Análise do desenpenho de transfer learning em Vision Transformers e CNNs

Andrey Cortez Rufino (11819487), Arthur Queiroz Moura (13671532), Jorge Luiz Franco (13695091)

[a]*Universidade de São Paulo - Instituto de Ciências Matemáticas e da Computação,*

Abstract

O presente trabalho visa, com o auxílio da técnica de transfer learning, aumentar a acurácia de nosso algoritmo de detecção de animais (elucidado no trabalho anterior). Para isso, vamos continuar a experimentar com a arquitetura que teve a melhor performance no trabalho anterior (CNN), mas também vamos experimentar com outra arquitetura de rede neural, a Vision Transformers (ViT). Os códigos desenvolvidos nesse projeto estão disponíveis em https://drive.google.com/file/d/1BGAajG12HZ_7-RfacVXT6m4x9yl7q6jw/view?usp=sharing e os modelos aqui treinados treinados, EffNetB2 e ViT-B/16, podem ser testados usando, respectivamente, os links https://huggingface.co/spaces/jorgerix/animal_images e https://huggingface.co/spaces/jorgerix/animal_vit.

Keywords: Redes Neurais, Transfer learning, Convolutional neural networks, Vision transformers

1. Introdução

Dada uma imagem da face de um animal pertencente a uma das classes: galinha, águia, urso, panda, veado, elefante e macaco, como podemos, apenas com base na imagem, determinar a qual classe esse animal pertence?

Esse foi o problema proposto na entrega anterior. Nela, conseguimos determinar um algoritmo capaz de resolver o mesmo com 76% de acurácia. O objetivo do presente trabalho é de criar um algoritmo capaz de superar esse baseline.

O primeiro passo para melhorar a performance de nosso algoritmo é conhecer suas fraquezas. No relatório anterior, treinamos o algoritmo com um dataset pequeno (719 imagens) e, por isso, sua capacidade de extrair características dos animais nas mais diversas condições foi amplamente reduzido. Para resolver esse problema, aumentar o dataset seria uma solução. No entanto, ela é trabalhosa e sua execução vai muito além dos objetivos desse estudos e, por isso, resolvemos recorrer a outra técnica, o transfer learning.

Transfer learning é uma abordagem poderosa e eficiente na área de aprendizado de máquina que permite aproveitar o conhecimento adquirido em uma tarefa específica e aplicá-lo em um novo contexto ou domínio. Ao utilizar modelos pré-treinados em conjuntos de dados massivos, como redes neurais convolucionais treinadas em grandes conjuntos de imagens, é possível extrair características gerais e representações de alto nível que capturam informações relevantes. Essas representações podem, então, ser transferidas e ajustadas para resolver problemas diferentes ou em domínios específicos, o que exige menos dados de treinamento e acelera o processo de desenvolvimento de modelos. Além disso, o transfer learning facilita a reutilização de modelos existentes, o que economiza tempo e recursos computacionais. Essa técnica tem sido amplamente adotada em diversas aplicações, desde visão computacional e processamento de linguagem natural até medicina e finanças, impulsionando avanços significativos na inteligência artificial (1).

Uma das redes que planejamos utilizar e ajustar ao nosso projeto é a EfficientNetB2. Essa arquitetura de rede neural convolucional se destaca por sua eficiência e desempenho em uma ampla gama de tarefas de visão computacional. A EfficientNetB2 é baseada em um processo de escalonamento composto por diferentes níveis de largura, profundidade e resolução, que permitem um equilíbrio ideal entre eficiência computacional e capacidade de representação. Com sua estrutura eficiente, a rede oferece uma capacidade de aprendizado poderosa, permitindo extrair recursos relevantes e de alto nível a partir de imagens de entrada (2). Ao ajustar a EfficientNetB2 para o nosso projeto específico, poderemos aproveitar o conhecimento prévio adquirido em conjuntos de dados massivos e adaptá-lo para resolver nosso problema, o que resulta em um modelo mais rápido e eficaz.

Além da CNN, pretendemos usar outra arquitetura de rede neural, os ViTs, também conhecidos como Vision Transformers. Eles são uma abordagem inovadora para tarefas de visão computacional que aplicam a arquitetura transformer, originalmente desenvolvida para processamento de linguagem natural, ao domínio de imagens. Ao contrário das redes convolucionais tradicionais, os ViTs não dependem de operações convolucionais, mas sim de uma combinação de camadas de atenção e camadas totalmente conectadas. Isso permite que os ViTs capturem relacionamentos de longo alcance entre os diferentes elementos de uma imagem, transformando-as em sequências de tokens (Figura 1). Esses tokens são então processados pelo modelo transformer para extrair informações e realizar tarefas como classificação, segmentação e detecção de objetos (3). A abordagem dos Vision Transformers tem se mostrado promissora, alcançando resultados competitivos em diversos conjuntos de dados de referência e estendendo o sucesso da arquitetura transformer

ao campo da visão computacional.

Para o caso do ViT, pretendemos usar a rede ViT-B/16, proposta em (4), uma variante da arquitetura Vision Transformer (ViT) para tarefas de visão computacional. Ele utiliza transformadores para processar imagens divididas em patches de 16x16 pixels. Com várias camadas de transformadores, incluindo atenção multi-cabeça e camadas totalmente conectadas, o ViT-B/16 captura relacionamentos de longo alcance entre os patches e realiza a classificação das imagens. Ele tem se destacado em tarefas de classificação de imagens, alcançando resultados impressionantes em diversos conjuntos de dados.

Com o uso dessas técnicas, buscamos atingir melhorias significativas em relação aos algoritmos criados anteriormente.

Esse trabalho pode ser muito significativo para auxílio em estudos ecológicos: os classificadores de animais podem ser usados em pesquisas ecológicas para identificar diferentes espécies em imagens de câmeras de monitoramento ou em fotos tiradas em campo. Essas informações podem ser usadas para entender a dinâmica das populações de animais, estudar interações entre espécies e investigar padrões ecológicos em um determinado ambiente. Além disso, como mostrado adiante no nosso trabalho, o classificador obtido tem o poder de classificar animais com base em imagens de desenhos animados, isso pode ser muito benéfico para ferramentas educacionais de crianças na primeira idade, por exemplo, quando forem aprender sobre animais, eles podem colocar os seus desenhos animados preferidos e o algoritmo fala qual a espécie do animal da imagem.

Figure 1: Exemplificação de como o ViT transforma uma imagem em tokens

2. CNN vs ViT

As redes convolucionais (CNNs) têm sido amplamente utilizadas para tarefas de visão computacional, como classificação de imagens e detecção de objetos. Elas são especialmente eficazes em tarefas que requerem extrair características espaciais locais das imagens. As CNNs têm uma estrutura hierárquica de camadas convolucionais e camadas de pooling, que ajudam a capturar padrões visuais em diferentes escalas.

Por outro lado, as Transformers Visuais (ViTs) são baseadas nas Transformers, que originalmente foram projetadas para processar sequências de texto. As ViTs adaptam a arquitetura Transformer para lidar com imagens dividindo-as em patches e adicionando informações de posição. Essa abordagem permite que as ViTs capturem relações globais entre os patches, em vez de depender apenas de características locais.

Uma das principais diferenças entre as CNNs e as ViTs é a forma como elas tratam a informação espacial. As CNNs mantêm a estrutura espacial das imagens por meio de operações convolucionais e de pooling, enquanto as ViTs perdem essa informação ao dividir as imagens em patches. No entanto, as ViTs compensam essa perda de informação espacial capturando relações globais mais ricas entre os patches.

Outra diferença importante é o número de parâmetros. As CNNs geralmente têm um número significativo de parâmetros, especialmente em arquiteturas mais profundas, o que pode levar a problemas de sobreajuste em conjuntos de dados menores. Por outro lado, as ViTs têm menos parâmetros, tornando-as mais eficientes em termos de uso de memória e treinamento em conjuntos de dados menores.

Em termos de desempenho, as CNNs ainda são amplamente utilizadas e são muito eficazes em muitas tarefas de visão computacional. No entanto, as ViTs têm mostrado resultados promissores em várias tarefas, especialmente em problemas de visão de alto nível, como segmentação semântica e geração de imagens.

A figura 2 destaca a arquitetura da EfficientNet, uma das redes convolucionais mais populares e eficientes atualmente. Já a figura 3 mostra uma visão geral dos Vision Transformers, incialmente apresentados em (4).

Em suma, as CNNs são excelentes em capturar características espaciais locais, enquanto que as ViTs se destacam em capturar relações globais entre os patches de uma imagem, de modo que as duas sejam boas em diferentes contextos.

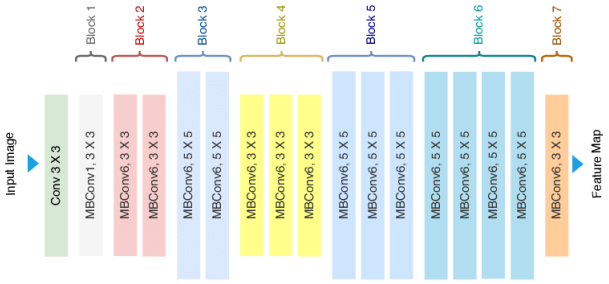

Figure 2: Arquitetura da EfficientNet

3. Métodos

3.1. O Dataset

Para essa entrega, usamos o mesmo dataset que foi usado anteriormente. Nele, temos 716 imagens de animais de sete classes diferentes (Figura 4) e, em todas as imagens, o animal está de face virada para o observador.

As sete classes estão igualmente balanceadas no dataset, assim como se pode ver na Figura 5

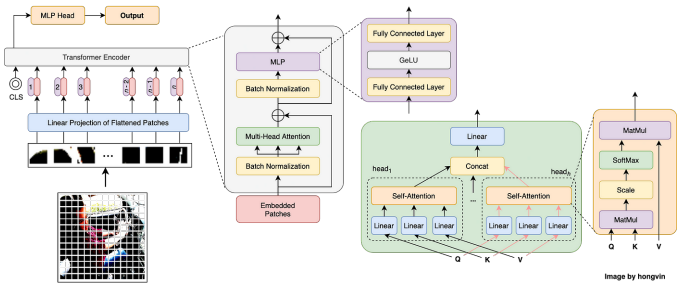

Figure 3: Arquitetura do ViT

Figure 4: Uma imagem de cada classe do dataset

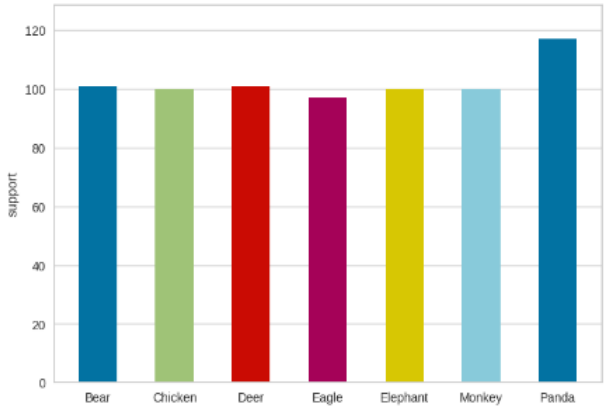

Figure 5: Plot de balanço de classes, comprovando que o dataset não tem um bias

Esses dados serão utilizados para fazer o fine-tuning, uma técnica utilizada no transfer learning em que um modelo pré-treinado é ajustado ou refinado em um conjunto de dados específico para uma tarefa específica.

O conjunto de dados é dividido em conjuntos de treinamento e de teste, geralmente seguindo a proporção de 70% para treinamento, 30% para teste. Em seguida, define-se o DataLoader, que carregará os dados do conjunto de treinamento e validação para fácil iteração durante o treinamento.

3.2. Treinando a rede EffNetB2

O treinamento do EfficientNet-B2 (EffNetB2) segue algumas etapas. Primeiramente, é necessário instalar a biblioteca torchvision e importar as classes relevantes, incluindo EfficientNet e nn, o módulo de redes neurais do PyTorch. Uma instância do modelo EffNetB2 é criada, permitindo ajustar o número de classes de saída, que são sete, e congelando os parâmetros do modelo pré-treinado, definindo `requires_grad = False`.

Após a definição do DataLoader, escolheu-se o otimizador de gradiente descendente estocástico, com taxa de aprendizado de 0.01. Além disso, para função de perda escolhida foi a de entropia cruzada.

Com todas as configurações em vigor, o modelo EffNetB2 pode ser treinado. Isso é feito através de um loop de treinamento que itera sobre um total de 10 épocas (epochs). Em cada época, os lotes de dados no DataLoader são percorridos. Os dados de entrada são passados pelo modelo EffNetB2 para obter as previsões, em seguida, a perda é calculada usando a função de perda e as previsões obtidas. Esse processo é repetido até que todas as épocas sejam concluídas, aperfeiçoando gradualmente o modelo EffNetB2 no reconhecimento de faces de animais.

Os resultados de acurácia e loss de cada época são registrados para uma análise posterior.

3.3. Treinando a rede ViT-B/16

O primeiro passo é a definição do modelo VIT-B/16. É necessário importar as bibliotecas necessárias e criar uma instância do modelo VIT-B/16 utilizando as implementações disponíveis no PyTorch.

Com o DataLoader configurado, o modelo é treinado iterando sobre um total de 10 épocas (epochs). Em cada época, os lotes de dados são percorridos e os cálculos são realizados. Os dados de entrada são passados pelo modelo VIT-B/16, as previsões são obtidas e a perda é calculada com base nas previsões e nos rótulos reais. O otimizador é então utilizado para atualizar os pesos do modelo com o objetivo de minimizar a perda. Esse processo é repetido até que todas as épocas sejam concluídas.

Os resultados de acurácia e loss de cada época são registrados para uma análise posterior.

4. Resultados

De antemão, trazemos os resultados do experimento anterior na Tabela 1.

	Acurácia	Loss
Rede Neural Linear	74%	1.67
Rede Neural Não Linear	71%	0.95
CNN	79%	2.00

Table 1: Resultados Anteriores

	Acurácia	Loss
EffNetB2	98.8%	0.07
ViT-B/16	100%	0

Table 2: Resultados do transfer learning

Então trazemos resultados das novas redes na tabela 2

Os gráficos com evolução dos parâmetros da rede em relação ao número de épocas de treinamento podem ser encontrados nas figuras 6 e 7, para a EffNetB2 e o ViT-B/16 respectivamente.

Temos ainda uma comparação entre o tempo de resposta de cada rede, seu tamanho e sua acurácia (Figura 8).

5. Conclusões

O primeiro ponto a ser observado é a melhoria avassaladora em relação aos métodos anteriores, isso se deve pelo fato de que, por estarmos falando de modelos pré treinados em datasets massivos, eles possuem capacidade de extração de caracteristicas visuais muito superiores aos modelos anteriores.

No caso da EffNetB2 obtivemos uma acurácia quase que perfeita, já no caso do ViT, obtivemos uma acurácia perfeita, ou seja, podemos dizer que esse problema de classificação, dentro das limitações propostas, foi resolvido com sucesso.

Ainda vale fazer as devidas comparações entre a EffNetB2 e o ViT, apesar de a EffNet ter tido um desempenho quase que insignificantemente inferior, ela é muito mais compacta e possui um tempo de resposta muito mais rápido que o do ViT. Assim, considerando o tipo de aplicação, seu uso pode ser mais

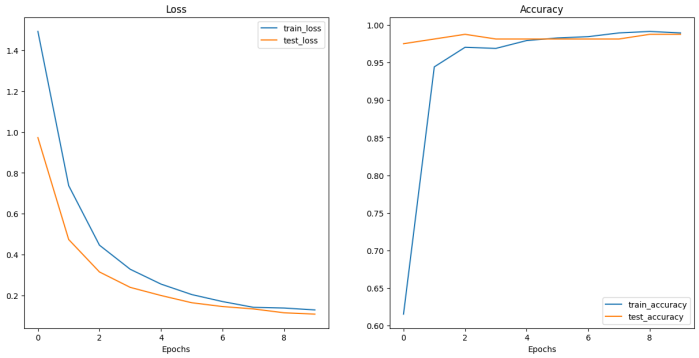

Figure 6: Resultados do transfer learning na rede EffNetV2

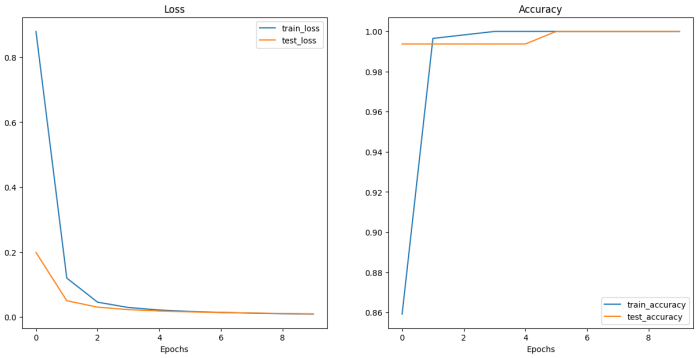

Figure 7: Resultados do transfer learning na ViT

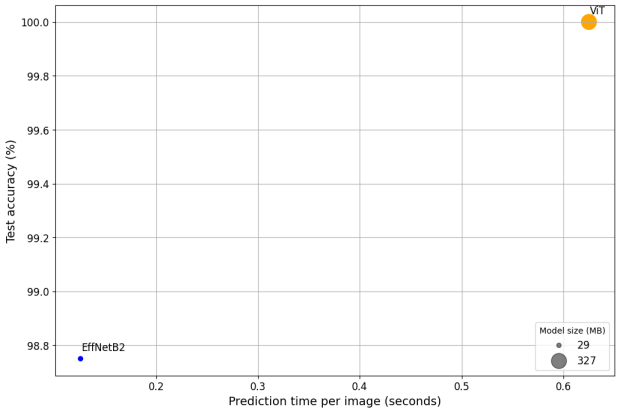

Figure 8: Comparação entre tamanho das redes, velocidade e performance

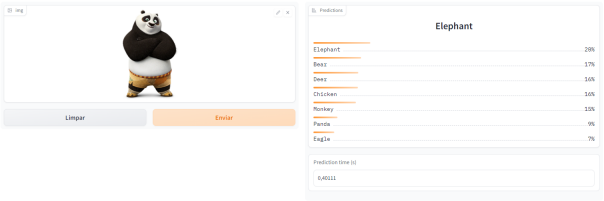

Figure 9: Resultados da EffNet quando sua entrada é uma animal estilizado

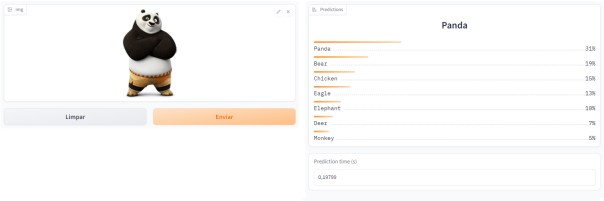

Figure 10: Resultados do ViT quando sua entrada é um animal estilizado

interessante, por exemplo, ela seria ideal para uma aplicação de visão em sistemas embarcados.

O ViT, apesar de ser mais pesado, possui capacidades que, segundo testes preliminares feitos por nós, vão muito além dos requerimentos pedidos pelo problema. Por exemplo, nas figuras 9, 10 e 11,12, conseguimos perceber claramente que o ViT possui uma capacidade muito superior de extrair o animal da imagem, mesmo que ele esteja estilizado ou que nem sequer represente o animal de verdade, mostrando que seu tamanho superior e seu tempo de processamento mais lento não são em vão. Vale ressaltar que esses testes foram preliminares e que poderiam ter uma análise aprofundada em um seguinte estudo.

Dito tudo isso, conseguimos superar o baseline proposto pelo projeto anterior e conseguimos aplicar os conhecimentos adquiridos em sala de aula à um projeto de visão computacional, sendo assim, podemos dizer que o experimento foi um sucesso.

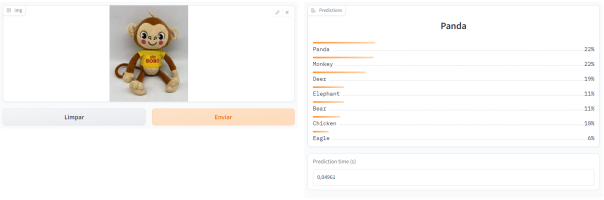

Figure 11: Resultados da EffNet quando sua entrada é um animal de pelúcia

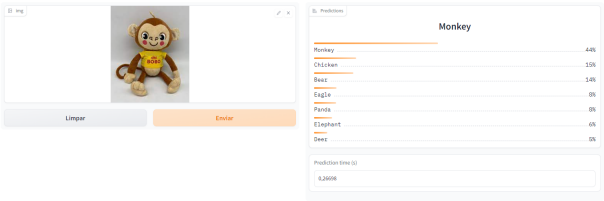

Figure 12: Resultados do ViT quando sua entrada é um animal de pelúcia

References

[1] PAN, S. J.; YANG, Q. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2009.

[2] TAN, M.; LE, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. In: *International Conference on Machine Learning (ICML)*. [S.l.: s.n.], 2019.

[3] TOUVRON, A. et al. Going deeper with image transformers. *arXiv preprint arXiv:2103.17239*, 2021.

[4] DOSOVITSKIY, A. et al. An image is worth 16x16 words: Transformers for image recognition at scale. In: *International Conference on Learning Representations (ICLR)*. [S.l.: s.n.], 2021.