DI725 Project Phase 2

Ahmet Bekcan
Data Informatics
Middle East Technical University
Ankara, Turkey
ahmet.bekcan@metu.edu.tr

***Abstract*—This report introduces CNNGemma, a lightweight vision-language model that integrates configurable CNN encoders and tokenization methods. Due to hardware and time constraints, pretrained MobileNetV3-Large and EfficientNet-B0 models were fine-tuned on the RISCM dataset using single- and multi-token strategies. The dataset was preprocessed to remove duplicate captions, resulting in over 190,000 image-caption pairs. Benchmarking shows that MobileNetV3-Large offers the highest inference speed, and preliminary results suggest that single-token models generate more coherent captions, likely due to reduced complexity for the language model.**

***Index Terms*—transformers**

I. Introduction

In this phase, two different CNN-based models, MobileNetV3-Large [1] and EfficientNet-B0 [3], are selected as image encoders. Their performance is compared with SigLIP by measuring throughput (FPS). The evaluated hyperparameters include the image encoder (MobileNet vs. EfficientNet) and the tokenization method (single vs. multiple). The codebase is structured to facilitate easy selection of these hyperparameters during model loading. Finally, the dataset is examined in detail, and duplicate captions are removed.

II. Preprocessing Dataset

After inspecting the dataset, it was observed that some images had duplicate captions, which could introduce bias toward those captions. Therefore, they were removed from the dataset. To identify low-quality descriptions, a pretrained CLIP [2] model was used to compute similarity scores between images and captions. The image-caption pairs with the lowest similarity scores were examined, and it was found that the captions accurately represented the images. As a result, all captions were retained. The final dataset was saved as separate CSV files for training, testing, and validation. As a result of preprocessing, the training set contains 151,296 image-caption pairs, the validation set contains 20,295 pairs, and the test set contains 19,483 pairs.

III. Changes in Project Proposal Methodology

Several changes have been made to the project proposal due to limited hardware and time resources. PaliGemma will not be trained on the dataset as originally planned. Instead, pretrained MobileNet and EfficientNet models will be fine-tuned on the RISCM dataset using two different tokenization methods. The first method involves extracting the final layer of the CNN models and passing it to the language decoder as a single image token. The second method involves extracting multiple tokens from the final feature layer of the CNN encoders. With these two hyperparameters, CNN architecture and tokenization method, a total of four different models will be trained, and their performances will be compared.

IV. Modeling

In the codebase, a model named CNNGemma is implemented. This model supports different architectures and tokenization methods, which are specified through configuration. A separate JSON configuration file is created for each model variant and saved within the codebase. Based on the architecture defined in the configuration, CNNGemma loads the corresponding pretrained weights as the image encoder. Additionally, the model adapts to the specified tokenization method.

The following approaches are used to obtain single and multiple tokens:

- **Single Token:** The final layer of the CNN encoder, just before the classifier, is extracted and used as a single image token.
- **Multiple Tokens:** The final feature layer of the CNN encoder is reshaped into 49 image tokens.

V. Preliminary Results & Benchmarking

A. Inference Speed

The inference speed of different image encoders was measured using dummy inputs with a batch size of 32. Each dummy input was passed through the image encoders 100 times, and throughput was calculated at each step. The comparison results can be seen in the Figure 1, and Table I

TABLE I
Throughput (FPS) Comparison of Image Encoders

Image Encoder	Throughput (FPS)
MobileNetV3-Large	3487
EfficientNet-B0	1584
SigLIP	219

Among the three encoders, MobileNetV3-Large was found to be the fastest, followed by EfficientNet-B0. Both models are significantly faster than SigLIP, demonstrating that replacing the image encoder with a lightweight alternative can substantially improve inference speed.

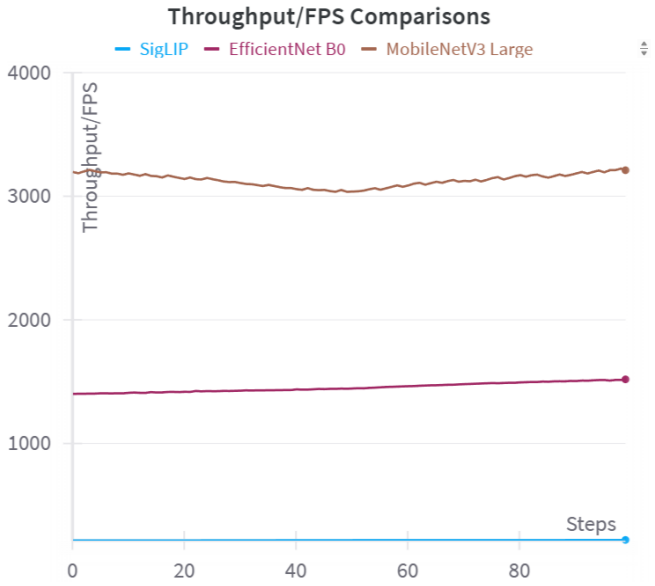

Fig. 1. Throughput comparisons of different image encoders

To verify that the CNNGemma model functions correctly, example inferences were performed for each combination of hyperparameters. The results of these runs can be found in the cnngemma-inference-test.ipynb file located in the notebooks folder of the codebase. Upon inspecting the outputs, it was observed that although the generated captions were inaccurate, models using single-token outputs from the image encoder produced more coherent text compared to those using multiple tokens. This may be due to the language model being less confused by the smaller number of untrained image tokens, suggesting that models using a single token may be easier to train than those using multiple tokens.

VI. Conclusion & Future Work

The inference speed of CNN-based encoders is significantly faster than SigLIP, demonstrating that replacing a transformer-based image encoder with a lightweight CNN model is a promising approach to improve inference efficiency. In the next phase of the project, four different models, varying in CNN architecture and tokenization method, will be fine-tuned and their inference results will be thoroughly compared.

References

[1] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3, 2019.

[2] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[3] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020.