

Regularização em Redes Neurais com Reamostragem de Dados para Monitoramento Urbano Inteligente

Rodrigo Alexander Schinemann

rodrigo.schinemann@ufpr.br

Departamento de Informática
Universidade Federal do Paraná
Curitiba, Paraná, Brasil

Rayson Laroca

rayson@ppgia.pucpr.br

Programa de Pós-Graduação em Informática
Pontifícia Universidade Católica do Paraná
Curitiba, Paraná, Brasil

Paulo Lisboa de Almeida

paulorla@ufpr.br

Departamento de Informática
Universidade Federal do Paraná
Curitiba, Paraná, Brasil

Abstract

The advancement of smart cities has led to a growing demand for efficient urban management systems, such as parking space management. In this context, regularization has proven to be an essential technique for improving the generalization of deep learning models. This paper proposes a regularization method based on the use of smaller data subsets across multiple epochs with resampling, aiming to balance model learning and reduce the risk of overfitting. Using the MobileNetV3-Small, four experiments were conducted with the PKLot and CNRPark-EXT datasets. The results indicate that the proposed method achieved competitive results while utilizing only a fraction of the original data during training. By employing these strategies, it was possible to reduce the data required per epoch by up to 97%, while maintaining an average accuracy close to 89%. Furthermore, training with the PKLot dataset highlighted the positive impact on model robustness when using datasets with greater diversity and quantity. This study underscores the importance of new regularization approaches to enhance the efficiency and generalization of deep learning models in urban applications.

Palavras-chave

Regularização, Redes Neurais Convolucionais, Cidades Inteligentes, Gerenciamento de Estacionamentos, *Overfitting*.

1 Introdução

A transformação das cidades em ambientes inteligentes é um dos maiores desafios tecnológicos do século XXI, prometendo melhorias significativas na qualidade de vida e na eficiência dos serviços urbanos [27]. Entre os desafios enfrentados nesses ambientes, a gestão de espaços de estacionamento se destaca como uma questão crítica, especialmente em cidades com alta densidade populacional e demanda crescente por mobilidade [7]. Tecnologias baseadas em aprendizado de máquina, como redes neurais artificiais, têm o potencial de transformar esses serviços, promovendo soluções mais ágeis, precisas e escaláveis [21].

No entanto, a aplicação de redes neurais artificiais em cenários reais enfrenta desafios significativos. Um dos principais é o fenômeno do *overfitting*, em que o modelo aprende padrões específicos dos dados de treinamento, mas falha em generalizar para novos cenários [11]. Em ambientes urbanos, essa limitação é agravada pela variabilidade dos dados, como condições climáticas, iluminação e mudanças nos padrões de uso [26]. Esse desafio ressalta a necessidade de estratégias eficazes para melhorar a capacidade de generalização dos modelos, garantindo maior robustez e confiabilidade em aplicações reais.

A regularização é uma técnica essencial para melhorar a generalização dos modelos, especialmente em cenários com alta variabilidade nos dados [22]. Métodos tradicionais, como modificações na entrada dos dados ou ajustes na estrutura interna das redes neurais, já foram amplamente estudados. No entanto, uma abordagem ainda pouco explorada é a manipulação da quantidade de dados utilizados durante o treinamento, visando reduzir o *overfitting* e aprimorar a generalização. Diante desse cenário, este estudo propõe uma abordagem alternativa de regularização, que explora a utilização de subconjuntos reduzidos de dados combinados com reamostragem ao longo do treinamento. Essa estratégia busca melhorar a generalização dos modelos sem comprometer a eficiência computacional.

Estudos recentes [22] apontam que redes neurais convolucionais leves, como a MobileNetV3 [13], ainda são pouco exploradas em pesquisas sobre regularização. Projetada para maior eficiência computacional, essa arquitetura apresenta menor número de parâmetros e cálculos menos complexos, tornando-a uma opção promissora para cenários urbanos. Diante dessa lacuna na literatura, este estudo adota a MobileNetV3 para avaliar o impacto da estratégia de regularização proposta.

O restante deste artigo está organizado da seguinte forma. A Seção 2 apresenta os trabalhos relacionados, enquanto a Seção 3 aborda a fundamentação teórica. A metodologia é detalhada na Seção 4. Os resultados e as discussões são explorados na Seção 5, seguidos pelas limitações e possíveis direções para trabalhos futuros, discutidas na Seção 6. As conclusões são apresentadas na Seção 7.

2 Trabalhos Relacionados

Esta seção explora o uso de redes neurais e técnicas de regularização aplicadas a diferentes contextos, com foco em problemas urbanos, eficiência computacional e estratégias para melhorar a generalização dos modelos.

Uma revisão abrangente sobre a aplicação do aprendizado profundo em cidades inteligentes é apresentada por Muhammad et al. [21], que organizam um levantamento de técnicas utilizadas em domínios urbanos, incluindo monitoramento de tráfego e gestão de estacionamentos. No entanto, o estudo destaca desafios, como a necessidade de modelos mais robustos para lidar com a variabilidade dos ambientes urbanos dinâmicos.

Técnicas tradicionais de regularização, como *Dropout* [24], *L2 Regularization* [12] e *Batch Normalization* [14] são amplamente reconhecidos por sua eficácia na redução do *overfitting*, conforme destacado na revisão abrangente de Moradi et al. [20], que compara diferentes estratégias e seus impactos no desempenho de redes profundas. Além dessas técnicas estruturais, abordagens baseadas em manipulação dos dados, como *data augmentation* [23], também se

mostraram eficazes para aumentar a robustez dos modelos, especialmente em problemas de classificação. Técnicas como validação cruzada estratificada [17] também oferecem benefícios significativos, embora sua aplicação a conjuntos de dados urbanos, como PKLot [4] e CNRPark-EXT [2], ainda seja limitada, apontando para lacunas importantes na pesquisa.

Em paralelo, arquiteturas leves e computacionalmente eficientes, como a MobileNetV3-Small [13], têm se destacado como soluções promissoras em cenários com restrições de recursos, como cidades inteligentes. Apesar de suas vantagens, a literatura ainda carece de estudos aprofundados sobre o uso dessas redes no contexto de regularização. Santos et al. [22] destacam que a maioria das pesquisas se concentra em arquiteturas mais profundas, sugerindo que redes leves continuam sendo um campo aberto para investigação.

Outro trabalho relevante é o de Killamsetty et al. [16], que introduz o *framework* GLISTER para selecionar subconjuntos de dados otimizados, visando melhorar a generalização e eficácia de modelos de aprendizado de máquina. O GLISTER utiliza métodos de otimização submodular para selecionar subconjuntos que maximizem a perda de validação ou minimizem o gradiente da perda, promovendo aprendizado eficiente e robusto. Além disso, o *framework* foi estendido para cenários de aprendizado ativo, permitindo a seleção iterativa de amostras informativas. Esses aspectos fazem do GLISTER uma referência relevante para a abordagem explorada neste estudo, que também investiga a manipulação de subconjuntos de dados para regularização.

Por fim, Kadra et al. [15] apresentam uma abordagem chamada "coquetéis de regularização", na qual diferentes técnicas de regularização são combinadas e otimizadas para cada conjunto de dados. O estudo investiga 13 métodos de regularização e explora a busca por combinações ideais dessas técnicas para diferentes conjuntos de dados, incluindo *Dropout*, *L2 Regularization* e *Batch Normalization*, otimizadas para diferentes conjuntos de dados. Este estudo demonstra que redes simples, como perceptrons multicamadas, podem superar métodos mais tradicionais, como *Gradient-Boosted Decision Trees* [9, 10], em tarefas tabulares. Essas descobertas reforçam a relevância de explorar novas estratégias de regularização, como a manipulação de subconjuntos de dados, que é proposta neste artigo.

Embora métodos tradicionais de regularização e estratégias de seleção de subconjuntos ofereçam avanços significativos, alternativas como a manipulação da quantidade de dados usados por época permanecem pouco exploradas, apesar de seu potencial. Ao consolidar esses avanços, é evidente que técnicas como manipulação de subconjuntos, arquiteturas leves e *data augmentation* oferecem caminhos promissores para melhorar a robustez e a generalização de modelos em cenários urbanos. Além disso, a aplicação dessas abordagens em contextos dinâmicos, como cidades inteligentes, continua sendo um campo de grande potencial para futuras investigações.

3 Fundamentação Teórica

O aprendizado de máquina tem se consolidado como uma ferramenta poderosa para a resolução de problemas complexos em diversas áreas, desde visão computacional até análise de dados urbanos. O sucesso de um modelo depende de sua capacidade de generalização, ou seja, de manter um bom desempenho em dados não vistos

anteriormente [8, 19]. Essa característica é essencial em contextos como cidades inteligentes, onde as condições variam constantemente [4, 5, 25].

Nos últimos anos, o aprendizado profundo emergiu como um padrão de referência na comunidade científica, demonstrando desempenho equivalente ou superior ao humano em tarefas complexas, como classificação de imagens e detecção de padrões [1]. Uma revisão abrangente conduzida por Muhammad et al. [21], destaca o potencial do aprendizado profundo para aplicações urbanas, mas também aponta desafios, especialmente no que diz respeito à generalização dos modelos em cenários dinâmicos e variáveis.

Apesar de seu alto desempenho, modelos profundos frequentemente sofrem com *overfitting*, especialmente em conjuntos de dados com volume limitado ou alto nível de ruído [20]. Para mitigar esse problema, técnicas de regularização desempenham um papel essencial, ajudando o modelo a generalizar melhor sem comprometer sua capacidade de aprendizado. De acordo com Santos et al. [22], as técnicas de regularização podem ser classificadas em:

- Modificações nos dados de entrada, como *data augmentation* [23], que aumenta a diversidade do conjunto de dados aplicando transformações geométricas ou fotométricas.
- Modificações internas na arquitetura da rede, como ajustes nos pesos, *kernels*, e normalizações para reduzir a complexidade do modelo.
- Modificações na saída, como ajustes nos rótulos de treinamento para suavizar ou equilibrar a distribuição das classes.

Entre essas abordagens, *data augmentation* se destaca como uma solução eficaz para aumentar a diversidade do conjunto de treinamento sem a necessidade de coletar novos dados [23]. Em aplicações como detecção de vagas de estacionamento, essa técnica pode simular variações reais nas condições climáticas e de iluminação, tornando o modelo mais robusto.

Além das abordagens tradicionais, estratégias baseadas na manipulação de subconjuntos de dados vêm sendo exploradas para melhorar a eficiência computacional e a robustez do aprendizado. Métodos como GLISTER [16] utilizam uma seleção otimizada de subconjuntos para melhorar a generalização, porém empregam uma abordagem fixa, determinando previamente quais amostras serão usadas durante o treinamento.

Este estudo propõe uma abordagem de regularização utilizando reamostragem dinâmica dos subconjuntos ao longo do treinamento, permitindo que o modelo ajuste continuamente os exemplos usados em cada época. Essa abordagem se diferencia de métodos como *data augmentation*, pois, em vez de apenas transformar amostras existentes, modifica ativamente a composição do conjunto de treinamento ao longo do tempo, garantindo maior diversidade e adaptação do modelo.

4 Metodologia

O objetivo deste artigo é propor um método de regularização para melhorar a capacidade de generalização de modelos. A abordagem avalia se o uso de uma quantidade reduzida de dados, combinado com um maior número de épocas e reamostragem, pode superar o desempenho obtido com o uso do conjunto completo de dados em menos épocas. Ao realizar o treinamento em um conjunto de dados e a avaliação de desempenho em outro, buscamos avaliar se a proposta

impacta positivamente na capacidade do modelo em classificar as instâncias corretamente e generalizar para diferentes cenários [18].

A metodologia adotada neste estudo segue um modelo experimental controlado, no qual diferentes abordagens de regularização são comparadas com base em seu impacto na generalização do modelo. As variáveis do experimento incluem a quantidade de dados utilizados por época, o número total de épocas e o método de seleção dos subconjuntos. A hipótese central testada é que a reamostragem dinâmica de subconjuntos, ao longo de múltiplas épocas, pode melhorar a generalização do modelo sem comprometer sua capacidade de aprendizado. Essa abordagem busca reduzir o *overfitting* ao expor o modelo a diferentes subconjuntos de dados em cada iteração, forçando-o a aprender representações mais amplas em vez de memorizar padrões específicos.

A ideia principal deste artigo é a introdução da reamostragem dinâmica como estratégia de regularização. Diferentemente de métodos tradicionais que utilizam todo o conjunto de dados por época, a abordagem proposta trabalha com subconjuntos variáveis ao longo do treinamento. Essa estratégia busca evitar que o modelo memorize padrões específicos, forçando-o a aprender representações mais gerais dos dados. Para avaliar o impacto dessa técnica, foram desenhados quatro experimentos distintos, cada um explorando diferentes formas de manipulação dos dados.

A escolha da MobileNetV3-Small como modelo base se deve ao seu equilíbrio entre eficiência computacional e capacidade de aprendizado, sendo frequentemente utilizada em aplicações com restrições de recursos [13]. O número de épocas foi definido com base em experimentos preliminares e na hipótese de que um maior número de iterações, aliado à reamostragem dinâmica, poderia mitigar o *overfitting* em subconjuntos reduzidos.

Para os experimentos, foram utilizados dois conjuntos de dados conhecidos de imagens de estacionamento. O conjunto de dados PKLot [4], composto por 12.400 imagens (1280×720 pixels) capturadas de estacionamentos em dias ensolarados, nublados e chuvosos. Duas imagens extraídas deste conjunto de dados são mostradas na Figura 1. As imagens são organizadas em diretórios por condições climáticas e datas. Segmentando as imagens, pode-se obter cerca de 1.200.000 imagens de vagas de estacionamento rotuladas¹. O conjunto de dados CNRPark-EXT [2], com 4.278 imagens (1000×700 pixels) também capturadas de estacionamentos em dias ensolarados, nublados e chuvosos (veja duas amostras na Figura 2). As imagens são organizadas em diretórios por condições climáticas, datas e câmeras. Segmentando as imagens, obtém-se cerca de 150.000 imagens de vagas de estacionamento rotuladas. As imagens de ambos os conjuntos de dados tem seus dados anotados incluindo as coordenadas de cada vaga e seu rótulo (ocupada ou vazia).



Figura 1: Duas amostra do conjunto de dados PKLot.

¹O número de vagas de estacionamentos no conjunto de dados PKLot é superior ao do artigo original, uma vez que nossa equipe realizou a anotação de novas amostras.



Figura 2: Duas amostras do conjunto de dados CNRPark-EXT.

Para os experimentos, foi selecionada a rede MobileNetV3-Small pré-treinada nos pesos do ImageNet [6]. Para o treinamento, todas as camadas da rede foram descongeladas para atualização dos pesos, permitindo o *fine-tuning* da rede. A última camada foi alterada para a classificação binária. A função de perda utilizada foi a *BCEWithLogitsLoss*, com otimizador Adam e taxa de aprendizado inicial de 0,001. O *scheduler* de aprendizado utilizado foi o *ReduceLROnPlateau*, que ajusta a taxa de aprendizado quando a perda de validação não melhora após 20% das épocas. As imagens foram redimensionadas para 96×96 pixels, normalizadas com a média e desvio padrão do próprio conjunto dos dados e o *batch size* utilizado foi de 32.

A métrica *Equal Error Rate* (EER) é utilizada nos testes da rede treinada. O cálculo fornece um limiar de decisão que equilibra as taxas de falsos positivos (*False Positive Rate* – FPR) e falsos negativos (*False Negative Rate* – FNR). Para calcular o EER, utiliza-se a curva ROC (*Receiver Operating Characteristic*), que relaciona o FPR e a taxa de verdadeiros positivos (*True Positive Rate* – TPR) para diferentes valores de limiar de decisão. Esses limiares correspondem aos valores usados para converter as probabilidades previstas pelo modelo em classes binárias.

Inicialmente, a FNR é calculada como o complemento da TPR, conforme demonstrado na Equação 1:

$$FNR = 1 - TPR \quad (1)$$

O EER é definido como o ponto em que a FPR e a FNR são aproximadamente iguais, conforme mostrado na Equação 2:

$$FPR \approx FNR \quad (2)$$

Matematicamente, o EER é identificado ao minimizar a diferença absoluta entre FPR e FNR para os valores de limiar da curva ROC. Esse cálculo é formalizado na Equação 3:

$$EER = FPR [\operatorname{argmin} (|FPR - FNR|)] \quad (3)$$

O limiar correspondente ao EER, denominado threshold_{EER} , é determinado pela Equação 4:

$$\text{threshold}_{EER} = \text{thresholds} [\operatorname{argmin} (|FPR - FNR|)] \quad (4)$$

O valor calculado threshold_{EER} é utilizado como critério para a classificação no modelo, equilibrando os dois tipos de erro.

A estratégia usada para selecionar imagens para compor os conjuntos de treinamento e validação é que imagens do mesmo dia podem pertencer a apenas um desses conjuntos. Isso evita que imagens relacionadas ao mesmo carro estacionado no mesmo espaço por horas, mostrando apenas variações de luz, possam aparecer simultaneamente nos conjuntos de treinamento e validação.

Considerando a importância de evitar vieses nos resultados, a proposta neste protocolo é alocar 70% das imagens dos conjuntos de dados PKLot e CNRPark-EXT para treinamento e 30% para validação. É crucial ressaltar que cada conjunto de dados será tratado de forma independente, ou seja, não haverá combinação de imagens entre eles durante o treinamento. Para a fase de teste, cada modelo treinado em um conjunto de dados será avaliado no outro conjunto. Essa abordagem tenta garantir uma avaliação mais robusta e imparcial dos modelos, evitando que eles memorizem características específicas de um único conjunto.

O método proposto explora diferentes configurações de épocas e subconjuntos de dados durante o treinamento. Foram realizados quatro experimentos distintos, cada um utilizando diferentes estratégias de divisão e amostragem dos dados. Para cada experimento, três treinamentos foram conduzidos em ambos os conjuntos de dados (PKLot e CNRPark-EXT), totalizando 24 treinamentos. Os dados de validação permaneceram constantes e não foram manipulados durante os experimentos, garantindo comparabilidade entre as diferentes abordagens. A divisão dos dados está disposta na Tabela 1.

Tabela 1: Divisão dos dados em treinamento, validação e teste.

Cenário	Treino	Val	Teste
PKLot → CNRPark-EXT			
Ocupadas	358.119	185.465	81.982
Vazias	494.442	161.831	66.934
CNRPark-EXT → PKLot			
Ocupadas	57.072	24.910	543.584
Vazias	46.234	20.700	656.273

Primeiro experimento (Base): A rede foi treinada por 15 épocas utilizando todos os dados de cada conjunto. Esse experimento estabelece uma linha de referência sobre as outras propostas.

Segundo experimento (Dataloader): A rede foi treinada por 150 épocas utilizando apenas 10% dos dados de treino a cada época. Os dados são selecionados aleatoriamente a cada época.

Terceiro experimento (Amostragem Simples): Nesse experimento, a rede foi treinada por 150 épocas utilizando um método de amostragem específica. Em cada época, foi selecionada aleatoriamente uma única amostra representativa para cada veículo identificado por um código único (Carro ID). Por exemplo, uma imagem aleatória do Carro X e outra do Carro Y foram selecionadas, como ilustrado na Figura 3. Além disso, para as vagas vazias (sem veículos), as amostras foram escolhidas de forma aleatória, mantendo um balanceamento entre as classes ocupadas e vazias.

Quarto experimento (Amostragem Dupla): Esse experimento combina o método da Amostragem Simples com uma abordagem adicional para selecionar imagens de vagas vazias. A rede foi treinada por 150 épocas utilizando a Amostragem Simples para escolher imagens de vagas ocupadas, conforme descrito no experimento anterior. No entanto, para as vagas vazias, a seleção foi feita com base nos identificadores exclusivos da Vaga (Vaga ID) associados a cada veículo selecionado pela Amostragem Simples. Especificamente, para cada Carro ID escolhido, uma imagem correspondente a mesma Vaga foi selecionada. Por exemplo, uma amostra

aleatória da Vaga Z foi selecionada para o Carro X e outra para o Carro Y, como mostrado na Figura 3.



Figura 3: Exemplos de uma mesma vaga de estacionamento (conjunto de dados PKLot).

Para definir o melhor modelo treinado para realizar o teste foi utilizado o menor valor da *Loss* da etapa de validação. Para avaliar os resultados dos testes foi utilizada a acurácia, como mostrado na Equação 5:

$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

O hardware utilizado para treinar e testar as redes foi um processador Intel Xeon Silver 4214 de 2.20 GHz, com 24 núcleos, 125 GB de memória RAM, e duas GPUs NVIDIA RTX A5000 (24 GB).

5 Resultados e Discussões

Os experimentos realizados avaliaram a eficácia do método de regularização proposto, comparando diferentes abordagens de treinamento em dois conjuntos de dados: PKLot e CNRPark-EXT. As quatro configurações definidas nos métodos foram testadas: Base, Dataloader, Amostragem Simples e Amostragem Dupla. Conforme mostrado na Tabela 2, os experimentos destacam diferenças no comportamento das configurações avaliadas e evidenciam o impacto da estratégia de regularização na generalização dos modelos.

O treinamento com o conjunto de dados CNRPark-EXT e teste com o PKLot obteve os resultados abaixo:

- A configuração *Dataloader* obteve média de acurácia de 90,78%, superior a média de acurácia da *Base* utilizando menos imagens por época. A configuração *Amostragem Dupla* obteve média de acurácia de 91,63%, superior a média de acurácia da *Dataloader* utilizando a mesma quantidade de imagens por época. Em ambos os casos as médias das melhores épocas na validação é muito próxima. A *Amostragem Simples* manteve os mesmos resultados de acurácia que a *Base* usando menos imagens por época;

- Na Figura 4, é possível notar que em todos os experimentos o *Loss* do treinamento tem boa convergência para valores abaixo de 0,02. Nota-se muita oscilação do *Loss* nas primeiras épocas da validação e pode-se concluir que nos quatro experimentos existem problemas de *overfitting*.

O treinamento com o conjunto de dados PKLot e teste com o CNRPark-EXT obteve os resultados abaixo:

- A configuração *Dataloader* obteve média de acurácia de 95,05%, superior a média de acurácia da *Base* utilizando menos imagens por época. A configuração *Amostragem Dupla* obteve média de acurácia de 94,75%, muito próxima a acurácia da *Dataloader* utilizando menos imagens por época com apenas oito épocas de diferença. A *Amostragem Simples* manteve os mesmos resultados de acurácia que a *Base* usando menos imagens por época;
- Na Figura 5, é possível notar que em todos os experimentos a *Loss* do treinamento tem boa convergência para valores abaixo de 0,02. Observa-se boa estabilidade no *Loss* da validação com pouca oscilação. Destaque para os experimentos *Amostragem Simples* e *Amostragem Dupla* que têm menos indícios de *overfitting*.

Os modelos treinados com o conjunto de dados CNRPark-EXT apresentaram desempenho inferior de até seis pontos percentuais na acurácia. Esse resultado pode ser atribuído a menor quantidade de dados, o que tornou o conjunto mais suscetível ao *overfitting*, evidenciado pelo maior crescimento da *Loss* de validação. A maior diversidade e quantidade de dados no PKLot contribuíram para uma *Loss* de validação mais estável, destacando as configurações de *Amostragem Simples* e *Amostragem Dupla* como as mais robustas.

Com o conjunto de dados do CNRPark-EXT, o modelo treinado com a configuração *Base* obteve uma convergência média na época 2,33 processando cerca de 247.693 amostras (resultado de $2,33 \times 103.306$). A configuração *Dataloader*, por sua vez, alcançou uma acurácia média superior ao treinar por 5 épocas, processando apenas 53.155 amostras (resultado de 5×10.331), uma redução de aproximadamente 78,54% no número de amostras processadas. A configuração *Amostragem Simples* manteve a acurácia da *Base* e apresentou uma leve redução em relação ao *Dataloader*, processando 235.447 amostras (resultado de $22,33 \times 10.544$). Isso representa 4,94% menos amostras que a *Base*, mas a diferença em relação ao *Dataloader* foi de 77,42% superior em quantidade de amostras. Já a configuração *Amostragem Dupla* alcançou resultados semelhantes ao *Dataloader*, processando 13.588 amostras a mais, com uma diferença superior de 0,85% em acurácia.

Com o conjunto de dados PKLot, a configuração *Base* convergiu em uma média de 3,33 épocas, processando aproximadamente 2.839.028 amostras (resultado de $3,33 \times 852.561$). Por outro lado, a configuração *Dataloader* alcançou resultados superiores em termos de acurácia média ao treinar por 22,33 épocas, processando apenas 1.903.766 amostras (resultado de $22,33 \times 85.256$). Isso representa uma redução de 32,94% no número de amostras processadas em comparação à configuração *Base*. Embora as configurações *Amostragem Simples* e *Amostragem Dupla* tenham apresentado acurácia média geral menor do que a configuração *Dataloader*, e praticamente equivalente à configuração *Base*, ambas utilizaram um número consideravelmente menor de amostras. A *Amostragem Simples* em

16,67 épocas processou cerca de 353.524 amostras (resultado de $16,67 \times 21.190$), obtendo uma redução de 87,56% em relação à configuração *Base* e de 81,44% em relação à configuração *Dataloader*. *Amostragem Dupla* em 30,33 épocas processou aproximadamente 642.693 amostras (resultado de $30,33 \times 21.190$), obtendo uma redução de 77,36% em relação à configuração *Base* e de 66,24% em relação à configuração *Dataloader*.

Ao utilizar as médias ponderadas entre os experimentos, observou-se que o experimento “*Amostragem Dupla*” obteve um aumento de 3% na acurácia, enquanto utilizou 96,44% menos imagens em comparação ao experimento “*Base*”. Já o experimento “*Amostragem Simples*”, quando comparado ao experimento “*Base*”, manteve a acurácia em 88%, mas resultou em uma redução de 97,23% na quantidade de imagens por época. Esses resultados demonstram que, ao adotar abordagens de amostragem mais eficientes, é possível otimizar o uso dos dados sem comprometer significativamente a performance do modelo.

Embora os experimentos tenham variado no número total de épocas para treinamento, observa-se que as melhores métricas foram alcançadas nas etapas iniciais do processo, independentemente do método utilizado. Apesar de pequenas diferenças nos valores de *Loss*, todas as configurações apresentaram desempenhos similares em termos de acurácia, sugerindo uma convergência consistente. Isso indica que estratégias como *early stopping* [3] poderiam ser aplicadas para interromper o treinamento antecipadamente, economizando tempo e recursos computacionais sem comprometer a qualidade dos resultados.

Para garantir uma comparação justa, o *early stopping* não foi utilizado, permitindo que todas as redes processassem a mesma quantidade de dados. Embora *Amostragem Simples* e *Amostragem Dupla* apresentassem leve diferença na acurácia, ambas usaram significativamente menos amostras, sendo alternativas viáveis para cenários com restrições computacionais. Essas estratégias mantiveram boa convergência, sugerindo potencial para otimizar recursos em aplicações reais. A comparação com trabalhos relacionados não foi realizada, pois não fazia parte do escopo deste estudo.

6 Limitações e Trabalhos Futuros

Esta seção discute as principais limitações deste artigo e sugere direções para pesquisas futuras.

Generalização: Embora os experimentos tenham sido conduzidos em dois conjuntos de dados robustos (PKLot e CNRPark-EXT), os resultados podem não ser diretamente aplicáveis a outros cenários urbanos ou condições mais diversificadas. Para validar a abordagem proposta, estudos futuros devem explorar bases de dados adicionais que representem contextos distintos, como vias públicas movimentadas, aeroportos ou centros comerciais.

Impacto da reamostragem no tempo de treinamento: A estratégia baseada em subconjuntos menores com reamostragem dinâmica aumenta o número de épocas necessárias para a convergência do modelo. Apesar dos benefícios observados em termos de generalização, o impacto dessa abordagem no tempo total de treinamento e no uso de recursos computacionais não foi explorado em profundidade. Uma possibilidade para mitigar essa questão é a investigação de técnicas como o *early stopping* [3], que pode interromper o treinamento quando a *loss* de validação estabilizar

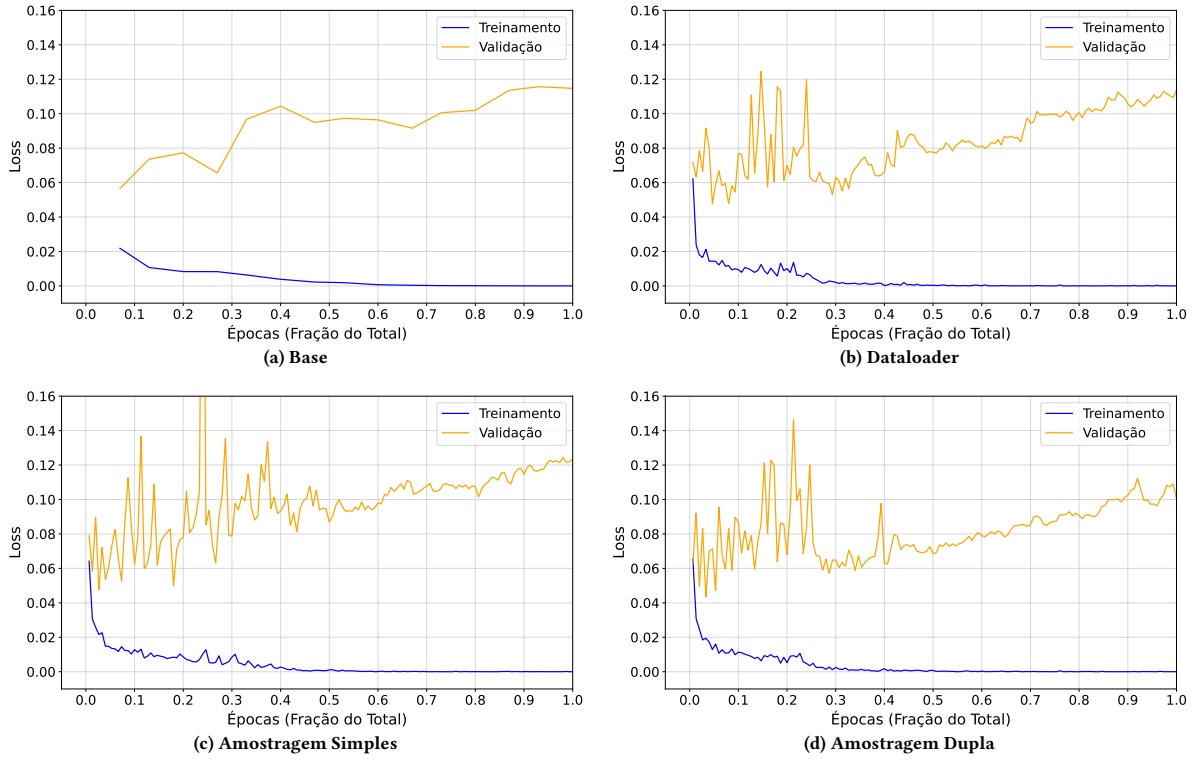


Figura 4: Curvas médias da função de perda ($Loss$) durante os treinamentos realizados com o conjunto de dados CNRPark-EXT.

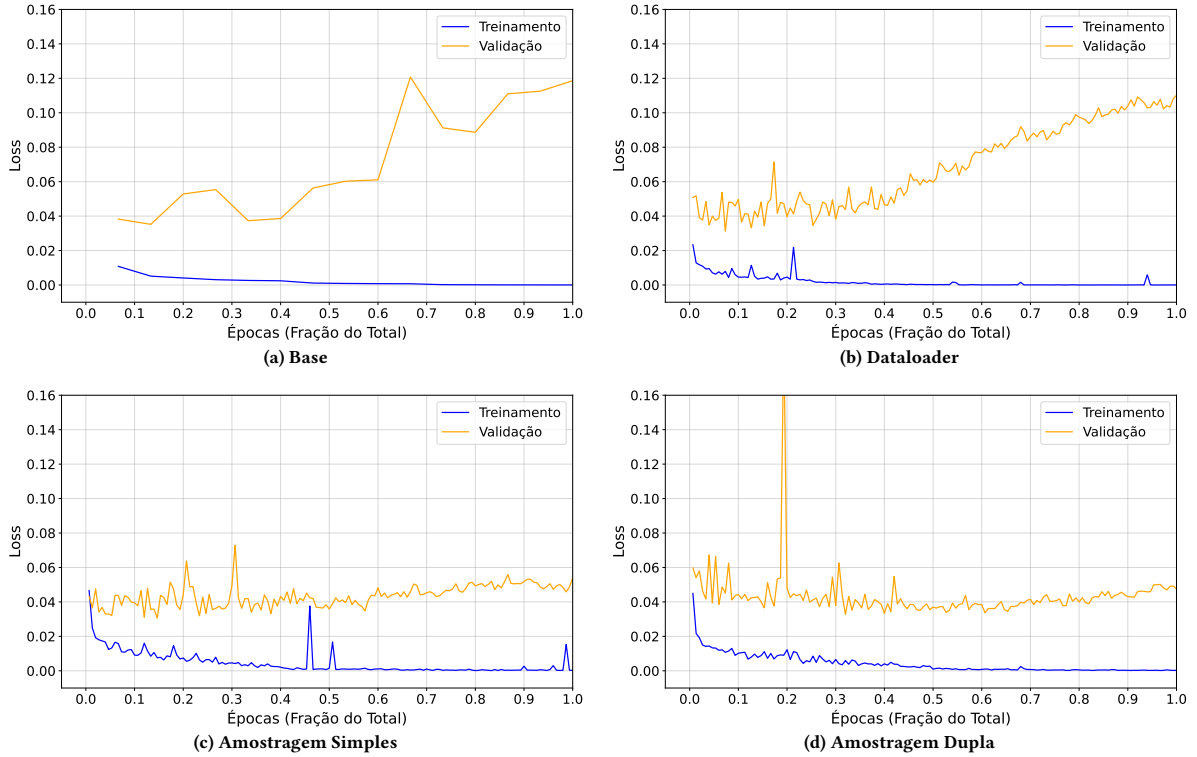


Figura 5: Curvas médias da função de perda ($Loss$) durante os treinamentos realizados com o conjunto de dados PKLot.

Tabela 2: Resultados dos experimentos.

Experimento	Treino	Melhor Época	AUC	Acurácia Teste (%)	Imagens Época
Treino com CNRPark-EXT	1	1,00	0,9987	91,62	103.306
Teste com PKLot	2	2,00	0,9986	88,34	103.306
Experimento Base	3	4,00	0,9988	84,38	103.306
	Média	2,33	0,9987	88,11	103.306

Treino com PKLot	1	6,00	0,9997	95,63	852.561
Teste com CNRPark-EXT	2	2,00	0,9996	94,73	852.561
Experimento Base	3	2,00	0,9996	93,18	852.561
	Média	3,33	0,9996	94,51	852.561

Média Ponderada Experimento Base		—	0,9995	88,82	544.044
=====					
Treino com CNRPark-EXT	1	3,00	0,9980	92,12	10.331
Teste com PKLot	2	5,00	0,9982	88,36	10.331
Experimento Dataloader	3	7,00	0,9987	91,86	10.331
	Média	5,00	0,9983	90,78	10.331

Treino com PKLot	1	23,00	0,9995	94,74	85.256
Teste com CNRPark-EXT	2	6,00	0,9996	95,18	85.256
Experimento Dataloader	3	38,00	0,9997	95,24	85.256
	Média	22,33	0,9996	95,05	85.256

Média Ponderada Experimento Dataloader		—	0,9995	91,25	71.550
=====					
Treino com CNRPark-EXT	1	19,00	0,9987	87,47	10.544
Teste com PKLot	2	15,00	0,9992	91,48	10.544
Experimento Amostragem Simples	3	33,00	0,9992	85,23	10.544
	Média	22,33	0,9990	88,06	10.544

Treino com PKLot	1	7,00	0,9996	94,52	21.190
Teste com CNRPark-EXT	2	28,00	0,9996	93,38	21.190
Experimento Amostragem Simples	3	15,00	0,9995	95,33	21.190
	Média	16,67	0,9995	94,41	21.190

Média Ponderada Experimento Amostragem Simples		—	0,9994	88,76	15.094
=====					
Treino com CNRPark-EXT	1	5,00	0,9985	91,59	10.544
Teste com PKLot	2	6,00	0,9988	90,83	10.544
Experimento Amostragem Dupla	3	8,00	0,9987	92,47	10.544
	Média	6,33	0,9987	91,63	10.544

Treino com PKLot	1	34,00	0,9996	94,88	21.190
Teste com CNRPark-EXT	2	14,00	0,9995	94,38	21.190
Experimento Amostragem Dupla	3	43,00	0,9996	95,00	21.190
	Média	30,33	0,9995	94,75	21.190

Média Ponderada Experimento Amostragem Dupla		—	0,9994	91,97	19.351

ou começar a aumentar, reduzindo o custo computacional sem comprometer a performance do modelo.

Limitação dos modelos utilizados: O estudo utilizou a rede MobileNetV3-Small devido ao seu equilíbrio entre eficiência computacional e capacidade de aprendizado. No entanto, outras arquiteturas, especialmente aquelas voltadas para regularização e

aprendizado eficiente, podem ser avaliadas em estudos futuros para verificar se apresentam ganhos adicionais em termos de generalização e desempenho.

Tendência ao *overfitting* em cenários de menor diversidade de dados: Os experimentos indicaram que o treinamento baseado no conjunto de dados CNRPark-EXT apresentou maior suscetibilidade ao *overfitting*. Esse efeito pode estar relacionado à menor diversidade de imagens e à quantidade reduzida de amostras disponíveis. Portanto, novas investigações podem explorar estratégias complementares de regularização ou aumento da variabilidade dos dados de entrada para mitigar esse efeito.

Estudos futuros podem avaliar o impacto da reamostragem dinâmica em outras arquiteturas e comparar com abordagens da literatura.

7 Conclusões

Os experimentos realizados demonstraram que o método de regularização proposto, baseado na utilização de subconjuntos menores de dados com reamostragem, contribuiu para melhorar a capacidade de generalização dos modelos.

Os resultados indicam que a menor quantidade de amostras no conjunto de dados CNRPark-EXT resultou em maior suscetibilidade ao *overfitting*, conforme evidenciado pelo crescimento da *Loss* de validação em todas as configurações, apesar da boa convergência da *Loss* de treinamento. Por outro lado, o conjunto de dados PKLot, por conter maior volume de amostras e maior variabilidade, apresentou menor oscilação da *Loss* de validação e acurácias superiores, reforçando a importância da qualidade e diversidade do conjunto de dados para o desempenho do modelo.

A análise das métricas revelou que os melhores níveis de desempenho foram consistentemente alcançados nas primeiras épocas de treinamento, independentemente do método utilizado. Esse resultado sugere que a amostragem dinâmica não apenas otimiza o aprendizado inicial, mas também mantém a eficiência geral do treinamento sem comprometer a acurácia final.

Os achados deste estudo indicam que a estratégia proposta pode ser viável em cenários que demandam eficiência computacional e boa generalização, especialmente quando aplicadas a conjuntos de dados mais diversos e volumosos.

Agradecimentos

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

Referências

- [1] Laith Alzubaidi, Jing Zhang, Ali J. Humaidi, et al. 2021. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data* 8, 1 (2021), 53. <https://doi.org/10.1186/s40537-021-00444-8>
- [2] G. Amato, F. Carrara, F. Falchi, C. Gennaro, C. Meghini, and C. Vairo. 2017. Deep learning for decentralized parking lot occupancy detection. *Expert Systems with Applications* 72 (2017), 327–334. <https://doi.org/10.1016/j.eswa.2016.10.055>
- [3] Yingbin Bai et al. 2021. Understanding and Improving Early Stopping for Learning with Noisy Labels. In *International Conference on Neural Information Processing Systems*. 24392–24403. <https://doi.org/10.48550/arXiv.2106.15853>
- [4] P. R. De Almeida, L. S. Oliveira, A. S. Brito Jr, E. J. Silva Jr, and A. L. Koerich. 2015. PKLot–A robust dataset for parking lot classification. *Expert Systems with Applications* 42, 11 (2015), 4937–4949. <https://doi.org/10.1016/j.eswa.2015.02.009>
- [5] Paulo Ricardo Lisboa de Almeida et al. 2022. A systematic review on computer vision-based parking lot management applied on public datasets. *Expert Systems with Applications* 198 (2022), 116731. <https://doi.org/10.1016/j.eswa.2022.116731>
- [6] Jia Deng et al. 2009. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 248–255. <https://doi.org/10.1109/CVPR.2009.5206848>
- [7] Tiago Dias et al. 2023. From data to action: Exploring AI and IoT-driven solutions for smarter cities. In *International Symposium on Distributed Computing and Artificial Intelligence*. 44–53. https://doi.org/10.1007/978-3-031-38333-5_5
- [8] V. Feldman and C. Zhang. 2020. What Neural Networks Memorize and Why: Discovering the Long Tail via Influence Estimation. In *International Conference on Neural Information Processing Systems (NeurIPS)*, Vol. 33. 2881–2891.
- [9] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.
- [10] Jerome H Friedman. 2002. Stochastic gradient boosting. *Computational statistics & data analysis* 38, 4 (2002), 367–378.
- [11] J. Gu et al. 2018. Recent advances in convolutional neural networks. *Pattern Recognition* 77 (2018), 354–377. <https://doi.org/10.1016/j.patcog.2017.10.013>
- [12] Arthur E Hoerl and Robert W Kennard. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* 12, 1 (1970), 55–67.
- [13] Andrew Howard et al. 2019. Searching for MobileNetV3. In *IEEE/CVF International Conference on Computer Vision (CVPR)*. 1314–1324.
- [14] Sergey Ioffe. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).
- [15] Aleksandra Kadra, Marius Lindauer, Frank Hutter, and Josif Grabocka. 2021. Well-tuned Simple Nets Excel on Tabular Datasets. In *International Conference on Neural Information Processing Systems (NeurIPS)*. 1–23.
- [16] Krishnateja Killamsetty et al. 2021. GLISTER: Generalization based data subset selection for efficient and robust learning. *AAAI Conference on Artificial Intelligence* 35, 9 (2021), 8110–8118. <https://doi.org/10.1609/aaai.v35i9.16988>
- [17] Ron Kohavi. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artificial Intelligence (IJCAI)*. 1137–1143.
- [18] R. Laroca et al. 2022. On the Cross-Dataset Generalization in License Plate Recognition. In *International Conference on Computer Vision Theory and Applications (VISAPP)*. 166–178. <https://doi.org/10.5220/0010846800003124>
- [19] R. Laroca et al. 2023. Do We Train on Test Data? The Impact of Near-Duplicates on License Plate Recognition. In *International Joint Conference on Neural Networks (IJCNN)*. 1–8. <https://doi.org/10.1109/IJCNN54540.2023.10191584>
- [20] R. Moradi, R. Berangi, and B. Minaei. 2020. A survey of regularization strategies for deep models. *Artificial Intelligence Review* 53, 6 (2020), 3947–3986. <https://doi.org/10.1007/s10462-019-09784-7>
- [21] A. N. Muhammad, A. M. Aseere, H. Chiroma, H. Shah, A. Y. Gital, and I. A. T. Hashem. 2021. Deep learning application in smart cities: recent development, taxonomy, challenges and research prospects. *Neural Computing and Applications* 33 (2021), 2973–3009. <https://doi.org/10.1007/s00521-020-05151-8>
- [22] C. F. G. D. Santos and J. P. Papa. 2022. Avoiding overfitting: A survey on regularization methods for convolutional neural networks. *ACM Computing Surveys (CSUR)* 54, 10s (2022), 1–25. <https://doi.org/10.1145/3510413>
- [23] Connor Shorten and Taghi M. Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data* 6 (2019), 60. <https://doi.org/10.1186/s40537-019-0197-0>
- [24] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.
- [25] Y. Tian and Y. Zhang. 2022. A comprehensive survey on regularization strategies in machine learning. *Information Fusion* 80 (2022), 146–166. <https://doi.org/10.1016/j.inffus.2021.11.005>
- [26] Shuguan Yang et al. 2019. A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources. *Transportation Research Part C: Emerging Technologies* 107 (2019), 248–265. <https://doi.org/10.1016/j.trc.2019.08.010>
- [27] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. 2014. Internet of things for smart cities. *IEEE Internet of Things Journal* 1, 1 (2014), 22–32. <https://doi.org/10.1109/JIOT.2014.2306328>