Mobile Scene Understanding: On-Device Deployment of Scene Description Models

Team Name: Living on the Edge

Boyi Qian, Eesha Shetty, Danyating (Amanda) Shen, Vibha Masti

October 13, 2023

1 Motivation

The goal is to create an on-device near real-time image scene description model to run on hardware that is available to everyone. The main purpose of this application is for it to serve as an accessibility tool to assist the visually-impaired. The tool would be most useful if it were to run on a mobile phone, which is a device most people always have on hand. The motivation to run this model on device is mainly to run offline while maintaining privacy, while also remaining lightweight with low inference latency.

2 Task Definition and Problem Setup

2.1 Target Hardware

Our target hardware is an iPhone. The main reason for choosing the iPhone is because it is the most realistic use-case for an on-device version of this model. Most people have a personal phone and would not need to purchase an additional device for this assisted accessibility. The Apple Neural engine is quite powerful and it will allow us to run our models on device. There is also good support for CoreML [1] for us to integrate our models.

2.2 I/O

Q. What are the Input and Output modalities?

Input Modality: Image
Output Modality: Text (scene description)

Q. What existing tools will you use to convert device inputs (that are not a core part of your project) to a format readable by the model, and vice versa?

Since our expected input is an image itself, we expect to only perform basic preprocessing required for it to be read by PyTorch Models (there already are functions in PyTorch for this). Our target output is text, which we can obtain from a softmax/argmax over the final linear layer of the model.

Q. Do you have or know of existing libraries?

For training and pre-processing, we will be using PyTorch.
For deployment and inference on iPhones, we will be using CoreML

Q. Will you need to perform training off-device? If so, do you need cloud compute credits (GCP or AWS), and if so how much?

Yes, we will be performing training off-device, we would only stick to deployment and inference on-device. We should be fine with the AWS credits provided to us by the class.

3 Research questions and experimental design

Q. Clearly state research questions (hypotheses)

1. Can we deploy a near-real time/low latency image scene description model on-device?
2. What is the compromise on accuracy/speed?

Q. How will you design your experiments to answer questions about datasets?

1. MS COCO [2]

2. Flickr30k [3]
3. Visual Genome [4]: (size: 100000, high quality), a knowledge base, an ongoing effort to connect structured image concepts to language
4. YFCC100M [5]: contains a total of 100 million media objects, of which approximately 99.2 million are photos and 0.8 million are videos, all of which carry a Creative Commons license
5. WIT [6]

Q. How will you design your experiments to answer questions about hardware?

Our target deployment hardware is an iPhone. We plan to run experiments on Xcode as it allows us simulate an iPhone without having to create an Apple Developer account [7]. For training, we plan on using cloud resources.

Q. How will you design your experiments to answer questions about ablations?

1. CLIP [8] has wonderful performance on zero-shot or few-shot learning but we always can using a suitable dataset to fine-tune the model and improve performance on downstream task. We are stilling looking for a special dataset to use in finetuning.
2. The current vision backbone of clip is ViT [9] but on device, it might be too large to store and run. So we will try several other architectures (eg: ResNet50 [10]) to replace the image encoder part. This idea will also be use in the text encoder part.
3. When we use a smaller image encoder, the quality of image embedding will decrease, we will also try to play around with image embeddings for example, apply an additional neural network on top of it to modify the image embedding.

Q. Other experiments or analyses?

We will do lots of runtime and size analysis. We will also create some customized images to see the predicted performance. If we are satisfied with our model size and model performance, we are going to move to see develop a solution in the video scenario.

4 Related Work and Baselines

4.1 Learning Transferable Visual Models From Natural Language Supervision [8]

This paper proposed an approach for pre-training visual models using language supervision, called CLIP (Contrastive Language-Image Pre-training). The multimodal model was pre-trained on image-caption pairs. They use transformer based encoders and decoders, with one encoder each for images and text.

The core idea behind CLIP is the use of a contrastive loss function during training. This loss function encourages the model to bring together matching pairs of text and image embeddings while pushing apart non-matching pairs. If a text description corresponds to an image, their embeddings should be close in the shared space, and if not, their embeddings should be far apart.

4.2 Geolocation Estimation of Photos using a Hierarchical Model and Scene Classification [11]

The paper aims to predict the geographical location of photos without any prior knowledge. This is challenging due to the vast variations in images, such as different daytimes, objects, or camera settings.

Many existing methods for geolocation estimation are limited to specific areas, imagery, or global landmarks. Few approaches predict GPS coordinates without such limitations.

The authors introduce deep learning methods that treat geolocalization as a scene classification problem. They divide the earth into geographical cells and propose to use hierarchical knowledge of these partitions. Additionally, they consider the content of the photo, such as whether it's an indoor, natural, or urban setting.

The earth is subdivided into geographical cells, treating geolocalization as a classification problem. Hierarchical knowledge from multiple partitionings is exploited. The content of the photo (e.g., indoor, natural, urban) is taken into account to provide contextual information at different spatial resolutions. The convolutional neural network (CNN) is used in the learning process.

One of the paper's strengths is its use of contextual information (e.g., distinguishing between indoor, natural, or urban settings) to help them understand the context of the multiple objects in the image. We could use a hierarchical approach to first classify broader categories and then delve into specifics.

4.3 Efficient Image Captioning for Edge Devices [12]

This paper was made in collaboration with Huawei and it proposed a lightweight image captioning model, which they call LightCap. Many image captioning are very compute and memory intensive as they rely on heavy object detectors and cross-modal fusion networks. The authors use a CLIP [8] backbone to speed up the process and extract grid features. They were able to create a small model of 40M parameters and deploy it on a mobile device. They were also able to achieve a near-real time inference speed of 188 ms.

The authors pre-trained their model from scratch, and then fine-tune it on an annotated dataset. They also perform ensembled knowledge distillation during both pre-training and fine-tuning.

While this paper is very close to our target use case, they have not made any code publicly available. However, they do detail the specifics of their architecture to a certain degree, but we would not be able to use make use of their pre-trained models. They also have not

4.4 Mobile ViT [13]

Vision Transformers are powerful but usually need lots of computing power. That's a problem for phones and other devices with limited resources. So, this paper introduces a solution: a lightweight transformer called MobileVIT.

It's designed to work just as well as the big ones but uses fewer computer parts. This MobileVIT can understand both small and big details in pictures, and it does this with fewer parts. It combines regular computer tricks like convolutions with the transformer's magic to learn about different things in pictures.

What's nice is that this MobileVIT is made by Apple, and it's easy to use in projects because it can be deployed with CoreML. So, it could be a perfect fit for our project as a strong starting point.

4.5 A Multi-Task Neural Architecture for On-Device Scene Analysis [14]

This article by Apple is a walkthrough of the Apple Neural Scene Analyzer (ANSA) to maintain their on-device scene analysis pipelines in production. It is also available to use by developers through their Vision APIs [15].

Their backbone model is build on MobileNetv3 [16], but optimized for Apple's hardware. Their model is only 16M parameters large, and the inference speed is in the order of 10ms. They prune and quantize their model weights, and benchmark their model against their internal implementation of ViT-B-16.

This article gives a good high-level understanding of how a vision model can be deployed on iPhones and the potential architectures that have shown promise.

4.6 Camera2Caption: A Real-Time Image Caption Generator [17]

This paper take common approaches in Image Captioning and presents a simplistic encoder and decoder based model, optimized for low-power devices. While their implementation is pretty impressive, there are a few issues they face.

The first issue is they train and perform inference with only one model instead of an ensemble that most implementations use to improve their score.

The second issue is they don't add visual attention, which could improve their performance but also increases paramaters in the model.

The third issue is they generate captions greedily instead of using beam search which would take more time.

While we could learn a lot from their approaches to aid our implementation, we could also aim to solve some of these issues to improve accuracy while maintaining a lightweight model

5 Potential challenges

5.1 Potential Challenges

1. **Variability in Lighting and Quality:** Photos taken in different lighting conditions or with motion blur might affect the accuracy of scene classification.

2. **Diverse Scenes:** The real world contains an almost infinite variety of scenes, making it challenging to classify every possible scenario accurately.

3. **Limited Training Data:** For some specific or rare scenes, there might be limited training data available.

5.2 Adjustments for Challenges

1. **Data Augmentation:** To handle different lighting conditions and photo qualities, use data augmentation techniques to artificially enhance the training dataset.

2. **Hierarchical Classification:** use a hierarchical approach to first classify broader categories (e.g., indoor vs. outdoor) and then delve into specifics.

3. **Transfer Learning:** Use pre-trained models on large datasets (like ImageNet [18]) and fine-tune them for specific scene classification tasks to overcome limited training data.

4. **Optimized Models:** Use lightweight and optimized neural network architectures (like MobileNet [19]) for faster processing, especially if real-time classification is desired.

5.3 Extensions if Things Go Smoothly

1. **Location-based Context:** Integrate the phone's GPS to provide location-based context to the scene classification. For instance, a beach in California might look different from one in Japan.

2. **Time-based Context:** Use the time metadata from the photo to provide time-based context (e.g., day vs. night, season).

3. **Interactive Feedback:** Allow users to correct misclassifications, which can be used to further train and refine the model.

4. **Integration with Other systems:** Allow other systems like Andriod to use our scene classification service.

6 Milestone and timeline

1. Oct-13-2023 (Lab 2): Baseline model running on device, benchmarked
2. Oct-27-2023 (Project sharing): establish expectations on target performance, framework for evaluation of image captioning set up
3. Nov-03-2023 (Lab 3): Quantized model running on device, performance measurements noted
4. Nov-17-2023 (Lab 4): Pruned model running on device, performance measurements noted
5. Dec-07-2203 (Final Project Report): final project report completed

7 Heilmeier questions

What are you trying to do? Articulate your objectives using absolutely no jargon.

The goal is to create an accessibility application that can describe a scene in an image on-device and in near real time

How is it done today, and what are the limits of current practice?

There exist many image tagging and analysis models on edge devices, but not as much on image descriptions on-device. Google photos performs image tagging, but it is online. Apple photos performs many on-device image computations, and image descriptions would be a natural extension. This could be extended later to real-time video descriptions to work as an assisted vision tool.

What is new in your approach and why do you think it will be successful?

We hope to be able to run this model with a low latency, on-demand, unlike most on-device ML that happens on the photos app on iPhones, probably done by them to save on power. Since we plan to make this model an on-demand one, it can be thougth of as a separate mode on the camera app. Opening the camera app in the scene describe mode should give a near real-time description of th esurrounding, requiring a burst of on-device processing, and hopefully not consuming as much power.

Who cares? If you are successful, what difference will it make?

If the project is successful, we can look at next steps in rela-time video scene understanding to create an accessibility tool for assisted real-time vision. by deploying it on a mobile phone, users would not need to purchase specialized hardware and can simply point their phone cameras to get a description of their surroundings.

What are the risks?

There are risks associated with the fine-tuning of an image model in near-real time, one of the biggest being not achieving our targeted performance goals. We might also face issues compressing large models like CLIP onto an iPhone.

How much will it cost (in compute)?

We plan to use the AWS credits provided to us through the course ($600 in total).

How long will it take?

We should be able to complete our work by the final proposal deadline (Dec-07-2023).

What are the midterm and final "exams" to check for success?

1. Mid-term Goal: Our immediate objective is to integrate our innovative methodology with the baseline model. We aim to ensure a seamless deployment of this integrated model onto our target hardware, including phone devices.

2. Final Goal: Our ultimate aspiration is to fulfill the objectives outlined in our problem statement. We anticipate tangible outcomes from our implemented and rigorously tested model. In the ideal scenario, we will present the final Minimum Viable Product (MVP) of our project.

References

[1] `https://developer.apple.com/documentation/coreml`. Accessed: 2023-10-06.

[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755, Springer, 2014.

[3] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 67–78, 2014.

[4] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, *et al.*, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International journal of computer vision*, vol. 123, pp. 32–73, 2017.

[5] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, "Yfcc100m: The new data in multimedia research," *Communications of the ACM*, vol. 59, no. 2, pp. 64–73, 2016.

[6] K. Srinivasan, K. Raman, J. Chen, M. Bendersky, and M. Najork, "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*, pp. 2443–2449, 2021.

[7] `https://developer.apple.com/documentation/xcode/running-your-app-in-simulator-or-on-a-device`. Accessed: 2023-10-06.

[8] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[9] A. Kolesnikov, A. Dosovitskiy, D. Weissenborn, G. Heigold, J. Uszkoreit, L. Beyer, M. Minderer, M. Dehghani, N. Houlsby, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[11] E. Muller-Budack, K. Pustu-Iren, and R. Ewerth, "Geolocation estimation of photos using a hierarchical model and scene classification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 563–579, 2018.

[12] N. Wang, J. Xie, H. Luo, Q. Cheng, J. Wu, M. Jia, and L. Li, "Efficient image captioning for edge devices," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, pp. 2608–2616, 2023.

[13] S. Mehta and M. Rastegari, "Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer," *arXiv preprint arXiv:2110.02178*, 2021.

[14] "A multi-task neural architecture for on-device scene analysis." `https://machinelearning.apple.com/research/on-device-scene-analysis`. Accessed: 2023-10-04.

[15] `https://developer.apple.com/documentation/vision`. Accessed: 2023-10-04.

[16] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, *et al.*, "Searching for mobilenetv3," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1314–1324, 2019.

[17] P. Mathur, A. Gill, A. Yadav, A. Mishra, and N. K. Bansode, “Camera2caption: a real-time image caption generator,” in *2017 international conference on computational intelligence in data science (ICCIDS)*, pp. 1–6, IEEE, 2017.

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.