Uma análise comparativa entre os modelos ViT16 e Dino V2 treinados com o dataset Mapillary Street-Level Sequences em tarefas de geolocalização

Angelo F. Oliveira, Augusto M. Grohmann

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)

angelo.fernandes.oliveira@gmail.com, augustomgrohmann@gmail.com

Abstract. *This paper presents a comparative analysis between ViT16 and DinoV2 architectures applied to image geolocation tasks, utilizing the Mapillary Street-Level Sequences dataset. The methodology addressed the critical issue of data leakage in image time series through a spatial split strategy based on the K-Means algorithm and weighted sampling. Experiments demonstrated the superiority of the self-supervised DinoV2 model, which achieved validation accuracy exceeding 92% and showed greater global generalization capabilities, in contrast to the ViT16 model, which exhibited signs of overfitting and lower performance (83.5%). Furthermore, the study discusses the divergence of geographical results obtained compared to the original DinoV2 literature, particularly regarding the African and American continents, suggesting pathways for future multimodal approaches.*

Keywords: Geolocation, Vision Transformers, DinoV2, Computer Vision, Deep Learning.

Resumo. *Este trabalho apresenta uma análise comparativa entre as arquiteturas ViT16 e DinoV2 aplicadas à tarefa de geolocalização de imagens, utilizando o dataset Mapillary Street-Level Sequences. A metodologia abordou o problema crítico de data leakage em séries temporais de imagens através de uma estratégia de divisão espacial baseada no algoritmo K-Means e amostragem ponderada. Os experimentos demonstraram a superioridade do modelo auto-supervisionado DinoV2, que atingiu acurácia de validação superior a 92% e demonstrou maior capacidade de generalização global, contrastando com o modelo ViT16, que apresentou sinais de overfitting e desempenho inferior (83,5%). Adicionalmente, discute-se a divergência dos resultados geográficos obtidos em relação à literatura original do DinoV2, especialmente nos continentes africano e americano, sugerindo caminhos para abordagens multimodais futuras.*

Palavras-chave: Geolocalização, Vision Transformers, DinoV2, Visão Computacional, Deep Learning.

1. Introdução

Tarefas de visão computacional tradicionalmente analisam imagens de entrada focando em descritores visuais para a extração de características. Embora arquiteturas

baseadas em Transformers (como os Vision Transformers) tenham alcançado desempenho e destaque notável, ainda estão sub exploradas as diferenças entre os dois.

Neste trabalho, propõe-se uma análise comparativa entre o modelo ViT16 e duas variações da arquitetura DinoV2 (uma com backbone congelado e outra com ajuste fino/fine-tuning), no contexto de tarefas de geolocalização - se, dado uma imagem de entrada, os modelos pré-treinados conseguem identificar a localização de origem. O objetivo principal é avaliar se há alguma diferença entre os dois, quais as melhores situações para usar um, ao invés de outro, suas semelhanças, etc.

2. Técnica utilizada

Como já citado, a implementação proposta é comparar o modelo de Vision Transformer ViT16 a duas abordagens do modelo DinoV2: uma utilizando o backbone congelado como extrator de características fixas ('Frozen') e outra aplicando ajuste fino ('Fine-Tuning') em todas as camadas para especialização na tarefa de cidades. Para isso, dentro do Pipeline Visual utiliza-se um modelo pré treinado, em que o processamento das imagens se baseia em redimensionamento e normalização e a saída retorna o vetor de características visuais (embeddings).

A camada de classificação (Head) acoplada ao backbone do Vision Transformer foi estruturada como um Perceptron Multicamadas (MLP),consistindo de uma camada linear de projeção para 512 dimensões, seguida por uma função de ativação ReLU e uma camada de *Dropout* com taxa de 0.5 para regularização, finalizando com a camada linear de saída correspondente às classes das cidades. Essa configuração visa reduzir o *overfitting* em representações de alta dimensionalidade.

Para promover a invariância do modelo a variações de iluminação, ângulo e enquadramento, aplicou-se um pipeline de Data Augumentation durante o treinamento. As transformações incluíram cortes aleatórios redimensionados (RandomResizeCrop) , rotações de até 20 graus, espelhamento horizontal e Color Jitter para variações aleatórias de brilho, contraste e saturação, visando representar cenários e variações reais das fotos. Tais operações forçam a rede a focar em características estruturais das cidades em vez de memorizar padrões de pixels estáticos, das câmeras utilizadas e da iluminação.

3. Implementação

Para a implementação e análise desse sistema, foi iniciado um projeto em python na estação de trabalho do pesquisador que possuía GPU e um .IPYNB para o pesquisador que não possuía pudesse utilizar as T4 disponíveis do Google Colab. Foi utilizada a versão Python 3 e as bibliotecas pandas, numpy, torch, sklearn e tqdm principalmente.

Para o modelo ViT16 e DinoV2 Frozen, utilizou-se um tamanho de batch de 128. Entretanto, devido às restrições de memória ao descongelar o backbone, o modelo DinoV2 Fine-Tuned utilizou um batch reduzido de 32. O número de épocas foi 5 para

todos os modelos. A taxa de aprendizado para o ViT16 foi de 1e-4 e para o DinoV2 Frozen de 1e-3. No modelo DinoV2 Fine-Tuned, aplicou-se uma estratégia de aprendizado diferencial: 5e-6 para o backbone (preservando o pré-treino) e 1e-4 para o classificador (head).

Um dos problemas mais importantes a ser tratado foi o data leakage. Muitas das fotos para cada cidade foram tiradas continuamente, de forma que uma divisão aleatória provavelmente comprometeria a validação do modelo. Visando mitigar isso, implementou-se uma estratégia de divisão espacial. Utilizou-se o algoritmo K-Means nas coordenadas GPS das imagens que as possuíam para segregar geograficamente as amostras, garantindo que locais físicos específicos não apareceriam simultaneamente no treino e na validação. Para as cidades que não tinham as informações de GPS, a separação foi feita com base na sequência de captura. Adicionalmente, realizou-se uma filtragem nos metadados para remover imagens noturnas e capturas internas de painéis de controle, focando estritamente em características visuais urbanas diurnas, e não em outros fatores externos que poderiam enviesar o modelo.

Dada a disparidade na quantidade de imagens entre cidades, adotou-se uma estratégia de amostragem ponderada (Weighted Random Sampler). Calculou-se o peso de cada amostra como o inverso da frequência de sua classe (1/Nclasse), forçando o modelo a visitar exemplos de classes minoritárias com a mesma frequência estatística das classes majoritárias durante a construção de cada batch Isso foi essencial para evitar que o modelo desenvolvesse vieses em direção às cidades com maior volume de dados.

Após a filtragem e remoção das classes com amostragem insuficiente (Nairóbi e Amman, com < 200 imagens cada), o dataset final consolidou-se em 28 classes (cidades). O volume total de dados processados resultou em 286.405 imagens válidas, divididas estrategicamente em 232.997 amostras para treinamento e 53.408 para validação.

O treinamento foi conduzido utilizando o otimizador AdamW, superior ao Adam padrão para generalização em Transformers. O decaimento de peso (weight decay) foi configurado em 1e-3 para o ViT e Dino Frozen, aumentando para 1e-2 no DinoV2 Fine-Tuned para maior regularização. A taxa de aprendizado seguiu um agendador do tipo Cosine Annealing, decaindo suavemente ao longo das épocas para facilitar a convergência em mínimos locais mais estáveis. Para a função de perda, utilizou-se a Cross Entropy com Label Smoothing de 0.1, uma técnica de regularização que penaliza a confiança excessiva do modelo nas predições, essencial para lidar com a similaridade visual entre cidades europeias.

Para facilitar a validação qualitativa e a demonstração dos resultados em tempo real, desenvolveu-se também uma interface gráfica denominada 'GeoWarlock'. A aplicação, construída com a biblioteca CustomTkinter, permite o carregamento de imagens inéditas e a seleção dinâmica entre os modelos treinados (ViT16, DinoV2 Frozen e Fine-Tuned). A ferramenta exibe as três classes com maior probabilidade de acerto,

permitindo uma análise visual da confiança dos modelos (Top-3) fora do ambiente de notebooks de desenvolvimento.

4. Resultados

Tivemos resultados variados durante o treinamento, na Tabela 1 organizamos como foi a evolução do treino do ViT16 com base nas épocas. Também geramos um gráfico para melhor visualização na figura 1. Num geral, os resultados apresentaram resultados altos de treino, no entanto um pouco menores em validação (e optamos por não treinar mais épocas já que a partir da época 4 para 5 percebemos redução na acurácia de validação, que é um forte índice de possível overfitting no modelo).

Para a avaliação dos resultados e inferência final, empregou-se a técnica de Test-Time Augmentation (TTA) visando maximizar a robustez das predições e eliminar ruído. Em vez de uma única passagem (single-crop), cada imagem de teste foi submetida a um processo de FiveCrop, gerando cinco recortes distintos (os quatro cantos e o centro da imagem). A predição final foi obtida através da média das probabilidades softmax resultantes desses cinco recortes, garantindo que a classificação considerasse o contexto global da cena e não apenas o centro geométrico.

Para os testes finais, utilizamos 5 capturas de telas diretas do Google Street View em pontos aleatórios para cada uma das cidades do dataset, de forma a simular fotos reais de rua. Tais resultados foram compilados na tabela 2. Também foi analisado os acertos por continente, por possuírem geralmente maior semelhança dentre si. Alguns valores nos surpreenderam, como a qualidade na Oceania/África (ambos com uma única cidade). Para o Vit16, a acurácia total por país para prints externos foi 63/140 = 45.0%.

Época/Métrica	Acurácia de treino	Acurácia de validação
1	81,8%	75,7%
2	93,7%	78,7%
3	96,4%	80,1%
4	98,1%	83,6%
5	98,9%	83,5%

Tabela 1. Resultados de cada época de treinamento do ViT16

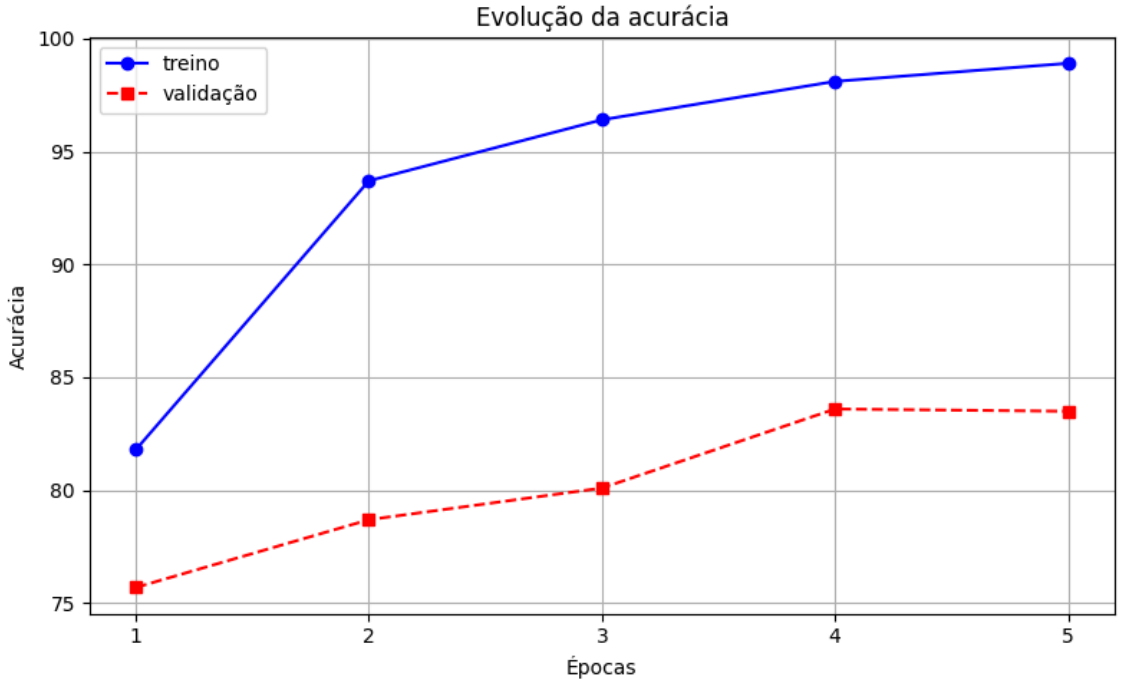

Figura 1. Gráfico de treino do ViT16.

Continente/Métricas	Total	Corretos	Acurácia
Europa	60	50	83%
Ásia	25	13	52%
América do Sul	10	7	70%
América do Norte	35	25	71%
Oceania	5	4	80%
África	5	4	80%

Tabela 2. Resultados por continente do modelo do ViT16 com prints externos.

Além do ViT16 clássico, como já foi citado, também foram testado os modelos Dino-V2, que obteveram resultados surpreendentes, como por exemplo a acurácia nos dados de validação superior à acurácia de treino na primeira época. O modelo Dino-V2 Frozen obteve a melhor acurácia total para testes externos, apesar da pequena margem. Seguindo o mesmo percurso do ViT16, na quinta época pode-se perceber que os

valores de acurácia da validação começaram a reduzir, enquanto os de acurácia de treino seguiram aumentando, o que pode indicar o overfitting do modelo. Tais resultados foram compilados na Tabela 3/Imagem 2. Na Tabela 4, foram compilados os resultados por continente do modelo, que também obteve resultados muito mais satisfatórios. Para o Dino V2 Frozen, a acurácia total por país para prints externos foi 103/140 = 73.6%.

Época/Métrica	Acurácia de treino	Acurácia de validação
1	88,7%	91,2%
2	92,9%	91,2%
3	94,4%	91,7%
4	95,4%	92,8%
5	96,1%	92,6%

Tabela 3. Resultados de cada época de treinamento do Dino V2 Frozen

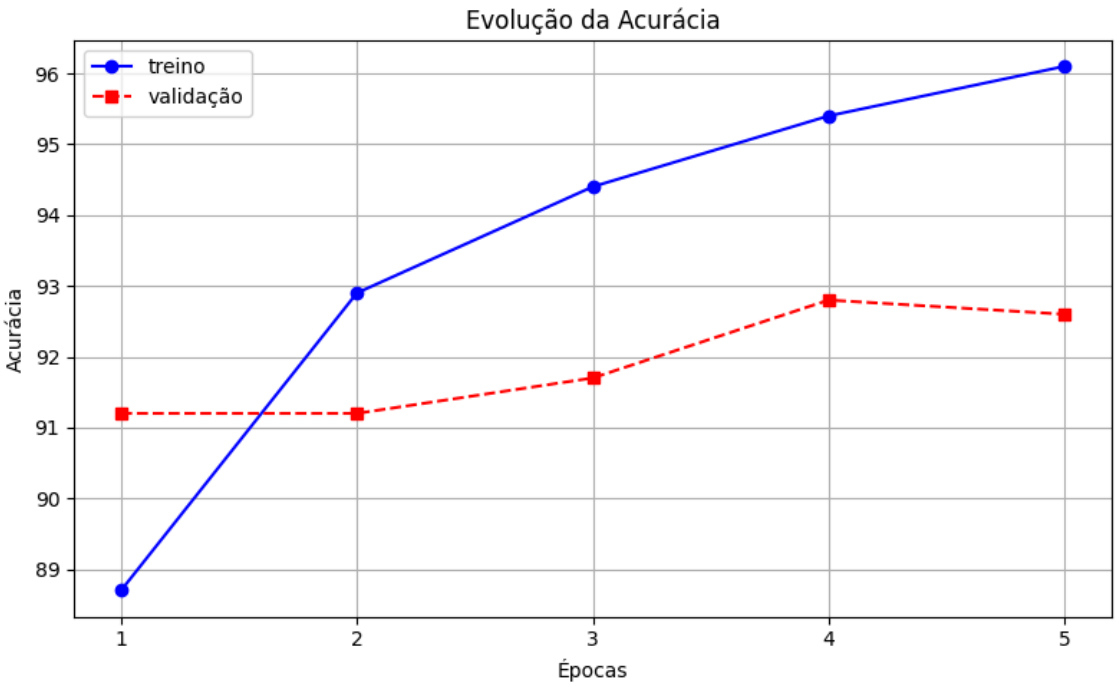

Figura 2. Gráfico de treino do DinoV2 Frozen.

Continente/Métricas	Total	Corretos	Acurácia
Europa	60	60	100%
Ásia	25	23	92%
América do Sul	10	9	90%
América do Norte	35	29	82%
Oceania	5	5	100%
África	5	5	100%

Tabela 4. Resultados por continente do modelo do DinoV2 Frozen com prints

Quanto ao DinoV2 Fine-Tuned, a acurácia de treino cresceu ainda mais rápido, enquanto a validação cresceu continuamente mas lentamente, indicando overfitting. A acurácia no teste com prints externos foi apenas um acerto menor, mas as estatísticas de acerto por continente diminuíram, indicando erros mais extremos, outro sinal que indica que houve overfitting, pois o modelo não está realmente aprendendo as características do local. Para o Dino V2 Fine-Tuned, a acurácia total por país para prints externos foi 102/140 = 72.9%

Época/Métrica	Acurácia de treino	Acurácia de validação
1	94.0%	91.5%
2	98.4%	92.8%
3	99.2%	92.8%
4	99.6%	93.9%
5	99.8%	94.6%

Tabela 5. Resultados de cada época de treinamento do Dino V2 Fine-Tuned

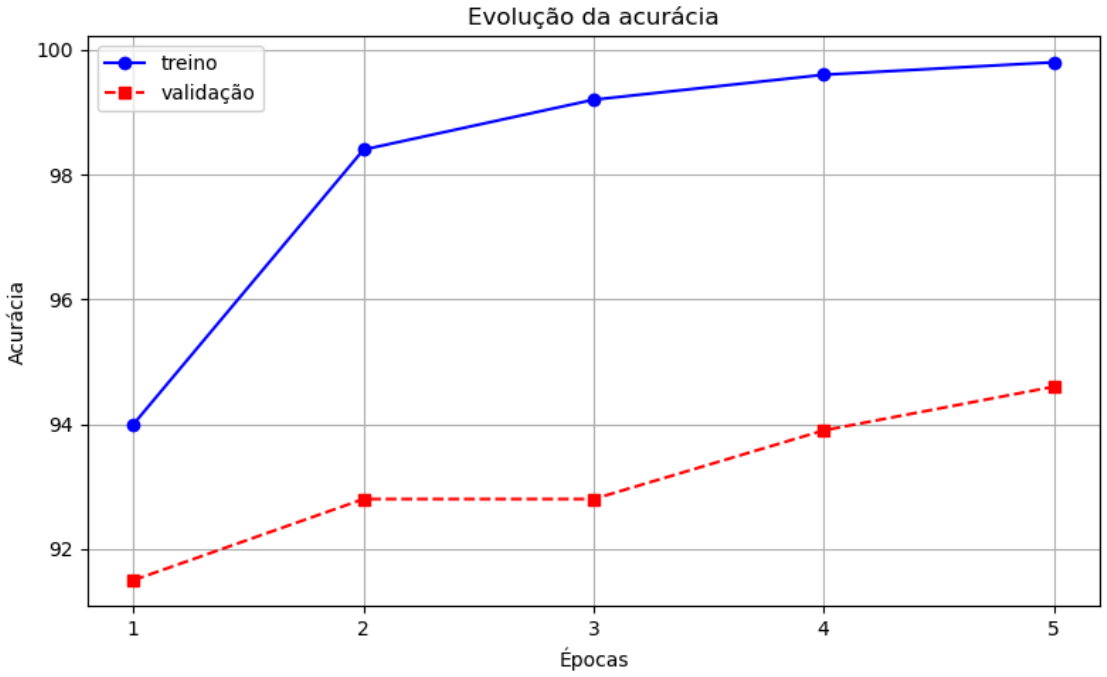

Figura 3. Gráfico de treino do DinoV2 Fine-Tuned.

Continente/Métricas	Total	Corretos	Acurácia
Europa	60	57	95%
Ásia	25	19	76%
América do Sul	10	8	80%
América do Norte	35	30	85%
Oceania	5	4	80%
África	5	5	100%

Tabela 6. Resultados por continente do modelo do DinoV2 Frozen com prints externos.

5. Conclusões

Dados os resultados obtidos, conclui-se que os modelos Dino se saem muito melhor em tarefas de geolocalização, ao tratá-los no Dataset Mapillary Street-Level Sequences, uma vez que a acurácia de todos os continentes foi melhor, sem contar na acurácia dos dados de validação num geral, que foram muito altos desde o início.

Tais resultados corroboram com alguns trabalhos relacionados, como o artigo “DINOv2: Learning Robust Visual Features without Supervision”, que apresentou o modelo DINOv2 como uma extensão para o estado da arte da época, por meio de utilização de um treinamento híbrido e otimizado de aprendizado auto-supervisionado. No entanto, algumas afirmações dele foram diferentes dos nossos resultados, uma vez que ele observou um desempenho melhor nas Américas e Europa, e um desempenho pior na África e Ásia; enquanto no nosso trabalho os piores resultados ficaram com as Américas e Ásia, e os melhores na Europa, África e Oceania.

Por fim, algumas coisas chamam a atenção no modelo do DinoV2 Frozen, como a acurácia de 100% na Europa, por exemplo, e a acurácia mais baixa ser na América do Norte, contrastando com os resultados publicados pelo time de pesquisas da Meta AI, ao lançar o DINOv2 ao mundo.

6. Possibilidades de melhoria/extensão

Algumas possibilidades para futuros trabalhos poderiam ser a utilização de bibliotecas/frameworks que possam realizar abordagens multimodais em tarefas de geolocalização, por exemplo: o uso de OCR para que os resultados do modelo Vit16, por exemplo, possa ser melhorado; Uma possibilidade é utilizar EasyOCR ou PaddleOCR, que possuem suporte a aplicações multilíngue e possam mesclar embeddings de texto com embeddings de imagem para adquirir conhecimento sobre imagens de entrada e classificar o local de origem de maneira precisa.

7. Referências

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P., Li, S., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A. & Bojanowski, P. (2024). "DINOv2: Learning Robust Visual Features without Supervision". *Transactions on Machine Learning Research*.

Keetha, N., Mishra, A., Karhade, J., Jatavallabhula, K. M., Scherer, S., Krishna, M., & Garg, S. (2023). "AnyLoc: Towards Universal Visual Place Recognition". *IEEE Robotics and Automation Letters*, 8(12), 8272-8279.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. & Houlsby, N. (2021). "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". *International Conference on Learning Representations (ICLR)*.

Warburg, F., Hauberg, S., Lopez-Antequera, M., Gargallo, P., Kuang, Y., & Civera, J. (2020). "Mapillary Street-Level Sequences: A Dataset for Lifelong Place Recognition". *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2626-2635.

Loshchilov, I., & Hutter, F. (2019). "Decoupled Weight Decay Regularization" (AdamW). *International Conference on Learning Representations (ICLR)*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). "Rethinking the Inception Architecture for Computer Vision" (Label Smoothing). *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2818-2826.