# **Large Multimodal Models for Document Understanding**

## **Business Context:**

This first entry documents algorithms known as large multimodal models (LMMs) that assist in understanding document information better, from image data (e.g. scanned images). Specifically, this entry highlights pre-trained LMMs for document understanding, for immediate use in the work setting.

## **What are LMMs?**

Large Multimodal Models (LMMs) are models that take in input in the form of images and text sequences and provide output in the form of text. They incorporate Large Language Models (LLMs) to align visual perception with the text capabilities of LLMs [5]. These models are large and complex, and training requires expensive computational resources, with large training and inference budgets. For practical purposes, the more affordable option is to use smaller scale LMM models, that use smaller scale LLMs with efficient training and inference [5].

While an exhaustive review of smaller-scale LMMs that have been developed and published is out of scope for this entry, a family of models called TinyLLaVA have been explored for their small scale practicality, publication and documentation within the past 12 months and their use of fewer parameters (3 billion parameters or less) with comparable performance to published LMMs with 7 billion parameters [5, 6]. Figure 1 and the following section documents how the TinyLLaVA framework operates.

## **How does a sample LMM work?**

As mentioned previously, TinyLLaVA is the LMM framework of choice that will be the focus of this entry. This framework is made up of a vision encoder to extract data from images, an intermediate connector that connects the vision encoder to the text functionalities of the framework and finally, a small-scale LLM decoder, that decodes the image data into text [5]. These components are all learnable parameters of the framework that can be finetuned for enhanced performance. The full framework is pictured in Figure 1 below.

Image

**Vision Encoder**

**Connector**

**Small-scale LLM**

Embedding Space

Text output

The vision encoder takes in an image and outputs a sequence of (visual) patch features. TinyLLaVA uses two vision encoders: CLIP [3] and SigLIP [6]. This vision encoder can be a Vision Transformer [3, 6, 7] that directly outputs a sequence of patch features or can be Convolutional Neural Networks (CNNs) that output grid features that need a reshape operation to obtain patch features.

The connector maps the visual patch sequences to the text embedding space (below). The connector of the TinyLLaVA framework is designed for effectively leveraging the capability of both the vision encoder and the pre-trained LLM in this framework. A two-layer Multi-Layer Perceptron (MLP) with GELU activation [2] was applied as a connector between the vision encoders and small-scale LLMs. Resamplers were also examined as connectors in this framework [8].

TinyLLaVA uses either Phi2 [1], StableLM-2 [4] and TinyLlama [9] LLMs. The small-scale LLM takes in a sequence of vectors (of variable length N) in the text embedding space, converts this high dimensional data to lower dimensions for processing in the embedding space, and outputs the corresponding text predictions. A tokenizer (to create word tokens) and embedding module is usually bound to the small-scale LLM, mapping text input sequences to the embedding space and from the embedding space to the text output sequences.

**Figure 1.** The TinyLLaVA framework with corresponding feature components.

## **Limitations of this analysis:**

This analysis was limited by a few factors. The first is that only models with associated publications (e.g. published in ARXIV), that were released in the past 12 months were observed, to capture practical models (i.e. available on the website Huggingface) to use immediately, with documentation to support the model’s application.

In addition, only LMMs that contained 3 billion parameters or less were included in this analysis, as deploying LMMs is computationally expensive, and the computational bottleneck arises due to LLMs that scale to billions of parameters [10, 11].

This entry does not include information about the inner mechanics of specific LLMs used within the TinyLLaVA framework, as further specificity is required in the initial business context, to target specific LLM research to this context. In addition, other methods to efficiently train and deploy small-scale LMMs are out of the scope for this work at present.

## **LMM Summary Table:**

The following LMMs are ordered by their model performance, with the highest performing model listed first.

|  |  |  |  |
| --- | --- | --- | --- |
| **Model Name** | **Number of parameters** | **LLM Component** | **Datasets used to train the model** |
| [TinyLLaVA-3.1B](https://huggingface.co/bczhou/TinyLLaVA-3.1B-Pretrain) | 3.19 billion parameters | Phi-2 | [**ShareGPT4V**](https://huggingface.co/datasets/Lin-Chen/ShareGPT4V)**:** An English-language dataset of GPT-4-Vision-powered multi-modal captions data of 1.2 million rows. Aims to bring LMMs towards GPT-4 vision capabilities[12].  [**LLaVA-Instruct-150K**](https://huggingface.co/datasets/liuhaotian/LLaVA-Instruct-150K)**:** An English-language dataset of GPT-generated multimodal instruction-following data of 100,000 to 1 million rows. Constructed for visual instruction tuning and for building LMMs towards GPT-4 vision/language capabilities[13].  [**LLaVA-Pretrain**](https://huggingface.co/datasets/liuhaotian/LLaVA-Pretrain)**:** An English language dataset constructed for the pretraining stage for feature alignment in visual instruction tuning, to build LMMs towards GPT-4 vision/language capability[13].Acknowledgement of Google LLC is required when sourcing this dataset. |
| [TinyLLaVA-2.0B](https://huggingface.co/bczhou/TinyLLaVA-2.0B) | 2.05 billion parameters | StableLM-2-1.6B |
| [TinyLLaVA-1.5B](https://huggingface.co/bczhou/TinyLLaVA-1.5B) | 1.5 billion parameters | TinyLlama |
| [tiny-llava-v1-hf](https://huggingface.co/bczhou/tiny-llava-v1-hf)[[1]](#footnote-1) | 1.41 billion parameters |  |

## **References:**

1. Li, Y., et al., *Textbooks are all you need ii: phi-1.5 technical report.* arXiv preprint arXiv:2309.05463, 2023.

2. Hendrycks, D. and K. Gimpel, *Gaussian error linear units (gelus).* arXiv preprint arXiv:1606.08415, 2016.

3. Radford, A., et al. *Learning transferable visual models from natural language supervision*. in *International conference on machine learning*. 2021. PMLR.

4. Team, S.A.L. *Stable LM 2 1.6B*. Available from: https://huggingface.co/stabilityai/stablelm-2-1\_6b.

5. Zhou, B., et al., *TinyLLaVA: A Framework of Small-scale Large Multimodal Models.* arXiv preprint arXiv:2402.14289, 2024.

6. Zhai, X., et al. *Sigmoid loss for language image pre-training*. in *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.

7. Dosovitskiy, A., et al., *An image is worth 16x16 words: Transformers for image recognition at scale.* arXiv preprint arXiv:2010.11929, 2020.

8. Li, B., et al., *Mimic-it: Multi-modal in-context instruction tuning.* arXiv preprint arXiv:2306.05425, 2023.

9. Zhang, P., et al., *Tinyllama: An open-source small language model.* arXiv preprint arXiv:2401.02385, 2024.

10. Touvron, H., et al., *Llama 2: Open foundation and fine-tuned chat models.* arXiv preprint arXiv:2307.09288, 2023.

11. Chiang, W.-L.a.L., Zhuohan and Lin, Zi and Sheng, Ying and Wu, Zhanghao and Zhang, Hao and Zheng, Lianmin and Zhuang, Siyuan and Zhuang, Yonghao and Gonzalez, Joseph E. and Stoica, Ion and Xing, Eric P. *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90\%\* ChatGPT Quality*. 2023; Available from: https://lmsys.org/blog/2023-03-30-vicuna/.

12. Chen, L., et al., *Sharegpt4v: Improving large multi-modal models with better captions.* arXiv preprint arXiv:2311.12793, 2023.

13. Liu, H., et al., *Improved baselines with visual instruction tuning.* arXiv preprint arXiv:2310.03744, 2023.

1. This is the legacy (first) model of the TinyLLaVA family of models. [↑](#footnote-ref-1)