

# Aprimorando o Desempenho da SqueezeNet na Classificação de Imagens

1<sup>st</sup> Murilo Costa

Engenharia de Software

Pontfícia Universidade Católica de Minas Gerais

Belo Horizonte, Brazil

murilo.costa@sga.pucminas.br

2<sup>nd</sup> Marcos Felipe

Engenharia de Software

Pontfícia Universidade Católica de Minas Gerais

Belo Horizonte, Brazil

marcos.muniz@sga.pucminas.br

3<sup>rd</sup> Davi Santos

Engenharia de Software

Pontfícia Universidade Católica de Minas Gerais

Belo Horizonte, Brazil

davi.santos@sga.pucminas.br

**Abstract**—Este trabalho investiga e avalia um conjunto de estratégias de otimização para aprimorar o desempenho da rede neural SqueezeNet, um modelo computacionalmente eficiente, em cenários de classificação de imagens com alta complexidade visual. A metodologia consistiu em um processo experimental iterativo, aplicando técnicas como *data augmentation*, modificações arquiteturais (módulos de atenção, conexões residuais), e uma combinação final de otimizações abrangentes (e.g., *Depthwise Separable Convolutions*) com um regime de treinamento avançado. Os resultados demonstram que, enquanto otimizações focadas nos dados elevaram a acurácia de teste de 74,09% para 81,90%, modificações arquiteturais isoladas se mostraram contraproducentes. O desempenho máximo de 86,86% foi alcançado apenas através da abordagem holística final. Conclui-se que a otimização eficaz de modelos eficientes para tarefas complexas depende de uma abordagem sinérgica que co-otimize a arquitetura e o protocolo de treinamento, em vez de modificações pontuais isoladas.

**Index Terms**—SqueezeNet, CNN, classificação de imagens, deep learning

## I. INTRODUÇÃO

As *Convolutional Neural Networks* (CNNs) são um tipo de modelo de aprendizado profundo empregado em tarefas de visão computacional, como a classificação de imagens [1]. Dentre as diversas arquiteturas de CNNs, modelos como a SqueezeNet [2] se caracterizam pela eficiência computacional, apresentando um número reduzido de parâmetros, o que os torna adequados para aplicação em ambientes com recursos limitados, como dispositivos móveis ou embarcados.

A aplicação da SqueezeNet em *datasets* personalizados, que se afastam das características de *benchmarks* como o CIFAR-10 [3], pode revelar limitações no desempenho do modelo. Imagens com dimensões superiores às de 32×32 e com fundos ricos em detalhes, por exemplo, podem comprometer a capacidade de generalização da rede, afetando a precisão na distinção entre classes visualmente semelhantes, como observado na confusão entre cães e gatos em experimentações. **A dificuldade em alcançar um desempenho satisfatório da SqueezeNet nestes cenários mais complexos, mantendo**

**sua eficiência computacional**, constitui o problema central abordado por esta pesquisa.

A capacidade de modelos eficientes como a SqueezeNet em manter um alto desempenho em uma gama mais ampla e complexa de *datasets* permite sua aplicação prática em mais contextos. Aprimorar a acurácia e a robustez desses modelos em cenários que se aproximam de desafios reais, como os encontrados em imagens com fundos detalhados e variações de resolução, contribui para viabilizar sua utilização em sistemas embarcados ou com restrições computacionais, onde a eficiência no uso de recursos e a precisão na classificação são requisitos mandatórios.

Nesse sentido, este trabalho propõe e avalia um conjunto de modificações na arquitetura da SqueezeNet, com o objetivo de otimizar seu desempenho em tarefas de classificação com imagens de maior complexidade. Ao final, apresentamos uma análise comparativa que identifica as alterações mais eficazes para o aprimoramento da robustez do modelo, oferecendo um caminho para sua adaptação a cenários mais desafiadores.

## II. MÉTODOS

Visando aprimorar o desempenho da arquitetura SqueezeNet para além do seu estado de referência, especialmente em *datasets* com maior complexidade visual, um conjunto de estratégias de otimização foi empregado. Estas estratégias foram selecionadas com base em práticas estabelecidas para o treinamento e a configuração de CNNs. As principais frentes de otimização investigadas podem ser agrupadas em: *I)* refinamento do tratamento de dados e do processo inicial de treinamento; *II)* modificações na arquitetura da rede para incorporar mecanismos de aprendizado; *III)* e a aplicação de técnicas avançadas para a condução do treinamento.

### A. Ambiente de Execução

Os experimentos foram realizados na plataforma *Google Colaboratory* (Colab), configurada com (Graphic Processing Unities) (GPUs) do tipo T4. Utilizou-se Python na versão 3.9,

juntamente com as bibliotecas PyTorch para a construção e treinamento das redes neurais, Torchvision para o carregamento e pré-processamento de *datasets*, e NumPy e Matplotlib como ferramentas de suporte.

### B. Modelo Base

Para otimizar a forma como o modelo interage com os dados de entrada e o aprendizado inicial, foram consideradas as seguintes técnicas. A técnica de *data augmentation* foi empregada como uma estratégia para expandir artificialmente a diversidade dos conjuntos de treinamento. Esta técnica, que envolve a aplicação de transformações geométricas e de cor às imagens existentes (como rotações, translações, e ajustes de brilho), justifica-se por sua capacidade de expor o modelo a uma maior variabilidade de exemplos, o que pode auxiliar na redução do *overfitting* e no desenvolvimento de representações de características generalizáveis. Adicionalmente, explorou-se o *fine-tuning*, um processo de ajuste fino dos pesos de uma rede previamente treinada ou a continuação do treinamento sob condições específicas. A justificativa para o seu uso reside na capacidade de adaptar o modelo a particularidades de um novo *dataset* ou de direcionar o aprendizado para classes que apresentam maior dificuldade de classificação, permitindo também otimizações na taxa de aprendizado para diferentes estágios do treinamento. O pré-processamento do *dataset* também incluiu a conversão de imagens em escala de cinza para o formato RGB quando necessário, garantindo a padronização da entrada para a rede neural e o aproveitamento da informação de cor, caso presente.

### C. Estratégias de Otimização

Para aprimorar o desempenho da SqueezeNet, diferentes estratégias de otimização foram concebidas e avaliadas experimentalmente. Cada estratégia, descrita a seguir, representa uma abordagem distinta que combina um conjunto de técnicas com justificativas específicas para a melhoria do modelo.

1) *Otimização Focada em Dados e Treinamento*: Esta abordagem inicial buscou aprimorar o desempenho sem alterar a arquitetura central da SqueezeNet. Para isso, empregou-se o *data augmentation*, que expande artificialmente a diversidade do conjunto de treinamento através de transformações visuais, com a justificativa de melhorar a generalização e reduzir o *overfitting*. Concomitantemente, aplicou-se o *fine-tuning*, um processo de ajuste fino do treinamento para adaptar o modelo às particularidades do *dataset* e direcionar o aprendizado para classes com maior dificuldade de classificação.

2) *Incorporação de Mecanismo de Atenção Simples*: Esta estratégia consistiu na modificação da arquitetura SqueezeNet pela introdução de um módulo de *Simple Attention* após cada bloco convolucional *Fire*. A justificativa para esta abordagem foi testar a hipótese de que um mecanismo de atenção capacitaria o modelo a ponderar e focar dinamicamente nas características mais discriminativas das imagens, mitigando assim o impacto de ruídos visuais, como fundos complexos.

3) *Adição de Conexões Residuais e Regularização*: Nesta abordagem, a arquitetura foi modificada para incorporar conexões residuais, inspiradas na ResNet. A justificativa para esta estratégia foi avaliar se os "atalhos" de gradiente poderiam facilitar o treinamento de representações mais complexas e mitigar o problema do *vanishing gradient*. Adicionalmente, a técnica de *dropout* foi aplicada como forma de regularização, com o intuito de prevenir o *overfitting* ao reduzir a co-adaptação dos neurônios.

4) *Abordagem de Otimização Holística*: Esta estratégia final representou uma reestruturação completa e sinérgica tanto da arquitetura quanto do protocolo de treinamento. As modificações arquiteturais incluíram a substituição da atenção simples por um *Channel Attention* mais sofisticado, o uso de *Depthwise Separable Convolutions* para maior eficiência, e a adoção da função de ativação SiLU. O regime de treinamento foi igualmente aprimorado com técnicas avançadas como o *scheduler OneCycleLR* para otimizar a taxa de aprendizado, a regularização por *label smoothing* para melhorar a generalização, o *gradient clipping* para estabilidade e a inicialização de pesos *Kaiming* para uma convergência mais robusta. A justificativa para esta abordagem foi testar o impacto combinado de um conjunto de otimizações de ponta, tratando a arquitetura e o treinamento como um sistema co-otimizado.

## III. RESULTADOS

Esta seção apresenta os resultados da sequência de experimentos realizados para a otimização da SqueezeNet. A apresentação inicia-se com a descrição do *dataset* utilizado para contextualizar os desafios e, em seguida, detalha o desempenho de cada etapa experimental em ordem cronológica, desde o modelo base até a versão final otimizada.

### A. Dataset

O CIFAR-10 é um *dataset* público amplamente reconhecido e utilizado como *benchmark* na comunidade de visão computacional. Ele é composto por 60.000 imagens coloridas de dimensões 32×32 *pixels*, divididas em 10 classes mutuamente

TABLE I  
SUMÁRIO COMPARATIVO DA ACURÁCIA DE TESTE POR ETAPA EXPERIMENTAL.

Etapa Experimental	Principais Modificações	Acurácia de Teste (%)
1. Modelo Base	SqueezeNet Original	74,09
2. Otimização Inicial	<i>Fine-Tuning + Data Augmentation</i>	81,90
3. Atenção Simples	Adição de Módulo <i>Simple Attention</i>	67,23
4. Conexões Residuais	Adição de Blocos Residuais à arquitetura com atenção	67,96
5. Otimização Final	Arquitetura e Treinamento Avançados (Abordagem Holística)	<b>86,86</b>

exclusivas, com 6.000 imagens por classe. Destas, 50.000 são destinadas ao treinamento e 10.000 para teste. A escolha do CIFAR-10 para este estudo justifica-se por sua capacidade de fornecer uma avaliação inicial do desempenho do modelo SqueezeNet e permitir comparações com resultados estabelecidos na literatura.

## B. Experimentações

Os resultados quantitativos obtidos ao longo da sequência de experimentos são detalhados nesta seção e sumarizados na Tabela I, que apresenta a acurácia de teste final para cada etapa principal de otimização.

A investigação iniciou-se com o treinamento da arquitetura SqueezeNet original para estabelecer um desempenho de referência (*baseline*). Este primeiro experimento, interrompido por *early stopping*, alcançou uma acurácia de teste de 74,09%. Em seguida, na segunda etapa, foram aplicadas estratégias de otimização focadas no processo de treinamento e nos dados, como *fine-tuning* e *data augmentation*, o que resultou em uma melhora substancial no desempenho, atingindo uma acurácia de teste de 81,90%. A terceira etapa experimental consistiu na adição de um módulo de *Simple Attention* à arquitetura; contudo, esta modificação, contrariamente à expectativa, levou a uma queda significativa no desempenho, resultando em uma acurácia de 67,23%. Na quarta etapa, foram incorporadas conexões residuais à arquitetura que já continha o módulo de atenção. Esta configuração produziu uma acurácia de 67,96%, representando uma melhora apenas marginal sobre a etapa anterior e permanecendo bem abaixo do desempenho obtido na segunda etapa. Por fim, o último experimento empregou um conjunto abrangente de otimizações avançadas, tanto na arquitetura (e.g., *Channel Attention*, *Depthwise Separable Convolutions*, SiLU) quanto no protocolo de treinamento (e.g., *OneCycleLR*, *label smoothing*). Esta abordagem integrada alcançou a maior acurácia de todo o estudo, com 86,86% no conjunto de teste, demonstrando a eficácia da combinação das otimizações e representando um aumento de aproximadamente 12,77 pontos percentuais sobre o *baseline* original.

## CONCLUSÕES

O presente trabalho investigou um processo experimental iterativo para otimizar o desempenho da arquitetura SqueezeNet, um modelo computacionalmente eficiente, visando aprimorar sua capacidade de classificação em *datasets* com alta complexidade visual. A análise dos resultados obtidos ao longo das diferentes etapas de otimização permite extrair conclusões importantes sobre as estratégias de aprimoramento de redes neurais. A investigação partiu de um desempenho de referência de 74,09% de acurácia, estabelecido pelo modelo SqueezeNet original. A primeira frente de otimização, focada em estratégias de tratamento de dados como *fine-tuning* e *data augmentation*, demonstrou ser uma abordagem de alto impacto inicial, elevando a acurácia para 81,90%. Este resultado evidencia que a otimização do processo de treinamento e da diversidade dos dados é uma etapa fundamental e eficaz antes de se proceder a modificações estruturais complexas.

Contudo, a pesquisa revelou que modificações arquiteturais isoladas podem ser contraproducentes. A introdução de um módulo de atenção simples e, posteriormente, de conexões residuais, sem um protocolo de treinamento que suportasse o aumento da complexidade da rede, resultou em uma queda no desempenho, com acurácias em torno de 67% a 68%. Esta observação crítica sugere que a simples adição de componentes teoricamente benéficos é ineficaz e pode até prejudicar o processo de aprendizado se não for realizada de maneira integrada, pois a maior complexidade do modelo não foi acompanhada por uma estratégia de treinamento capaz de explorá-la.

O desempenho máximo de 86,86% de acurácia foi alcançado apenas na etapa final, que combinou múltiplas otimizações avançadas. O sucesso desta fase não pode ser atribuído a um único fator, mas sim à sinergia entre as modificações arquiteturais sofisticadas (como *Channel Attention*, *Depthwise Separable Convolutions* e a ativação SiLU) e um regime de treinamento igualmente robusto (incluindo o *scheduler OneCycleLR* e a inicialização de pesos *Kaiming*). A eficácia desta abordagem holística, em nítido contraste com o fracasso das modificações isoladas, constitui a principal conclusão deste estudo. Portanto, conclui-se que a otimização de modelos de aprendizado profundo eficientes para tarefas desafiadoras depende de uma abordagem integrada que co-otimize a arquitetura da rede e seu protocolo de treinamento, tratando-os como um sistema unificado e interdependente para alcançar ganhos de desempenho significativos.

## REFERENCES

- [1] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791. keywords: Neural networks;Pattern recognition;Machine learning;Optical character recognition software;Character recognition;Feature extraction;Multi-layer neural network;Optical computing;Hidden Markov models;Principal component analysis.
- [2] Iandola, Forrest N., et al. "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and 0.5 MB model size." *arXiv preprint arXiv:1602.07360* (2016).
- [3] A. Krizhevsky, "Learning Multiple Layers of Features from Tiny Images," Tech. Rep., University of Toronto, Toronto, ON, Canada, 2009.
- [4] Targ, Sasha, Diogo Almeida, and Kevin Lyman. "Resnet in resnet: Generalizing residual architectures." *arXiv preprint arXiv:1603.08029* (2016).