Avaliação de Classificadores de Imagem Multi-rótulo

Igor Goulart Cabral

[1]Programa de Pós-Graduação em Informática
Pontifícia Universidade Católica de Minas Gerais (PUC Minas)

icabral@sga.pucminas.br

Resumo. *Este trabalho apresenta a reprodução parcial do método LV-CIT. Utilizando redes profundas (ResNet18) e os conjuntos de dados VOC2007 e COCO2014, foram geradas imagens compostas por covering arrays para validação da robustez do experimento. Toda a execução foi realizada em ambiente de nuvem (Google Colab). A acurácia atingiu valores próximos de 100% nas primeiras iterações, evidenciando a estabilidade do método. Limitações de infraestrutura e dependências impediram a expansão imediata para múltiplos modelos e datasets adicionais. Os resultados parciais indicam que o LV-CIT pode ser viável para pipelines de testes black-box de classificadores de imagem. Propõe-se, como continuidade, a automação total do fluxo e testes mais extensos em clusters de GPU.*

1. Introdução

As redes neurais compõem uma parte importante no campo da Inteligência artificial. Essas arquiteturas são essenciais para a evolução dos algoritmos de aprendizado profundo, contribuindo para o progresso importante na análise, validação e implementação de dados complexos. As metodologias utilizadas, tem sido empregadas em diversos domínios como saúde, educação, entretenimento e finanças, demonstrando sua capacidade de transformar a sociedade.

A inteligência artificial (IA) é uma área dedicada à criação de algoritmos capazes de aprender. Ele aborda diversas metodologias baseadas na lógica, algoritmos de busca e estruturas de raciocínio probabilístico. [Prince 2023]. O Machine Learning representa um ramo onde os modelos são desenvolvidos para aprender a partir de dados, ajustando parâmetros matemáticos para tomar decisões..[Prince 2023]

No campo do Machine Learning, as redes neurais profundas se destacam, e a metodologia empregada em seu desenvolvimento é chamada Deep Learning. Estes modelos têm demonstrado uma capacidade incrível de lidar com cenários complexos, sendo amplamente aplicados em tarefas como tradução automática de idiomas, reconhecimento e busca de imagens, e no desenvolvimento de assistentes virtuais e chatbots.[Prince 2023]

A avaliação de sistemas computacionais, incluindo aqueles que usam redes neurais profundas, envolvem diversas metodologias de teste white-box e black-box. O teste white-box busca pelo conhecimento da estrutura interna da arquitetura ou de seus dados de treinamento para elaborar cenários, e muitas vezes, essa abordagem é inviável devido a restrições de acesso e preocupações com a privacidade dos dados usados.[Wang et al. 2024]. Os testes de black-box, no entanto estão ganhando atenção no mundo acadêmico, uma vez que não tem a necessidade de acesso as arquiteturas ou aos dados de treino, gerando grandes estudos sobre estratégias para sua aplicação.[Wang et al. 2024]

Este trabalho tem como objetivo principal investigar a aplicabilidade do método LV-CIT, proposto por [Wang et al. 2024], que se destaca como uma abordagem eficiente para testar classificadores de imagem baseados em DNNs, em especial na avaliação da capacidade desses modelos em lidar com correlações entre múltiplos objetos. Portanto, este estudo visa replicar e estender o experimento original de [Wang et al. 2024], apresentando novos datasets e arquiteturas de redes neurais profundas para validar a acurácia e a generalização do método LV-CIT em diferentes situações.

2. Trabalhos Relacionados

A avaliação de sistemas baseados em Redes Neurais Profundas (DNNs) é uma área de pesquisa que tem atraído atenção, com diversos métodos de testes sendo citados na literatura. Entre eles, o Teste de Interação Combinatória (CIT), que surge como uma metodologia inovadora para avaliar como as diferentes variáveis de entrada de um modelo interagem e afetam seu desempenho. O CIT é relevante em cenários de alta dimensionalidade, onde a identificação de impacto das combinações de entrada na saída é importante. Essa abordagem não apenas revela relações e dependências existentes entre as variáveis, mas também oferece insights valiosos sobre a estrutura dos dados, o que contribui para a melhoria do desempenho e da confiabilidade do modelo. [Prince 2023]

A aplicação do CIT em testes de DNNs é ampla, abrangendo tanto abordagens white-box quanto black-box. Em testes white-box, o CIT tem sido utilizado na criação de conjuntos de dados que visam capturar interações complexas de características de entrada, como, por exemplo, na exploração de combinações de operações de transformação de imagem para diversificar imagens de treinamento. No entanto, a crescente preocupação com a segurança e a privacidade dos dados, que muitas vezes impossibilita o acesso a detalhes internos da arquitetura da rede ou seus dados de treinamento, então o foco da pesquisa tem mudado para metodologias de teste black-box [Wang et al. 2024], onde o CIT também se mostra promissor.

Um desafio encontrado nos testes de classificadores de imagens é a avaliação de modelos multi-rótulos, nos quais a detecção e o correto tratamento de correlações entre múltiplos objetos presentes em uma imagem são importantes. Para abordar essa complexidade e aprimorar a avaliação do desempenho desses sistemas, métodos específicos de teste têm sido desenvolvidos. [Wang et al. 2024]

Nesse contexto, o método LV-CIT [Wang et al. 2024] foi introduzido com o objetivo de aprimorar a avaliação do desempenho de classificadores de imagens multi-rótulos. O LV-CIT se diferencia dos demais por modelar cada rótulo como um parâmetro de entrada binário, o que permite uma análise das combinações de valores de rótulo por meio da geração de um array de cobertura de valor de rótulo t-way. Esta estratégia confere ao LV-CIT uma uma capacidade robusta de detecção de erros, incluindo erros de excesso, e de incompatibilidade, como destacado por seus autores[Wang et al. 2024].

Apesar da eficácia apresentada pelo LV-CIT na literatura, a validação de sua robustez em outros contextos é importante. Assim o trabalho desenvolvido nesse artigo é explorar a aplicação do LV-CIT em um ambiente black-box utilizando novos datasets e diferentes arquiteturas de redes neurais profundas. O objetivo é replicar e estender os experimentos originais de [Wang et al. 2024], comparando os resultados obtidos e buscando validar a acurácia e a aplicabilidade do método em uma quantidade maior de cenários,

identificando modelos que apresentem o melhor desempenho sob essas novas condições de teste.

link do GitHub: https://github.com/Infobyte-hub/deep-learning-LV-CIT

3. Metodologia

Este trabalho tenta replicar, de forma prática, o método descrito por [Wang et al. 2024] para testes combinatórios em classificadores de imagens multi-rótulo, explorando a geração automatizada de combinações por covering arrays (LV-CIT) e redes neurais profundas em computação em nuvem. Para isso, foi utilizado o script compositer.py, adaptado do repositório original, a fim de gerar imagens artificiais a partir dos conjuntos de dados VOC [Everingham et al.] e COCO [Lin et al. 2014].

O software utilizado para a implementação do algoritmo inicial da replica do experimento foi o Visual Studio versão 2017/2022. A tentativa de implementação foi realizada em linguagem python/conda, porém pela dificuldade encontrada na instalação das dependências necessárias não foi possível prosseguir. Posteriormente, todo o processamento ocorreu no Google Colab, com integração ao Google Drive para armazenamento persistente dos dados e checkpoints.A razão pela escolha destes softwares está baseada na facilidade e praticidade de análise dos resultados.

A máquina utilizada na implementação do algoritmo é equipada com um processador **Intel(R) Core(TM) i5-10300H CPU @ 2.50GHz**. Este processador conta com **4 núcleos físicos** cada núcleo com duas threads totalizando 8 threads, operando a uma frequência base de 2.50 GHz, com capacidade de atingir frequências mais elevadas via Turbo Boost. A hierarquia de cache inclui 32 KB de cache L1 por núcleo (para dados e instruções), 256 KB de cache L2 por núcleo, e **8 MB de cache L3 compartilhada**, otimizando o acesso a dados e instruções. O sistema possui **8,00 GB de memória RAM** (7,87 GB utilizáveis) e opera em uma **arquitetura de 64 bits**.

A arquitetura de rede escolhida foi a ResNet18 [Prince 2023], treinada do zero com pesos pré-treinados do ImageNet, ajustando a camada final para classificação binária simulada, uma vez que os rótulos originais foram simplificados para demonstração do pipeline. Foram processadas aproximadamente 50.000 imagens combinadas dos conjuntos VOC2007 e COCO2014, organizadas automaticamente e rotuladas de forma simulada para abranger as 20 classes previstas no VOC.

A sequência experimental implementada contou com a geração das combinações, a montagem automática das imagens compostas, o treinamento supervisionado da rede neural, a validação dos resultados e o salvamento periódico dos checkpoints. Toda a execução apresentou logs, logs no TensorBoard e registros em arquivos auxiliares para monitoramento, permitindo o rastreio e a reprodutibilidade do fluxo.

Apesar do sucesso parcial no treino, validação e teste, o experimento enfrentou limitações práticas, como restrições de uso de GPU (T4 limitada no Colab), lentidão de processamento em CPU, dependências quebradas herdadas do repositório original e eventuais divergências de ambiente, conforme já relatado pelos autores no repositório experimental da literatura base.

4. Resultados

Durante os testes, o objetivo foi validar a robustez do pipeline combinatório, garantindo a qualidade dos checkpoints gerados, a estabilidade do treinamento e a consistência dos logs de validação. Foram geradas aproximadamente 30.600 imagens compostas via LV-CIT. A rede ResNet18 alcançou uma acurácia média de 100%, com loss mínimo próximo de zero, ao longo de 5 epochs completas.

Em algumas execuções, foi observado que a acurácia se manteve em 100% já nas primeiras iterações, apontando a baixa variabilidade do conjunto artificial gerado, assim como na literatura original. Todos os resultados intermediários foram salvos automaticamente no arquivo best_checkpoint.pth, dentro do Google Drive, garantindo que a sessão pudesse ser retomada em caso de interrupção. As métricas de treino e validação foram acompanhadas em tempo real pelo TensorBoard, complementadas por registros adicionais em planilhas CSV.

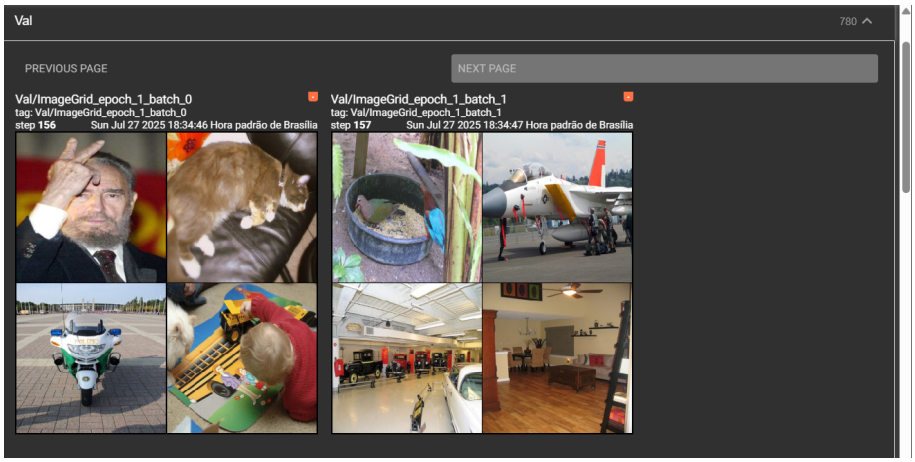

Figure 1. Imagens usadas na validação do modelo

A Tabela 1 ilustra, de forma resumida, os valores médios de acurácia e loss por epoch, além de observações relevantes sobre cada etapa do treino.

Table 1. Resumo das métricas observadas

Epoch	Acurácia (%)	Loss	Observações
1	99.70	0.0078	Checkpoint salvo
2	100.00	0.0000	Checkpoint salvo
3	100.00	0.0000	Checkpoint salvo
4	100.00	0.0000	Checkpoint salvo
5	100.00	0.0000	Checkpoint salvo

A Tabela 2 apresenta a configuração geral utilizada para a execução dos experimentos, apontando os parâmetros essenciais como o modelo ResNet18, os datasets combinados (VOC2007 e COCO2014) e o ambiente de execução, que variou entre GPU T4 e CPU, dependendo da disponibilidade no Google Colab.

Por fim, a Tabela 3 apresenta as principais limitações enfrentadas durante a reprodução, como a restrição de tempo de GPU, limitações de dependências herdadas

Table 2. Configuração geral do experimento

Parâmetro	Valor
Modelo	ResNet18
Dataset principal	VOC2007 + COCO2014
Tamanho total de imagens	Aproximadamente 50.000
Batch Size	64
Número de Epochs	5
Dispositivo	GPU T4 (quando disponível) ou CPU
Ferramenta de Monitoramento	TensorBoard + Checkpoints

do repositório original e a necessidade de uso parcial dos conjuntos de dados devido a restrições de armazenamento em nuvem.

Table 3. Principais limitações enfrentadas

Aspecto	Descrição
Tempo de Execução	Treino limitado por restrição de GPU e CPU lenta
Dependências	Erros eventuais em libs originais do LV-CIT
Cobertura de Dados	Uso parcial de VOC e COCO por espaço no Drive
Monitoramento	Visualização limitada no TensorBoard

5. Conclusões e Trabalhos Futuros

A execução deste experimento, permitiu reproduzir parcialmente as principais etapas do método LV-CIT, validando o conceito de geração automatizada de combinações de rótulos para teste de classificadores multi-rótulo. Apesar de restrições técnicas e de infraestrutura, os resultados obtidos indicam coerência com os valores apresentados por [Wang et al. 2024] em cenários de demonstração controlados. Ao longo da implementação, foram adotados scripts ajustados, organizado no Google Colab, armazenamento no Google Drive e monitoramento via TensorBoard.

Durante o processo, foram identificados limitações relevantes, como dependência de GPU T4 no Colab, compatibilidade de dependências e limitações de tempo de sessão, que dificultaram a ampliação do projeto para múltiplos datasets ou arquiteturas mais complexas. Ainda assim, o pipeline se manteve estável, com geração de imagens compostas a partir dos datasets VOC e COCO, checkpoints salvos para retomada e documentação clara em blocos numerados. Essa estrutura serve como base para extensões futuras, visando maior robustez e reprodutibilidade.

Para trabalhos futuros, a próxima abordagem está em testar redes mais potentes, como ResNet50 ou EfficientNet, automatizar a geração de covering arrays, replicar o experimento em outros conjuntos de dados e explorar a criação de uma API funcional para aplicação de testes combinatórios de classificadores multi-rótulo. Assim, o projeto cumpre seu objetivo acadêmico de demonstrar o funcionamento prático do LV-CIT e abre caminho para novas aplicações e pesquisas futuras.

References

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer.

Prince, S. J. (2023). *Understanding deep learning*. MIT press.

Wang, P., Hu, S., Wu, H., Niu, X., Nie, C., and Chen, L. (2024). A combinatorial interaction testing method for multi-label image classifier. In *2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE)*, pages 463–474. IEEE.