Cross-Modal Projection in Multimodal LLMs Doesn't Really Project Visual Attributes to Textual Space

Gaurav Verma [a] **Minje Choi** [a] **Kartik Sharma** [a]
Jamelle Watson-Daniels [b] **Sejoon Oh** [a] **Srijan Kumar** [a]
[a]Georgia Institute of Technology, [b]Harvard University
{gverma, minje.choi, ksartik, soh337, srijan}@gatech.edu
jwatsondaniels@g.harvard.edu

Abstract

Multimodal large language models (MLLMs) like LLaVA and GPT-4(V) enable general-purpose conversations about images with the language modality. As off-the-shelf MLLMs may have limited capabilities on images from domains like dermatology and agriculture, they must be fine-tuned to unlock domain-specific applications. The prevalent architecture of current open-source MLLMs comprises two major modules: an image-language (cross-modal) projection network and a large language model. It is desirable to understand the roles of these two modules in modeling domain-specific visual attributes to inform the design of future models and streamline the interpretability efforts on the current models. To this end, via experiments on 4 datasets and under 2 fine-tuning settings, we find that as the MLLM is fine-tuned, it indeed gains domain-specific visual capabilities, but the updates do *not* lead to the projection extracting relevant domain-specific visual attributes. Our results indicate that the domain-specific visual attributes are modeled by the LLM, even when only the projection is fine-tuned. Through this study, we offer a potential reinterpretation of the role of cross-modal projections in MLLM architectures.

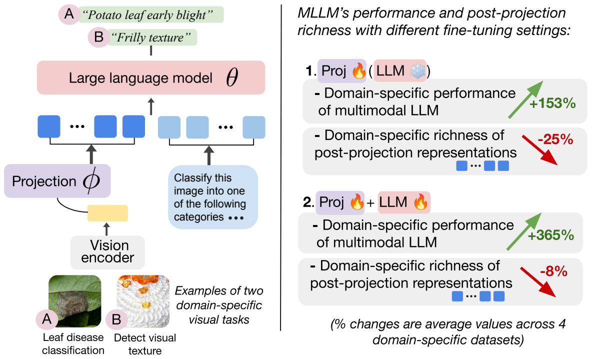
A "Potato leaf early blight"
B "Frilly texture"
Large language model θ
Projection ϕ
Classify this image into one of the following categories •••
Vision encoder
A Leaf disease classification
B Detect visual texture
Examples of two domain-specific visual tasks
MLLM's performance and post-projection richness with different fine-tuning settings:
1. Proj (LLM)
- Domain-specific performance of multimodal LLM
+153%
- Domain-specific richness of post-projection representations
-25%
2. Proj + LLM
- Domain-specific performance of multimodal LLM
+365%
- Domain-specific richness of post-projection representations
-8%
(% changes are average values across 4 domain-specific datasets)

Figure 1: **Overview of our study.** While the MLLM's domain-specific visual capability can be improved using fine-tuning strategies, the domain-specific richness of the image's post-projection representation does not improve. Results indicate that domain-specific visual attributes are predominantly modeled by the LLM parameters (whether frozen or not) and the projection does not necessarily play a role in mapping visual attributes to the LLM space.

1 Introduction

The recent wave of advancements in large language models (LLMs) has equipped them with the ability to "see" images, leading to multimodal large language models (MLLMs) like LLaVA (Liu et al., 2023c), GPT-4(V) (Achiam et al., 2023), and Gemini (Anil et al., 2023). MLLMs unlock the potential to converse with visual data using language. However, existing MLLMs are trained and evaluated for general-purpose multimodal tasks like question-answering on *natural images*[1] (Liu et al., 2023c; AI, 2024), which limits their applicability in specific domains like agriculture and dermatology. MLLMs with domain-specific visual capabilities can transform workflows in several industries, including healthcare, agriculture, circuit design, and satellite imaging (Miotto et al., 2018; Ferentinos, 2018; Anilturk et al., 2023; Kaselimi et al., 2022). While fine-tuning can improve domain-specific visual capabilities of general-purpose MLLMs, we adopt domain-specific fine-tuning as a strategic approach to understand the roles that the MLLM's key architectural components play in modeling visual attributes. A better understanding of the roles of MLLM's components in modeling visual attributes can inform future design choices as well as direct interpretability efforts.

Architecturally, open-source MLLMs comprise two key components: *(i)* a cross-modal projection layer that connects image representations with the LLM, and *(ii)* the LLM that processes the projected image representation and the text tokens; see Figure 1 (left). In the context of the projec-

[1] We use 'natural images' or 'internet images' to refer to common images encountered on social media platforms and the Web and contrast them with domain-specific images.

tion, researchers often consider the projection layer as the unit responsible for aligning features/concepts from the image to the LLM space (Li et al., 2023; Lin et al., 2023; Moon et al., 2023). Consequently, one prevalent fine-tuning strategy to adapt MLLMs for domain-specific visual tasks is to update the projection while keeping the LLM parameters frozen (Moon et al., 2023). Alternatively, the projection and the LLM parameters can be fine-tuned concurrently (Liu et al., 2023b).

In this work, we use domain-specific fine-tuning using the above two strategies to understand the role of the projection and the LLM parameters in acquiring domain-specific image modeling capabilities. We posit that if the projection plays a critical role in acquiring domain-specific image modeling capabilities, the post-projection representation – i.e., the representation of the image transformed by the projection, should be *richer*[2] in domain-specific features. Conversely, if the post-projection representation is not richer in domain-specific features, the domain-specific features are being identified or modeled by the LLM parameters.[3]

Our experiments and analysis with 4 different datasets show that, as expected, both the fine-tuning strategies boost domain-specific closed-set image classification performance of the MLLM. However, none of the strategies lead to extraction of richer domain-specific features by the update in the projection layer; see Figure 1 (right). This indicates that as MLLMs are fine-tuned to classify domain-specific images, the identification of domain-specific image attributes occurs in the LLM parameters, whether frozen or not. More broadly, our results add to the existing evidence that deep neural networks can be inherently multimodal (Goh et al., 2021; Schwettmann et al., 2023), and LLMs could model visual data with minimal assistance from the cross-modal projection.

We first discuss the fine-tuning strategies to improve the domain-specific capabilities of MLLMs (Section 2) and then analyze the role of projection in acquiring the new domain-specific capabilities (Section 3). Finally, we discuss the implications of our work and the future directions (Section 4).

2 Effect of Fine-tuning Projection Layer *versus* the Entire Multimodal LLM

We are interested in exploring two potential fine-tuning strategies that could help an MLLM in gaining domain-specific visual capabilities. The first approach involves simply fine-tuning the vision-to-language projection, e.g., a simple two-layer MLP with ~20M parameters. The second approach involves training the entire MLLM – i.e., the projection layer + the LLM with ~7B parameters. We conduct all our experiments with the LLaVA-1.5 model (Liu et al., 2023b), which uses the LLaMA-2-7B (Touvron et al., 2023) as the LLM backbone, as it is a strong representative of open-source state-of-the-art multimodal LLMs (Ge et al., 2023; Liu et al., 2023a; Yu et al., 2023).

Setting 1: Only fine-tuning the projection layer. LLaVA-1.5 involves pre-training the cross-modal projection layers to align image features with the pre-trained LLM's token embeddings by maximizing the next-token prediction likelihood of the MLLM. Let $\mathbf{X}_a$ denotes the ground-truth output corresponding to the question $\mathbf{X}_q$ regarding the image encoding $\mathbf{X}_v$, which is obtained from the frozen vision-encoder of CLIP (Radford et al., 2021). The projection layer, parameterized by ϕ, is trained to elicit the correct response from the frozen LLM, token-by-token while using the projected image-encoding $\mathbf{H}_v = \phi(\mathbf{X}_v)$, and considering previous tokens of the ground-truth answer. See Figure 2 (Appendix) for a pictorial illustration of the formulation. Since our focus is to perform domain-specific image classification using MLLMs, we consider $\mathbf{X}_a$ = `<label>` for a given image and construct $\mathbf{X}_q$ as:

> Classify this image into one of the following categories relating to `<task>`: `<classes_string>`. Only output a single final classification label and NOTHING ELSE.

For each example, we randomly shuffle the order of classes inside `<classes_string>` to avoid any position bias. We fine-tune the projection layers of the LLaVA-1.5 model for 1 epoch using the default hyper-parameters (Liu et al., 2023b). During inference, we perform zero-shot classification using the same prompt above for the MLLM with the updated projection.

Setting 2: Fine-tuning the MLLM end-to-end. Alternatively, we fine-tune all the MLLM parameters, i.e., the projection layers and the LLM parameters concurrently by maximizing the next token-

[2] We use domain-specific richness to indicate the "expressive power" of the representations (Bengio et al., 2012) towards the domain-specific task.

[3] Project webpage: `https://claws-lab.github.io/projection-in-MLLMs/`

Models/Variants	Agriculture		Textures		Dermatology		Humanitarian	
	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.
Random (Uniform)	0.0309	0.0339	0.0214	0.0218	0.0451	0.0483	0.2425	0.2664
CLIP (Zero-shot; LLaVA-1.5's vision encoder)	0.4165	0.4492	0.4582	0.4984	0.1783	0.2401	0.4139	0.4718
LLaVA-1.5 (Zero-shot)	0.1064	0.1255	0.1882	0.2138	0.0658	0.0672	0.5169	0.5678
LLaVA-1.5 (FT-Proj with labels)	0.2221	0.2478	0.4505	0.4654	0.2932	0.3403	0.6227	0.7151
LLaVA-1.5 (FT-E2E with labels)	0.5984	0.6525	0.7446	0.7496	0.4947	0.5464	0.7950	0.8554

Table 1: **Performance on domain-specific image classification datasets.** Fine-tuning LLaVA-1.5 end-to-end leads to the best domain-specific performance, while only fine-tuning the projection leads to a notable gain over LLaVA's zero-shot capabilities across all the datasets. It is worth noting that CLIP's zero-shot performance, which is the pre-projection image representation that LLaVA uses, is notably better than LLaVA's zero-shot performance. All the values are averaged over 5 experimental runs with different random seeds; the σ is $< 1\%$ for all values.

prediction likelihood of the MLLM. In other words, we update both ϕ and θ, where θ denotes the LLM paramters. We use the same strategy to construct $\mathbf{X}_a$ and $\mathbf{X}_q$ as in the previous setting. Again, we fine-tune the LLaVA-1.5 model for 1 epoch using the default hyper-parameters. Similar to the above setting, after training the MLLM, we perform zero-shot domain-specific image classification using the $\mathbf{X}_q$ constructed above.

We fine-tune the MLLM using these 2 strategies for each of the 4 datasets from different domains.

Image datasets. The 4 image classification datasets correspond to the following tasks: leaf disease classification, visual texture detection, skin disease identification, and humanitarian category classification. Figure 3 (Appendix) provides an illustration of the datasets under consideration.

(i) Agriculture: To enable scalable and early plant disease detection, Singh et al. (2020) curated PlantDoc. The dataset comprises 2,598 images categorized into 17 classes of leaf diseases.

(ii) Textures: With an aim to evaluate whether visual models can identify human-centric attributes like texture beyond detecting or describing objects/scenes, Cimpoi et al. (2014) curated 5,640 images categorized into 47 texture-related classes (like polka-dotted, wrinkled, and honeycombed).

(iii) Dermatology: We consider the DermNet dataset (Rimi et al., 2020), which comprises 19,561 images categorized into 23 types of skin diseases like Acne, Melanoma, Seborrheic Keratoses, etc.

(iv) Humanitarian: To aid development of computational methods that can help humanitarian organizations process images posted on social platforms during crises, Alam et al. (2018) and Ofli et al. (2020) curated the CrisisMMD dataset, which comprises 10,461 images categorized into 4 different categories. This dataset comprises images that are the closest to natural/internet images.

Domain-specific classification performance. Table 1 shows the image classification performance (macro-averaged F_1 scores and accuracy) of the MLLMs under various settings. For reference, we include zero-shot classification performance of CLIP[4], which is the visual encoder of the LLaVA-1.5 model (see Appendix A.1 for details). First, it is worth noting that the zero-shot performance of the original LLaVA-1.5 model is notably worse than CLIP's zero-shot performance. This indicates that while domain-specific image attributes are present in the pre-projection image embeddings that are obtained from a frozen vision encoder (i.e., $\mathbf{X}_v$), they are not being used by the MLLM parameters. This can be attributed to the corpus used to train MLLMs like LLaVA, which comprises natural images. Second, clearly, the results show that finetuning indeed improves performance on domain-specific classification, with significant improvements made when fine-tuning the entire MLLM ('FT-E2E') as opposed to only the projection layer ('FT-Proj'). The greater effectiveness of the FT-E2E can be attributed to greater representational space (~ $7B$) over FT-Proj (~ $20M$). With these observations, next, we focus on investigating the role of projection in capturing domain-specific image attributes.

3 Role of Projection in Learning Domain-Specific Image Attributes

Following up on results in Table 1, we ask: *does the projection learn to model the domain-specific image attributes on fine-tuning the MLLM?*

[4] https://huggingface.co/openai/clip-vit-large-patch14-336 (Wolf et al., 2019)

Task	**Setting**	**Post-proj MLP** (LLaVA-1.5; F_1)	**MLLM** (LLaVA-1.5; F_1)
Agriculture	Original	0.5701 (———)	0.1064 (———-)
	FT-Proj	0.4134 (-27.49%)	0.2221 (+108.74%)
	FT-E2E	0.5346 (-06.22%)	0.5984 (+462.41%)
Textures	Original	0.6401 (———)	0.1882 (———-)
	FT-Proj	0.4736 (-26.01%)	0.4505 (+139.37%)
	FT-E2E	0.6212 (-02.95%)	0.7446 (+295.64%)
Dermatology	Original	0.3105 (———)	0.0658 (———-)
	FT-Proj	0.2182 (-29.72%)	0.2932 (+345.59%)
	FT-E2E	0.2525 (-18.67%)	0.4947 (+651.82%)
Humanitarian	Original	0.7498 (———)	0.5169 (———-)
	FT-Proj	0.6025 (-19.64%)	0.6227 (+020.47%)
	FT-E2E	0.7238 (-03.46%)	0.7950 (+053.80%)

Table 2: **Estimating the domain-specific richness of the post-projection image representation using an independent MLP.** Compared to the original LLaVA-1.5 model, both fine-tuning strategies lead to worsened domain-specific richness of the post-projection image representation (second-last column), while the MLLM performance (last column) improves consistently. This implies that the domain-specific attributes are identified in the LLM, even when the LLM parameters are kept frozen as the projection is updated (i.e., 'FT-Proj').

Estimating post-projection richness. To answer the above question, we develop a reliable-yet-simple way to estimate domain-specific richness of the projected image representation, i.e., the post-projection representation, denoted by $\mathbf{H}_v = \phi(\mathbf{X}_v)$. We do this by training an independent multilayer perceptron (MLP) to perform the image classification task using $\mathbf{H}_v$ as the image representation. This classifier helps estimate the extent of domain-specific information (or expressive power (Bengio et al., 2012)) that can be extracted from the input, in this case the post-projection image representation $\mathbf{H}_v$. In other words, a better classification performance by this MLP will denote relative domain-specific richness of the post-projection embeddings used for training, and vice versa. We train one MLP each using the post-projection representation $\mathbf{H}_v$ obtained from the following three settings: (i) original LLaVA-1.5, (ii) LLaVA-1.5 with fine-tuned projection, and (ii) LLaVA-1.5 with end-to-end fine-tuning, while keeping the architecture of the MLP the same for consistent comparison. We provide the additional details, including architecture and training hyper-parameters, in Appendix A.2.

Comparing domain-specific richness of post-projection representation across different settings. Table 2 shows: *(a)* the domain-specific richness of post-projection representation $\mathbf{H_v}$ ('Post-proj MLP'), and *(b)* the corresponding MLLM performance ('MLLM'), across the three settings mentioned above (i.e., 'Original', 'FT-Proj', and 'FT-E2E'). We report the macro-averaged F_1 score on the test set of the respective dataset for both (a) and (b). There are two key trends in Table 2: *first*, when the 'Original' LLaVA-1.5 model's projection layer is fine-tuned ('FT-Proj'), the domain-specific richness of the post-projection representation diminishes, while a boost in the MLLM performance is observed. Similarly, *second*, with end-to-end fine-tuning of LLaVA-1.5 ('FT-E2E'), the domain-specific richness of the post-projection representation worsens while the MLLM performance boosts notably. These two trends are consistent across all the datasets considered in our study.

Domain-specific attributes are identified within the LLM. The two trends observed above reinforce the idea that as the MLLM gains previously-absent domain-specific image classification abilities via fine-tuning, the contribution of the projection layer in identifying relevant image attributes declines. Let us consider the two fine-tuning settings separately. In the first setting, the projection layer undergoes updates to assist the *frozen* LLM in more accurate label prediction, and yet captures lesser domain-specific image attributes. This indicates that the updates in projection layer merely facilitate better use of frozen LLM parameters for the domain-specific task and do not necessarily involve mapping image attributes to the frozen LLM space. In the second setting as well, when both the LLM parameters and projection layer undergo updates concurrently, the projection layer captures lesser domain-specific attributes, which indicates that the updates in the LLM parameters are predominantly responsible for the acquired domain-specific image classification capabilities. In sum, our results indicate that the modeling of domain-specific image attributes in MLLMs is done by the LLM parameters, whether they are kept frozen or undergo updates.

4 Discussion and Implications

Existing literature on interpretability of neural networks has discussed the notion of "multimodal neurons" – neurons that trigger in response to particular concepts spanning disparate modalities (Goh et al., 2021; Schwettmann et al., 2023; Pan et al., 2023). For instance, Goh et al. (2021) demonstrate that in the CLIP model, a single neuron could respond to the photographs, drawings, or images that relate to, let's say 'spiderman,' even though the input image may differ in terms of low-level visual attributes like color, edges, and corners. Similarly, Schwettmann et al. (2023) show that a specific neurons within a *frozen* text-only Transformer are responsible for detecting visual concepts, let's say like 'horses,' in the input images that are projected to align with the text-only transformer. Our study adds to this literature by showing that even the acquired abilities to detect visual attributes in an MLLM are reliant on the LLM parameters. Notably, when the LLM parameters are frozen, the cross-modal projection layer adapts to facilitate detection of visual attibutes in the LLM without extracting domain-specific attributes. In other words, when the LLM is frozen and the projection is fine-tuned, the projection parameters are updated to leverage the pre-existing domain-specific knowledge in the LLM parameters. In the future, we aim to interpret the layer- & neuron-level contributions in LLMs towards acquired multimodal reasoning.

5 Limitations and Broader Perspective

Limitations and future work: Our current work focuses on a representative cross-modal projection scheme (multilayer perceptron) in a state-of-the-art MLLM (LLaVA-1.5). Other open-source MLLMs have considered other projection schemes like a trainable linear layer (LLaVa-1; Liu et al. (2023c)), gated cross-attention (Flamingo; Alayrac et al. (2022)), and Q-Former (InstructBLIP; Dai et al. (2023)). Future work could extend the current study to other projection schemes and models. Beyond the adopted strategy of estimating the post-projection richness of image representations using an independent classifier, future work could also probe the MLLM using concept bottleneck methods (Koh et al., 2020), or analyze mutual information between representations (Bachman et al., 2019). Finally, while outside the scope of the current work, a holistic evaluation of the MLLM should focus on domain-specific capabilities as well as the general purpose capabilities.

Broader social impact: The authors do not foresee any negative social impacts of this specific work. However, we acknowledge that existing LLMs and MLLMs demonstrate different forms of biases (Wan et al., 2023; Nwatu et al., 2023) that could be inherited in domain-specific variants. In line with the ongoing effort towards mitigating social biases in deep neural networks, future efforts that aim to interface modality-specific reasoning with LLMs, should consider the additional biases that LLMs may introduce on top of the modality-specific networks.

Datasets and code: The datasets used in this study are publicly available and were curated by previous research. We abide by their terms of use. We release the code for our experiments to aid reproducibility and enable future research on this topic: `https://github.com/claws-lab/projection-in-MLLMs`

6 Acknowledgements

This research/material is based upon work supported in part by NSF grants CNS-2154118, ITE-2137724, ITE-2230692, CNS2239879, Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00112290102 (subcontract No. PO70745), CDC, and funding from Microsoft. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the position or policy of DARPA, DoD, SRI International, CDC, NSF, and no official endorsement should be inferred. Gaurav is partially supported by the JP Morgan AI Research PhD Fellowship and the Snap Research Fellowship. We thank the members of the CLAWS Lab for their helpful feedback.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Abien Fred Agarap. 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Landing AI. 2024. Introducing domain-specific large vision models. `https://landing.ai/blog/introducing-domain-specific-large\-vision-models/`. Accessed: 2024-02-14.

Firoj Alam, Ferda Ofli, and Muhammad Imran. 2018. Crisismmd: Multimodal twitter datasets from natural disasters. In *Proceedings of the 12th International AAAI Conference on Web and Social Media (ICWSM)*.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Onder Anilturk, Edwin Lumanauw, James Bird, Juan Olloniego, Dillon Laird, Juan Camilo Fernandez, and Quinn Killough. 2023. Automatic defect classification (adc) solution using data-centric artificial intelligence (ai) for outgoing quality inspections in the semiconductor industry. In *Metrology, Inspection, and Process Control XXXVII*, volume 12496, pages 830–836. SPIE.

Philip Bachman, R Devon Hjelm, and William Buchwalter. 2019. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32.

Yoshua Bengio, Aaron C Courville, and Pascal Vincent. 2012. Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR, abs/1206.5538*, 1(2665):2012.

M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. 2014. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Albert Li, Pascale Fung, and Steven C. H. Hoi. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. *ArXiv*, abs/2305.06500.

Konstantinos P Ferentinos. 2018. Deep learning models for plant disease detection and diagnosis. *Computers and electronics in agriculture*, 145:311–318.

Wentao Ge, Shunian Chen, Guiming Chen, Junying Chen, Zhihong Chen, Shuo Yan, Chenghao Zhu, Ziyue Lin, Wenya Xie, Xidong Wang, et al. 2023. Mllm-bench, evaluating multi-modal llms using gpt-4v. *arXiv preprint arXiv:2311.13951*.

Gabriel Goh, Nick Cammarata, Chelsea Voss, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. 2021. Multimodal neurons in artificial neural networks. *Distill*, 6(3):e30.

Maria Kaselimi, Athanasios Voulodimos, Ioannis Daskalopoulos, Nikolaos Doulamis, and Anastasios Doulamis. 2022. A vision transformer model for convolution-free multilabel classification of satellite imagery in deforestation monitoring. *IEEE Transactions on Neural Networks and Learning Systems*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. 2020. Concept bottleneck models. In *International conference on machine learning*, pages 5338–5348. PMLR.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. 2023. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *arXiv preprint arXiv:2306.00890*.

Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. 2023. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*.

Fuxiao Liu, Tianrui Guan, Zongxia Li, Lichang Chen, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. 2023a. Hallusionbench: You see what you think? or you think what you see? an image-context reasoning benchmark challenging for gpt-4v (ision), llava-1.5, and other multi-modality models. *arXiv preprint arXiv:2310.14566*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023b. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023c. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. 2018. Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics*, 19(6):1236–1246.

Seungwhan Moon, Andrea Madotto, Zhaojiang Lin, Tushar Nagarajan, Matt Smith, Shashank Jain, Chun-Fu Yeh, Prakash Murugesan, Peyman Heidari, Yue Liu, et al. 2023. Anymal: An efficient and scalable any-modality augmented language model. *arXiv preprint arXiv:2309.16058*.

Joan Nwatu, Oana Ignat, and Rada Mihalcea. 2023. Bridging the digital divide: Performance variation across socio-economic factors in vision-language models. *arXiv preprint arXiv:2311.05746*.

Ferda Ofli, Firoj Alam, and Muhammad Imran. 2020. Analysis of social media data using multimodal deep learning for disaster response. In *17th International Conference on Information Systems for Crisis Response and Management*. ISCRAM, ISCRAM.

Haowen Pan, Yixin Cao, Xiaozhi Wang, and Xun Yang. 2023. Finding and editing multi-modal neurons in pre-trained transformer. *arXiv preprint arXiv:2311.07470*.

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

Tanzina Afroz Rimi, Nishat Sultana, and Md Ferdouse Ahmed Foysal. 2020. Derm-nn: skin diseases detection using convolutional neural network. In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 1205–1209. IEEE.

Sarah Schwettmann, Neil Chowdhury, Samuel Klein, David Bau, and Antonio Torralba. 2023. Multimodal neurons in pretrained text-only transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2862–2867.

Davinder Singh, Naman Jain, Pranjali Jain, Pratik Kayal, Sudhakar Kumawat, and Nipun Batra. 2020. Plantdoc: A dataset for visual plant disease detection. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, pages 249–253.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yixin Wan, George Pu, Jiao Sun, Aparna Garimella, Kai-Wei Chang, and Nanyun Peng. 2023. "kelly is a warm person, joseph is a role model": Gender biases in llm-generated reference letters. *arXiv preprint arXiv:2310.09219*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2023. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*.

A Appendix

A.1 Zero-Shot Classification Using CLIP

We perform zero-shot classification using the CLIP model (`clip-vit-large-patch14-336`;), which is the same as the vision encoder used for obtaining pre-projection representation of the input image (i.e., $\mathbf{X}_v$) by the LLaVA-1.5 model. The CLIP model embeds both image and text data into a common space using a contrastive learning objective. We use the pre-trained model to compute the cosine similarity between the image representations and the representation of the dataset-specific label strings obtained from the textual backbone of CLIP. Following this, we consider the most similar label string to be the predicted label for the given image, and compute classification metrics on the test set to quantify CLIP's zero-shot performance.

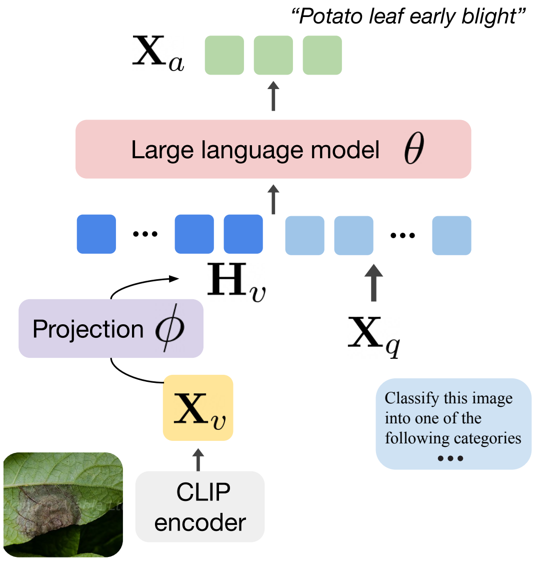
"Potato leaf early blight"
$\mathbf{X}_a$
Large language model θ
$\mathbf{H}_v$
$\mathbf{X}_q$
Projection ϕ
$\mathbf{X}_v$
CLIP encoder
Classify this image into one of the following categories
•••

Figure 2: **Architecture of the MLLM** considered in this study. ϕ and θ denote tunable parameters of the projection and the large language model, respectively.

A.2 Multilayer Perceptron for Estimating Post-Projection Richness

We train a multilayer perceptron for estimating the domain-specific richness of the post-projection image representation (i.e., $\mathbf{H}_v$). The MLP takes the tokens corresponding to the image as input and learns to perform the classification task using the examples from the standard train set. Architecturally, the MLP comprises a token-level average pooling step to obtain the image representation, followed by subsequent layers, and eventually the output layer of size equivalent to the number of classes in the dataset. We use ReLU activation (Agarap, 2018) to induce non-linearity. We keep the architecture of this MLP fixed across all the settings to control for the number of learnable parameters and the representational power of the neural network, therefore allowing us to estimate the richness of the input embeddings with respect to the target task. Each model is trained with a batch size of 128. We use Adam optimizer (Kingma and Ba, 2014) with a learning rate initialized at 10^{-4} and adopt early stopping based on the loss values to avoid overfitting. As a sanity check, we note that an MLP trained using our setup on the post-projection embeddings obtained from the original LLaVA-1.5 model for the HUMANITARIAN task (a natural images dataset), achieves close to the state-

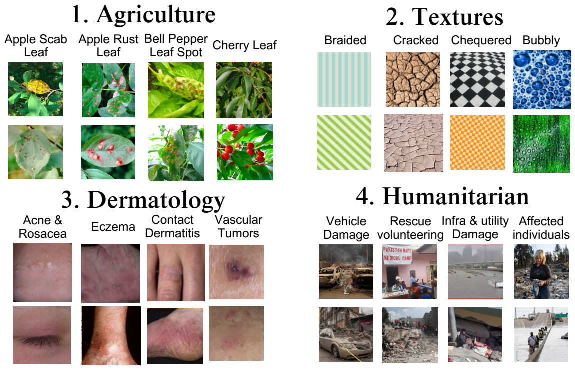

Figure 3: **Illustration of the 4 domain-specific image classification datasets** used in this study. The datasets are from diverse domains; for brevity we only show some of the representative labels from each of the datasets. Images best viewed with zoom.

Task	F_1 score	Acc.
Agriculture	0.6991	0.7118
Textures	0.7644	0.7638
Dermatology	0.6046	0.6492
Humanitarian	0.7506	0.8238

Table 3: **Classification performance of MLP-based image-only classifiers.** A simple MLP performs better on 3 out of 4 tasks than the fine-tuned multimodal LLM; see Table 1 for MLLM results.

of-the-art performance reported on this task (Alam et al., 2018). This indicates that our setup enables a reliable estimate of the richness/expressive power of the post-projection representations.

A.3 Performance of Image-only Models

As reference to the performance of MLLM's domain-specific capabilities (before and after fine-tuning), we include the performance of simple image-only classification models. We use the 1024-dimensional image embeddings obtained from a pre-trained CLIP model (`clip-vit-large-patch14-336`) and train a multilayer perceptron with layers of size (1024 (input layer), 2000, 3600, 1024, 600, 256, # of classes (output layer)). We use the same design choices as used for training the MLPs described in Sec. A.2, and evaluate the models on respective test sets of the dataset. The results are presented in Table 3. Although it is not the primary focus of this work, it is interesting to note that for the domain-specific tasks – i.e., all the 3 tasks except HUMANITARIAN the MLP (with ~ $20M$ parameters) performs better than the fine-tuned MLLM (with ~ $7B$ parameters). Both the model use CLIP embeddings as input representation of the image and are fine-tuned with the same amount of labeled data.

A.4 Compute Resources

All the experiments discussed in this study were conducted using two NVIDIA A100 GPUs (80 GB). Each fine-tuning run of the MLLM took about 1 hour requiring both the GPUs, with additional time for inference; multiple inference runs could be carried over a single GPU. The training and evaluation of the MLPs took less than 20 minutes each. Each run of zero-shot evaluation of CLIP was done on a single GPU in less than 15 minutes.