049/iet-its.2017.0406.

[2] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3. 2019. URL https://arxiv.org/abs/1905.02244.

[3] Andre G. Hochuli, Alceu S. Britto, Paulo R. L. de Almeida, Williams B. S. Alves, and Fábio M. C. Cagni. Evaluation of different annotation strategies for deployment of parking spaces classification systems. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2022. doi: 10.1109/IJCNN55064.2022.9892783.

[4] Paulo R.L. de Almeida, Luiz S. Oliveira, Alceu S. Britto, Eunelson J. Silva, and Alessandro L. Koerich. Pklot – a robust dataset for parking lot classification. *Expert Systems with Applications*, 42(11):4937–4949, 2015. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2015.02.009. URL https://www.sciencedirect.com/science/article/pii/S0957417415001086.

[5] Giuseppe Amato, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro, Carlo Meghini, and Claudio Vairo. Deep learning for decentralized parking lot occupancy detection. *Expert Systems with Applications*, 72:327–334, 2017. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2016.10.055. URL https://www.sciencedirect.com/science/article/pii/S095741741630598X.

[6] Rafael Martín Nieto, Álvaro García-Martín, Alexander G. Hauptmann, and José M. Martínez. Automatic vacant parking places management system using multicamera vehicle detection. *IEEE Transactions on Intelligent Transportation Systems*, 20(3):1069–1080, 2019. doi: 10.1109/TITS.2018.2838128.

[7] Giuseppe Amato, Paolo Bolettieri, Davide Moroni, Fabio Carrara, Luca Ciampi, Gabriele Pieri, Claudio Gennaro, Giuseppe Riccardo Leone, and Claudio Vairo. A wireless smart camera network for parking monitoring. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6. IEEE, 2018.

[8] Giuseppe Amato, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro, and Claudio Vairo. Car parking occupancy detection using smart camera networks and deep learning. In *2016 IEEE Symposium on Computers and Communication (ISCC)*, pages 1212–1217, 2016. doi: 10.1109/ISCC.2016.7543901.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[10] Shuo Zhang, Xin Chen, and Zixuan Wang. Bcfpl: binary classification convnet based fast parking space recognition with low resolution image. In *International Conference on Image, Signal Processing, and Pattern Recognition (ISPP 2024)*, volume 13180, pages 1442–1449. SPIE, 2024.

[11] Paulo Ricardo Lisboa de Almeida, Jeovane Honório Alves, Rafael Stubs Parpinelli, and Jean Paul Barddal. A systematic review on computer vision-based parking lot management applied on public datasets. *Expert Systems with Applications*, 198:116731, 2022.

[12] Yusufbek Yuldashev, Mukhriddin Mukhiddinov, Akmalbek Bobomirzaevich Abdusalomov, Rashid Nasimov, and Jinsoo Cho. Parking lot occupancy detection with improved mobilenetv3. *Sensors*, 23(17):7642, 2023.

[13] Paulo Luza Alves, André Hochuli, Luiz Eduardo de Oliveira, and Paulo Lisboa de Almeida. Optimizing parking space classification: Distilling ensembles into lightweight classifiers. *arXiv preprint arXiv:2410.14705*, 2024.

[14] Lin Wang and Kuk-Jin Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *IEEE transactions on pattern analysis and machine intelligence*, 44(6):3048–3068, 2021.

[15] Andre Gustavo Hochuli, Jean Paul Barddal, Gillian Cezar Palhano, Leonardo Matheus Mendes, and Paulo Ricardo Lisboa de Almeida. Deep single models vs. ensembles: Insights for a fast deployment of parking monitoring systems. In *2023 International Conference on Machine Learning and Applications (ICMLA)*, pages 1379–1384, 2023. doi: 10.1109/ICMLA58977.2023.00208.