Austin Carnahan
Deep Learning
FA 2024

Project Proposal:
BoiCLIP Lite: Optimizing BioCLIP For Mobile Environments

Optimizing large vision models for mobile platforms presents a significant challenge due to the computational demands of deep learning. BioCLIP, a powerful model based on OpenCLIP for biological classification, excels in classifying organisms and learning hierarchical relationships in the tree of life but is too resource-intensive for mobile deployment. This project will explore reducing the size and computational requirements of BioCLIP through knowledge distillation or transfer learning, aiming to make it feasible for mobile devices. Deploying this optimized model on mobile platforms would enable biologists to classify and discover species in real time, leveraging mobile sensors for field data collection and analysis

Data Sources

The **TreeOfLife-10M** dataset, introduced in the original BioCLIP paper, is a large-scale collection of over 10 million images covering more than 450,000 taxa. The dataset was specifically curated to train BioCLIP, enabling it to learn not only species-level distinctions but also hierarchical relationships. This biological diversity makes TreeOfLife-10M ideal for training models that generalize well to fine-grained classification tasks, as it reflects the inherent structure and complexity of the tree of life.

Methods

---

1) Use knowledge distillation to create a smaller version of the BioCLIP model. The student model will be trained to mimic the larger, pre-trained BioCLIP model's outputs, using the TreeOfLife-10M dataset. This dataset provides a wide range of biological images, with the student model learning to recognize and classify organisms by following BioCLIP's hierarchical predictions across more than 450,000 species.

After training, the student model will be optimized for mobile use with techniques like quantization and model pruning to shrink the model’s size and make it run faster. We’ll then test how well it performs on mobile devices, checking its speed, memory use, and accuracy to ensure it's efficient without sacrificing too much in terms **of performance.**

2) The mobile-optimized model will be implemented using **TensorFlow.js** and the **tfjs-react-native** platform adapter. This will involve converting the trained student model to TensorFlow.js format, integrating it with React Native's camera and UI components, and optimizing for performance on mobile hardware. The goal is to publish a reusable package to npm, tentatively named "react-bio-clip" to make the model accessible to other developers.

Evaluation

---

1. **Model Performance**
   Compare the performance of the lightweight model against the original BioCLIP model on the Tree Of Life dataset. Evaluate top-1 and top-5 accuracy of classifications.

   Since BioCLIP learns hierarchical relationships, evaluate whether the student model retains this ability to structure its predictions according to the ‘tree of life’.

2. **Mobile Device Performance**
   Evaluate the real-world performance on a mobile device. Some key factors include model size (total size of weights and architecture) and prediction latency for images at different resolutions.

Readings

CLIP Models:

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021, February 26). ***Learning transferable visual models from natural language supervision***. arXiv.org. https://arxiv.org/abs/2103.00020

Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., & Jitsev, J. (n.d.). ***Reproducible scaling Laws for Contrastive Language-Image Learning***. arxiv.org. https://arxiv.org/abs/2212.07143

Stevens, S., Wu, J., Thompson, M. J., Campolongo, E. G., Song, C. H., Carlyn, D. E., Dong, L., Dahdul, W. M., Stewart, C., Berger-Wolf, T., Chao, W., & Su, Y. (2023, November 30). ***BioCLIP: a Vision foundation model for the Tree of Life.*** arXiv.org. https://arxiv.org/abs/2311.18803

Knowledge Distillation:

Gou, J., Yu, B., Maybank, S. J., & Tao, D. (2021). ***Knowledge distillation: a survey.*** International Journal of Computer Vision, 129(6), 1789–1819. https://doi.org/10.1007/s11263-021-01453-z

Hinton, G., Vinyals, O., & Dean, J. (2015, March 9). ***Distilling the knowledge in a neural network***. arXiv.org. https://arxiv.org/abs/1503.02531
Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2020, December 23). ***Training data-efficient image transformers & distillation through attention.*** arXiv.org. https://arxiv.org/abs/2012.12877

Reducing Model Size and Latency:

Shu, H., Li, W., Tang, Y., Zhang, Y., Chen, Y., Li, H., Wang, Y., & Chen, X. (2023, December 21). ***TinySAM: Pushing the envelope for efficient segment anything model***. arXiv.org. https://arxiv.org/abs/2312.13789

Zhao, X., Ding, W., An, Y., Du, Y., Yu, T., Li, M., Tang, M., & Wang, J. (2023, June 21). ***Fast segment anything***. arXiv.org. https://arxiv.org/abs/2306.12156

Zhang, C., Han, D., Qiao, Y., Kim, J. U., Bae, S., Lee, S., & Hong, C. S. (2023, June 25). ***Faster segment anything: towards lightweight SAM for mobile applications***. arXiv.org. https://arxiv.org/abs/2306.14289

Hernandez, D., Kaplan, J., Henighan, T., & McCandlish, S. (2021, February 2). ***Scaling laws for transfer.*** arXiv.org. https://arxiv.org/abs/2102.01293

Shuoyuan Wang, Yixuan Li, Hongxin Wei (Oct 2024). ***Understanding and mitigating miscalibration in prompt tuning for Vision-Language models.*** (n.d.). https://arxiv.org/html/2410.02681v1

EVE: Efficient Vision-Language Pre-training with Masked Prediction and Modality-Aware MoE. (n.d.). https://arxiv.org/html/2308.11971v2

Jia, C., Yang, Y., Xia, Y., Chen, Y., Parekh, Z., Pham, H., Le, Q., V., Sung, Y., Li, Z., & Duerig, T. (2021, February 11). **Scaling up Visual and Vision-Language representation learning with noisy text supervision.** arXiv.org. https://arxiv.org/abs/2102.05918

Deploying Models on Mobile Devices:

Use TensorFlow.js in a React Native app. (n.d.). TensorFlow. https://www.tensorflow.org/js/tutorials/applications/react_native