Aprimoramento de Modelos de Classificação com Dados Enriquecidos via Web Scraping: Um Estudo de Caso da Competição Dog Breed Identification

Marcos V. M. Faria[1], Ludmila Dias[1], Eduardo O. P. Ferreira[1], Thiago M. Paixão.[1], Francisco A. Boldt[1]

[1]Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo (IFES) - Serra, ES, Brasil

{marcos.faria,ludmila.dias}@estudante.ifes.edu.br

duduoliveirae11@gmail.com

{thiago.paixao,franciscoa}@ifes.edu.br

Abstract. *This article presents a study on the use of Web Scraping for automated data extraction from the web, aimed at enhancing classification models through the enrichment of the training data base. In our experiments, we utilized two databases: one from the Kaggle "Dog Breed Identification" competition, which served as a case study, and another resulting from the merger of this with a database extracted via scraping. In the extraction process, we employed the Puppeteer library and other auxiliary tools at specific stages of the process. The classification model adopted was Xception. The results were compared based on the metrics of Accuracy, Recall, Precision, and F1 Score. We conclude that the addition of data via web scraping can improve classification performance, provided that the data is properly cleaned.*

Resumo. *Este artigo apresenta um estudo sobre o uso de Web Scraping para a extração automatizada de dados da web, visando aprimorar modelos de classificação por meio do enriquecimento da base de dados de treinamento. Nos experimentos, utilizamos duas bases de dados: uma proveniente da competição "Dog Breed Identification" do Kaggle, que serviu de estudo de caso, e uma resultante da fusão desta com a base de dados extraída via scraping. No processo de extração, empregamos a biblioteca Puppeteer e outras ferramentas auxiliares em determinadas etapas do processo. O modelo de classificação adotado foi o Xception. Os resultados das bases de dados foram comparados através das métricas de Acurácia, Recall, Precisão e F1 Score. Concluímos que a adição de dados via web scraping pode melhorar o desempenho de classificação, desde que uma limpeza dos dados seja aplicada.*

1. Introdução

A Inteligência Artificial (IA) é um campo interdisciplinar da Ciência da Computação que abrange várias subáreas dedicadas a executar tarefas por meio de algoritmos que simulam a inteligência humana [Russel and Norving 2022]. Uma dessas subáreas é o Aprendizado de Máquina (ML, do inglês *Machine Learning*), cujo objetivo é realizar tarefas por meio

de modelos treinados a partir de dados, isto é, sem a necessidade de instruções explícitas. Dentre as abordagens principais, temos o aprendizado supervisionado, onde algoritmos são treinados com dados rotulados. Isso significa que cada entrada do algoritmo é associada a uma saída conhecida, permitindo que o modelo aprenda a mapear entradas semelhantes às saídas correspondentes. Exemplos dessa abordagem incluem a classificação, que prevê categorias para novas amostras, e a regressão, que prevê valores contínuos com base em variáveis independentes.

Somente com a viabilidade dos modelos de Aprendizado Profundo (DL, do inglês *Deep Learning*) é que diversas tarefas, especialmente aquelas de visão computacional, tornaram-se viáveis [Chollet 2021]. Os modelos tradicionais de ML não apresentavam desempenho satisfatório nesse tipo de tarefa devido à sua limitada capacidade de capturar a complexidade e a variabilidade dos dados visuais [Voulodimos et al. 2018], além da dificuldade em extrair características significativas das imagens de forma eficaz. Por outro lado, DL utiliza técnicas baseadas em redes neurais artificiais com múltiplas camadas. Para alcançar um desempenho satisfatório, esses modelos requerem uma quantidade substancial de dados para treinamento devido ao elevado número de parâmetros a serem ajustados [Munappy et al. 2019].

Com o passar do tempo, foram realizados diversos esforços para criar bases de imagens em larga escala para treinamento de modelos de ML em diferentes contextos. O exemplo notável nesse contexto é a base ImageNet [Deng et al. 2009], criada por Fei-Fei Li, que percebeu a necessidade de bases mais amplas e de melhor qualidade para aprimorar a capacidade dos algoritmos de interpretar imagens. Em 2007, esse esforço resultou em uma base de dados com aproximadamente 15 milhões de imagens, abrangendo mais de 22.000 categorias distintas. Outros iniciativas incluem o MNIST, uma base para reconhecimento de dígitos manuscritos [Deng 2012], e o COCO (Common Objects in Context), que contém mais de 200.000 imagens rotuladas em diversas categorias de objetos comuns [Lin et al. 2014].

Embora a ImageNet tenha tido um impacto muito positivo no campo de visão computacional, sua construção foi possível graças a um esforço coletivo, com contribuições de pessoas de diversos países para a rotulagem manual das imagens. Apesar da alta eficácia desse método, ele também implicou em um considerável custo operacional. Plataformas como o UCI Machine Learning Repository [University of California 2024], Kaggle Datasets [Kaggle 2024] e Papers With Code [with Code 2024] oferecem conjuntos de dados de diversas categorias. No entanto, nem sempre é viável encontrar bases de dados abrangentes sobre qualquer tópico ou na quantidade desejada.

A proposta deste trabalho é aprimorar um modelo de classificação utilizando uma base de dados gerada por um processo de extração via Web Scraping, que será denominada base “Scraper”. Como estudo de caso, escolhemos a competição Dog Breed Identification do Kaggle para avaliar a eficácia dessa base extraída. Treinamos o modelo Xception [Chollet 2017] em dois cenários distintos: no primeiro, utilizamos a base original da competição, chamada base “Kaggle”; no segundo, empregamos uma fusão da base “Kaggle” com a base “Scraper”, resultando na base “Mesclada”. Além disso, para aprimorar a base “Mesclada”, implementamos e testamos duas abordagens distintas de limpeza: uma manual e outra automática, utilizando o YOLOv8 [Jocher et al. 2022].

2. Referencial Teórico

2.1. Rotulagem de imagens

A construção de uma base de dados envolve várias etapas importantes, entre as quais se destacam a coleta de dados, a rotulagem, o pós-processamento e a verificação de qualidade [Sager et al. 2021]. Em específico, a rotulagem de imagens envolve a criação ou atribuição de uma descrição que conecta as características visuais da imagem ao texto resultante. Atualmente, existem diferentes maneiras para realizar a rotulagem de imagens, variando desde métodos manuais, onde humanos anotam cada imagem, geralmente utilizando uma interface visual interativa [Torralba et al. 2010], ou então, métodos automatizados, que tem sido o foco de pesquisas nessa área. Cada abordagem possui características específicas: a rotulagem manual, em geral, garante maior confiabilidade, resultando em uma base de dados de maior qualidade, como exemplo o ImageNet citado anteriormente. Por outro lado, existem abordagens automáticas e semi-automáticas que utilizam diferentes métodos para atribuir rótulos a imagens com mínima intervenção humana [Zhang et al. 2012].

2.2. Trabalhos correlatos

No trabalho [Srinivasan et al. 2021], foi construído o WIT (Wikipedia-based Image Text Dataset) utilizando dados extraídos de forma automatizada da Wikipedia para treinamento de modelos de ML que operam com dados visuais e textuais. A base (multimodal e multilíngue) conta com mais de 37 milhões de exemplos de imagem-texto e 11,5 milhões de imagens únicas em 108 idiomas. O WIT destaca-se por seu grande volume e cobertura multilíngue, visando aprimorar a capacidade dos modelos em entender e processar informações de forma cruzada entre diferentes línguas e contextos culturais. O objetivo do nosso trabalho, de maneira semelhante, é extrair dados de forma automatizada da Wikipedia e construir uma base para aprimorar um modelo de classificação. Simplificadamente, utilizaremos apenas imagens de cachorros, tendo como estudo de caso uma página específica.

No artigo [Sirisuriya 2023], é analisada a importância do web scraping como uma fonte de dados para algoritmos de ML. O autor destaca como o web scraping automatiza o processo de extração de dados de sites, utilizando softwares conhecidos como web scrapers, que carregam e extraem dados automaticamente com base nas necessidades do usuário. São descritas diferentes técnicas de web scraping, incluindo a extração manual e automatizada de dados, além do uso de ferramentas como BeautifulSoup, Scrapy e Selenium. O artigo também aborda aplicações práticas do web scraping em diversas áreas, como análise de sentimentos, classificação de imagens, processamento de linguagem natural e sistemas de recomendação, enfatizando sua capacidade de enriquecer conjuntos de dados existentes, melhorar a precisão dos modelos e realizar análises competitivas. A principal contribuição deste estudo é fornecer uma base teórica sólida para a utilização do web scraping em projetos de ML, evidenciando sua importância na obtenção de dados diversificados para o desenvolvimento de modelos mais eficientes. Além disso, o estudo destaca os desafios relacionados à complexidade e à lentidão do processo, que exigem conhecimentos especializados e atenção às implicações legais e éticas.

O trabalho [Correia et al. 2021], teve como objetivo reconhecer gestos de mãos utilizando uma base de dados adquirida por sensores inerciais. O estudo foi dividido

em duas partes: a primeira, utilizando o algoritmo KMeans para segmentação dos dados de gestos; e a segunda, empregando o algoritmo de Floresta Aleatória para classificação dos gestos segmentados, com e sem extração de características. Os experimentos demonstraram que a melhor precisão foi obtida com a extração de características, atingindo uma acurácia de 83%. Concluiu-se com esse trabalho que é possível combinar técnicas de aprendizado de máquina para identificar, segmentar e classificar gestos de mãos em uma base de dados gerada por sensores inerciais. Este estudo é relevante para nosso trabalho, pois evidencia que o uso de dados específicos e técnicas de aprimoramento, como a extração de características, melhora significativamente a precisão do modelo de classificação. De forma semelhante, nosso trabalho propõe o enriquecimento de bases de dados via web scraping para aprimorar modelos de classificação, permitindo que eles identifiquem novos casos com maior eficácia.

3. Materiais e métodos

A metodologia para o aprimoramento do modelo de classificação é composta por três etapas principais: extração de dados, pós-processamento e o treino e teste do classificador. A extração de dados foi implementada utilizando a técnica de Web Scraping, analisando a estrutura HTML da página "List of dog breeds" [Wikipedia 2024] e extraindo as imagens com a biblioteca Puppeteer [Puppeteer 2024]. Testes iniciais mostraram que a fusão da base "Scraper" com a base de dados "Kaggle" não resultou em um desempenho de classificação satisfatório após o treinamento. Para melhorar a qualidade da base gerada, na etapa de pós-processamento foram empregadas duas abordagens de limpeza: uma manual e outra automatizada com o modelo de detecção YOLOv8. Além disso, as imagens foram recortadas com o YOLOv8 para extrair apenas a região de interesse, reduzindo o ruído causado por outros elementos. Na etapa de treino e teste, utilizamos o modelo de classificação Xception [Chollet 2017] para treinar: a base "Kaggle", a base "Mesclada" sem pós-processamento, e a base "Mesclada" com a limpeza manual e automatizada. Para validação, criamos uma base de teste a partir da base "Kaggle", utilizando quatro amostras de cada classe. Com isso, comparamos os resultados das quatro abordagens e avaliamos a acurácia, recall, precisão e F1 Score para determinar a melhor abordagem.

3.1. Extração de dados

Web Scraping é o processo de extrair dados de páginas web, normalmente em formato HTML, e estruturar esses dados para obter informações de interesse [R et al. 2023]. A etapa crucial deste projeto foi o desenvolvimento de um processo automatizado para a extração das imagens rotuladas da página "List of Dogs Breeds" do Wikipedia [Wikipedia 2024]. A primeira etapa envolveu o *parsing* da página, uma técnica que consiste na análise da estrutura HTML do site para identificar regiões que contêm informações de interesse. Ao localizar o conteúdo alvo, utilizamos o *Puppeteer* [Puppeteer 2024], uma biblioteca Node.js que oferece uma API de alto nível para interação com o navegador. Após identificar o *CSS selector* que se repetia para cada elemento, consistindo em uma tag HTML 1 com a referência para a página alvo, as páginas válidas foram acessadas até que não houvesse mais páginas restantes. Em cada uma dessas páginas, todos os links de imagens passaram por uma filtragem de tamanho, sendo utilizadas apenas as imagens com dimensões superiores a 70x70 pixels. Algumas imagens no Wikipedia Dog Breeds seguiam uma estrutura que disponibilizava diferentes dimensões para a mesma imagem. Para essas, recuperávamos o link apenas da imagem de

maior tamanho. Todos os links eram armazenados em um arquivo .JSON para, ao final do processo, serem baixados utilizando um script em Python. O processo descrito pode ser visualizado através de um fluxograma na Figura 2.

```
<a href="/wiki/Affenpinscher" title="Affenpinscher">Affenpinscher</a>
```

Figura 1. Exemplo de um elemento que contém o link para página com uma raça de cachorro.

Dessa forma, utilizando a estrutura da própria página, conseguimos atribuir rótulos às imagens e salvá-las de forma estruturada. Todos os códigos desenvolvidos e utilizados neste projeto estão disponíveis no repositório do GitHub[1].

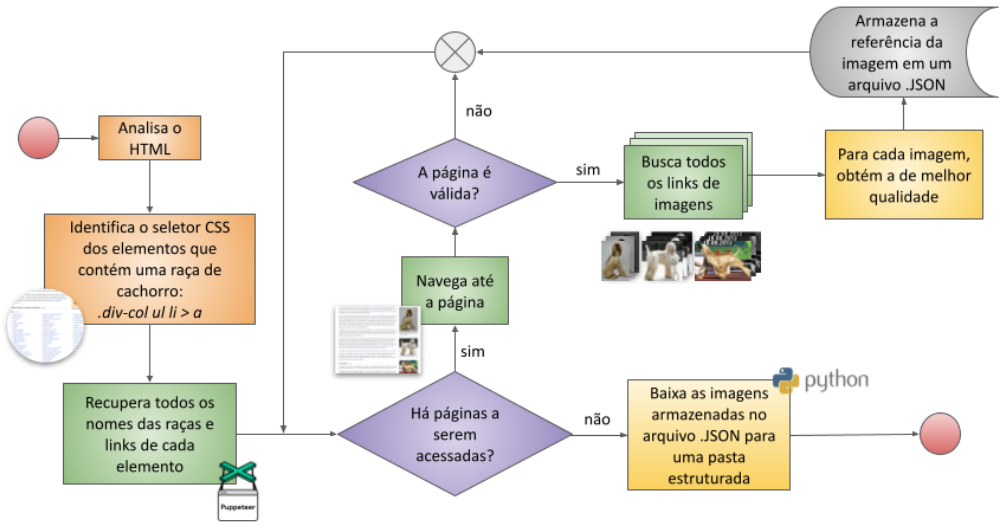

Figura 2. Fluxograma do processo de extração das imagens rotuladas.

3.2. Pós-processamento

Finalizada a fase de extração de dados, realizamos testes iniciais com a base "Mesclada", a fim de compreender sua qualidade. Observamos que, ao incorporar a base"Scraper", houve uma redução na performance de classificação. Especificamente, a acurácia da base "Kaggle" era de 0.585, enquanto a da base "Mesclada" foi de 0.575. Supomos que isso ocorreu devido à adição de ruído na base, uma vez que nem todas as imagens extraídas necessariamente continham um cachorro e/ou estavam com uma qualidade adequada. Dessa forma, entendemos que seria necessário adicionar uma etapa de limpeza ao processo. Com o objetivo de entender a eficácia de cada método, a limpeza foi realizada de duas formas distintas: manualmente e de forma automática. Além disso, aplicamos também uma etapa de extração da região de interesse da imagem através do croping do objeto detectado.

[1]https://github.com/fboldt/scraper/tree/main/WebScraping

3.2.1. Limpeza manual

O objetivo principal dessa limpeza manual era remover o ruído identificado após o enriquecimento dos dados via scraping. Durante o processo de limpeza, foram retiradas as imagens que não continham cachorros ou que apresentavam uma qualidade ruim, como pinturas e desenhos que não deixavam evidente a presença de um cachorro. Esse processo envolveu a inspeção visual de cada imagem e a remoção das que não atendiam aos critérios estabelecidos. Ao final, foram eliminadas 723 imagens inadequadas que poderiam comprometer o desempenho do modelo de classificação, alguns exemplos podem ser vistos na Figura 3. Sendo assim, a base "Mesclada" após o processo manual de limpeza ficou com 10368 amostras.

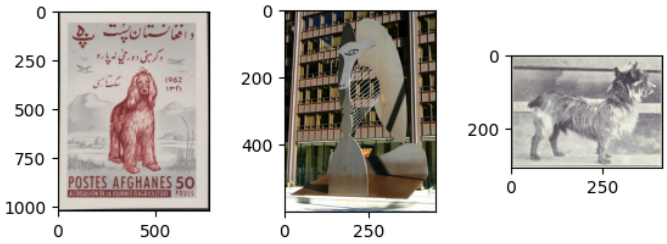

Figura 3. Imagens removidas no processo de limpeza.

3.2.2. Limpeza automática

Para a limpeza automatizada, foi implementado um script em Python utilizando a biblioteca Ultralytics [2], que permite a utilização dos modelos YOLO. O script carrega cada imagem usando OpenCV, faz o processamento com o modelo YOLOv8x para detectar objetos e extrai as caixas delimitadoras, classes e informações da detecção. Em seguida, verifica se a classe detectada é igual a 16, correspondente ao rótulo "dog". Se um cachorro for detectado, a imagem é salva na pasta de saída. Imagens sem a presença de cachorros foram removidas. Após a limpeza automática, a quantidade de amostras foi reduzida de 11091 para 10295, mostrando uma remoção significativa de dados irrelevantes. A limpeza da base de dados foi realizada por uma aluna da graduação, que contribuiu voluntariamente para o projeto de pesquisa, com o objetivo de documentar o processo para seu trabalho de conclusão de curso.

3.2.3. Extração da região de interesse

O modelo YOLOv8 foi utilizado para a extração da região de interesse (bounding boxes) em torno dos objetos relevantes na cena, especificamente aqueles identificados como cachorros. Com base nas coordenadas fornecidas pelo modelo, cada região é extraída e armazenada separadamente utilizando a biblioteca OpenCV [3]. Na Figura 4, é apresentado um exemplo do processo de recorte da região de interesse da imagem utilizando o YOLOv8 em conjunto com o OpenCV. Nesse processo, o modelo YOLOv8 recebe a imagem

[2]https://docs.ultralytics.com/
[3]https://opencv.org/

à esquerda como entrada e retorna as coordenadas dos quatro pontos que delimitam a área onde o objeto de interesse, neste caso, um cachorro, foi identificado. Em seguida, o OpenCV é utilizado para recortar a imagem nessa região, removendo assim elementos que poderiam comprometer o desempenho da classificação na etapa de treinamento.

A justificativa para recortar apenas o elemento central da imagem, como o cachorro, é que a remoção de informações irrelevantes ou potencialmente ruidosas pode aumentar a precisão do modelo de classificação. Ao focar exclusivamente no objeto de interesse, o modelo aprende de forma mais eficiente as características essenciais para a correta identificação, evitando a influência de elementos distratores no fundo da imagem [Chen et al. 2016].

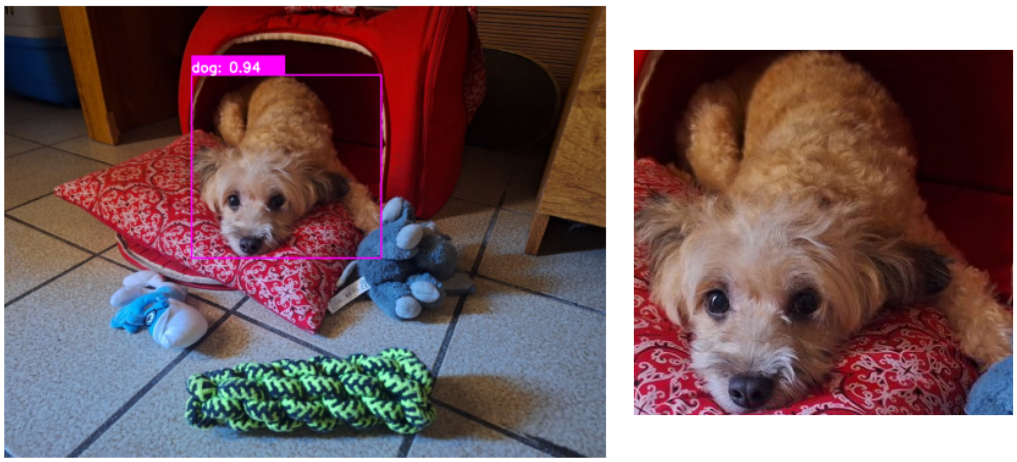

Figura 4. Exemplo de cropping utilizando YOLOv8 em conjunto com o OpenCV.

3.3. Treino e teste

3.3.1. Bases de dados

A base "Kaggle" é um subconjunto do ImageNet, contendo 10.222 imagens divididas entre 120 raças de cachorros. Como a plataforma Kaggle não forneceu uma base de testes rotulada, separamos quatro amostras de cada classe para criar a base de testes, totalizando 9.742 amostras de treino e 480 de teste.

A extração de dados por meio de *scraping* gerou a base "Scraper"com 2.470 imagens, distribuídas em 555 classes distintas. Para fusão com a base da "Kaggle", no entanto, selecionamos apenas as classes (raças) que eram comuns entre elas. A fusão das bases manteve as 120 raças da base "Kaggle", mas agora com um maior número de amostras: 9.742 imagens da base de treino, acrescidas de 1.349 imagens aproveitadas da base construída via scraping.

Na imagem disponível no repositório [4], vemos como a fusão enriqueceu a base "Kaggle". No eixo vertical, está representada a quantidade de imagens para cada classe, enquanto no eixo horizontal são exibidas as raças presentes na base de dados. A

[4]https://github.com/fboldt/scraper/blob/main/WebScraping/artigo/imagens/kaggle-scraper-dataset-horizontal-bars.png

distribuição da base “Kaggle” é representada pelas barras em azul escuro, enquanto as da base “Mesclada” é representada pelas barras em azul claro.

3.3.2. Experimentos

Para os experimentos, estabelecemos uma quantidade máxima de 50 épocas de treinamento e configuramos a taxa de aprendizado inicial em 0.001, ajustando-a dinamicamente por meio de um callback que a reduzia em 0.05 sempre que a acurácia de validação não melhorava após 2 épocas consecutivas. O modelo foi treinado e testado em quatro diferentes bases de dados: “Kaggle” (sem limpeza), “Mesclada” sem limpeza, “Mesclada” com limpeza automática, e “Mesclada” com limpeza manual, permitindo a avaliação do impacto da qualidade dos dados nos resultados.

O modelo Xception [Chollet 2017] foi escolhido com base em sua eficácia na classificação de imagens, como demonstrado no estudo de [Valarmathi et al. 2023]. Nesse estudo, que comparou diferentes algoritmos de deep learning para a identificação de raças de cães, o Xception alcançou uma acurácia de validação de 91,9%, destacando-se como uma opção robusta para esse tipo de tarefa. O modelo foi inicialmente treinado com pesos pré-definidos do ImageNet, facilitando o aproveitamento de características visuais gerais antes do treinamento específico (fine tuning). Essa técnica é conhecida como Transferência de Aprendizado (do inglês *Transfer Learning*) e pode acelerar significativamente o processo de treinamento e aumentar a precisão do modelo em reconhecer novas imagens fora do conjunto de treinamento. Para otimizar a performance e a generalização do modelo, foram incorporadas técnicas de aumento de dados, como rotação, espelhamento e zoom aleatórios, que ajudam a diversificar as amostras de treinamento e mitigar o *overfitting*.

4. Resultados

A Tabela 1 apresenta os resultados obtidos nos testes para cada base. Para avaliação, utilizamos as métricas de Acurácia, Recall, Precisão e F1 Score. Inicialmente, podemos observar que a adição de dados por meio de scraping, sem um processo de limpeza, aparentemente introduziu ruído no dataset, ocasionando uma redução moderada tanto na acurácia quanto no recall. Esse resultado era esperado, visto que, mesmo com uma análise prévia da página antes da extração não é possível assegurar a qualidade das imagens obtidas.

Por outro lado, os dados que passaram por um processo de limpeza, seja através do modelo automático YOLOv8 ou manualmente, mostraram melhorias consideráveis. Comparado à base do Kaggle, a limpeza com YOLOv8 aumentou a acurácia de 0.585 para 0.604, indicando uma melhora na generalização, embora o recall só tenha aumentado 0.9 ponto percentual. A limpeza manual resultou em melhorias significativas em todas as métricas em comparação com a base de dados Kaggle. A acurácia aumentou em 4,2 pontos percentuais, passando de 0,585 para 0,627. O recall teve um incremento de 9,9 pontos percentuais, subindo de 0,714 para 0,813, indicando uma maior capacidade do modelo em identificar corretamente os casos positivos. A precisão também melhorou em 9,4 pontos percentuais, passando de 0,719 para 0,813, demonstrando que o modelo se tornou mais eficaz em evitar falsos positivos. Por fim, o F1 Score, que é a média

harmônica entre recall e precisão, aumentou em 9,7 pontos percentuais, de 0,716 para 0,813, destacando o desempenho geral alcançado com a limpeza manual dos dados. Esses resultados reforçam a importância do tratamento cuidadoso dos dados para melhorar a performance do modelo.

Dataset	Acurácia	Recall	Precisão	F1 Score
Kaggle	0.585	0.714	0.719	0.716
Mesclada sem limpeza	0.575	0.710	0.711	0.710
Mesclada com limpeza automática	0.604	0.723	0.723	0.723
Mesclada com limpeza manual	0.627	0.813	0.813	0.813

Tabela 1. Resultados obtidos

5. Conclusão

Este trabalho propôs aprimorar modelos de classificação utilizando dados enriquecidos via Web Scraping, com um estudo de caso na competição "Dog Breed Identification" do Kaggle. Foram realizados experimentos com o objetivo de validar se a fusão entre as bases "Kaggle" e "Scraper" trariam uma melhora na performance de classificação do modelo Xception. Adicionalmente, foram testadas variações dessa base dados ("Mesclada"), submetidas a dois tipos de limpeza de dados: uma limpeza manual e outra automática, realizada através da YOLOv8. Baseado nos resultados apresentados, concluímos que a adição de dados via web scraping pode resultar em melhor desempenho de classificação desde que seja aplicada a limpeza de dados. Para trabalhos futuros, planejamos aplicar o mesmo processo em diferentes contextos, visando não apenas extrair imagens rotuladas de cenários mais complexos, onde é difícil encontrar dados, mas também para aprimorar modelos de classificação diferentes.

Referências

Chen, J., Bai, G., Liang, S., and Li, Z. (2016). Automatic image cropping : A computational complexity study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions.

Chollet, F. (2021). *Deep learning with Python*. Simon and Schuster.

Correia, C. H. G., Komati, K. S., and Boldt, F. d. A. (2021). Reconhecimento de gestos de mão em sequência a partir de sensores inerciais. *Journal of Health Informatics*, 12.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142.

Jocher, G., Chaurasia, A., Stoken, A., Borovec, J., NanoCode012, Kwon, Y., Michael, K., TaoXie, Fang, J., imyhxy, Lorna, Yifu), Wong, C., V, A., Montes, D., Wang, Z., Fati, C., Nadar, J., Laughing, UnglvKitDe, Sonck, V., tkianai, yxNONG, Skalski, P., Hogan, A., Nair, D., Strobel, M., and Jain, M. (2022). ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation.

Kaggle (2024). Kaggle datasets.

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.

Munappy, A., Bosch, J., Olsson, H. H., Arpteg, A., and Brinne, B. (2019). Data management challenges for deep learning. In *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 140–147.

Puppeteer (2024). Puppeteer.

R, R. R. N., S, N. R., and M., V. (2023). Web scrapping tools and techniques: A brief survey. In *2023 4th International Conference on Innovative Trends in Information Technology (ICITIIT)*, pages 1–4.

Russel, S. and Norving, P. (2022). *Inteligência Artificial - Uma Abordagem Moderna*. GEN LTC, 4th edition.

Sager, C., Janiesch, C., and Zschech, P. (2021). A survey of image labelling for computer vision applications. *Journal of Business Analytics*, 4(2):91–110.

Sirisuriya, S. D. S. (2023). Importance of web scraping as a data source for machine learning algorithms - review. In *2023 IEEE 17th International Conference on Industrial and Information Systems (ICIIS)*, pages 134–139.

Srinivasan, K., Raman, K., Chen, J., Bendersky, M., and Najork, M. (2021). Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21. ACM.

Torralba, A., Russell, B. C., and Yuen, J. (2010). Labelme: Online image annotation and applications. *Proceedings of the IEEE*, 98(8):1467–1484.

University of California, I. (2024). Uci machine learning repository.

Valarmathi, B., Gupta, N. S., Prakash, G., Reddy, R. H., Saravanan, S., and Shanmugasundaram, P. (2023). Hybrid deep learning algorithms for dog breed identification—a comparative analysis. *IEEE Access*, 11:77228–77239.

Voulodimos, A., Doulamis, N., Doulamis, A., and Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018.

Wikipedia (2024). List of dog breeds.

with Code, P. (2024). Papers with code.

Zhang, D., Islam, M. M., and Lu, G. (2012). A review on automatic image annotation techniques. *Pattern Recognition*, 45(1):346–362.