Benchmarking Multimodal Large Language Models for Image Classification

Tuan Nguyen
Neural Networks and Deep Learning
University of Colorado Boulder
tuan.nguyen@colorado.edu

Abstract

Recent Multimodal Large Language Models (MLLMs) show strong potential for solving vision-language tasks, particularly image classification. Several studies have demonstrated that zero-shot classification approaches can outperform traditional methods. This project benchmarks the image classification performance of both proprietary and open-source MLLMs, including Gemini Flash, GPT-4o, and LLaVA-70B, using the Pets and Caltech-101 datasets. We evaluate different prompting techniques, comparing the effects of zero-shot prompting versus Chain of Thought prompting. Our results highlight the contribution of each component in zero-shot image classification algorithms and reveal performance differences across models under both default and advanced prompting strategies.

1 Introduction

The rise of multimodal applications and the growing demand for intelligent visual understanding have sparked significant interest in leveraging advanced models for vision-related tasks. As real-world scenarios increasingly involve interpreting both images and text, Multimodal Large Language Models (MLLMs) such as GPT-4V, Gemini 2.5, and LLaVA are becoming essential tools for researchers and developers. These models promise a unified framework for handling complex vision-language tasks, with potential benefits including reduced need for task-specific architectures, improved zero-shot generalization, and streamlined deployment across diverse domains.

Despite their promise, current research often evaluates MLLMs in isolation or focuses primarily on text generation tasks, leaving a gap in understanding their effectiveness in structured tasks like image classification. While CLIP and similar models have shown success in vision language alignment, there is limited experimental validation of how well state-of-the-art MLLMs perform on classification tasks particularly in zero-shot and few-shot scenarios. Moreover, the impact of different prompting strategies on classification performance remains underexplored, especially when comparing open-source models with proprietary counterparts.

Recent studies have introduced innovative methods to evaluate the image classification capabilities of MLLMs, often surpassing traditional solutions. Rather than training conventional classifiers, these approaches leverage MLLMs to extract enriched contextual information directly from images. This enriched information, along with the original visual content, is encoded, typically using CLIP-based architectures, and compared with candidate label representations for classification.

However, a comparison between state-of-the-art proprietary models and open-source MLLMs remains unexplored. To address this gap, this project systematically evaluates a range of leading MLLMs on image classification tasks using various prompting-based strategies. Pets and Caltech-101 were selected as the datasets due to their manageable number of classes, which are well suited for prompt-based classification. In contrast, ImageNet with its 1,000 classes poses challenges for open-source MLLMs, as passing all class labels into a single prompt often exceeds model input limits or leads to degraded performance.

This project compares the image classification performance of three MLLMs: Gemini Flash 2.0 [12], GPT-4o [7], and LLaVA-70B [5]. It further investigates the contribution of individual components within a fusion-based model. Additionally, the effectiveness of Zero-Shot and Chain of Thought [13] prompting techniques is evaluated for each model.

2 Related Work

Image classification has long been a core task in computer vision, traditionally addressed using convolutional neural networks (CNNs) such as AlexNet [4], VGG [10],

ResNet [3], and EfficientNet [11]. These models are trained in a supervised fashion to map input images directly to discrete class labels by learning hierarchical feature representations. Subsequent advances in vision transformers [2] (ViTs) further improved performance by leveraging self-attention mechanisms for global context modeling. In parallel, the emergence of contrastive learning methods, most notably CLIP [9], reformulated classification as a matching problem between image and text embeddings, enabling zero-shot generalization across open-vocabulary labels. Building on this foundation, recent work explores the use of multimodal large language models (MLLMs) that combine pretrained vision encoders with autoregressive language models (e.g., GPT, LLaMA). Models such as Gemini [12], GPT-4 [7], and LLaVA [5] enable image classification by prompting the model with natural language queries, often incorporating candidate labels or descriptive templates. This prompt-based paradigm introduces flexibility and cross-domain transfer capabilities but also raises new challenges in prompt design, label grounding, and evaluation consistency.

A growing body of research has investigated the use of multimodal large language models (MLLMs) for image classification tasks, offering novel prompt-based alternatives to traditional vision pipelines. Yijiashun et al. (2024)[8] proposed a multi-agent system integrating LLaVA-34B, GPT-4o, and Gemini 1.5 Pro, utilizing Zero-Shot and Chain of Thought prompting. Their method achieved superior performance in marine mammal classification compared to both traditional machine learning techniques (e.g., KNN, SVM) and deep learning baselines (e.g., VGG, ResNet-50, CLIP).

In a similar vein, Zhaoheng et al. (2024) introduced LLaMP, an adaptive prompt learning framework that formulates classification as an image-text matching task. Their approach enables large language models to generate class-specific prompt vectors, which are then used to guide the CLIP text encoder, significantly enhancing few-shot classification performance [16]. Complementarily, Abdelrahman et al. (2025) proposed a modality fusion technique in which MLLMs generate comprehensive textual embeddings from images, which are then combined with visual features for zero-shot classification via a linear classifier. Unlike methods that require dataset-specific prompt engineering, their system employs a unified prompt design across datasets, yielding consistent performance improvements over traditional baselines [1].

Other notable efforts include Wu et al. (2024), who leveraged GPT-4 for zero-shot visual recognition, demonstrating that rich, descriptive language prompts can substantially boost accuracy and reporting an average top-1 gain of 7% across various benchmarks [14]. On the evaluation front, Liu et al. (2024) reformulated standard image classification datasets into a multiple-choice question (MCQ) format to systematically benchmark MLLMs. Their findings suggest that, in fine-grained settings, CLIP-style models still outperform most MLLMs [6]. Yuhui et al. (2024) observed that proprietary and open-source vision-language models, despite their scale and use of CLIP-based encoders, often lag behind CLIP in conventional classification tasks like ImageNet [15].

Algorithm 1 Abdelrahman's approach to perform zero-shot image classification [1].

1. **Input:** Image $\mathbf{X}$, class labels $\{l_i\}_{i=1}^{m}$, class label feature matrix $\mathbf{M}$, multimodal LLM g, cross-modal encoders f_i & f_t, initial class prediction prompt p_c, image description prompt p_d
2. $\widetilde{\mathbf{X}}_{if} = f_i(\mathbf{X})$ {Image feature}
3. $\mathbf{X}_{if} = \widetilde{\mathbf{X}}_{if} / \|\widetilde{\mathbf{X}}_{if}\|$ {Vector normalization}
4. $\widetilde{\mathbf{X}}_{df} = (f_t \circ g)(\mathbf{X}, p_d)$ {Image description feature}
5. $\mathbf{X}_{df} = \widetilde{\mathbf{X}}_{df} / \|\widetilde{\mathbf{X}}_{df}\|$ {Vector normalization}
6. $\widetilde{\mathbf{X}}_{pf} = (f_t \circ g)(\mathbf{X}, p_c)$ {Initial prediction feature}
7. $\mathbf{X}_{pf} = \widetilde{\mathbf{X}}_{pf} / \|\widetilde{\mathbf{X}}_{pf}\|$ {Vector normalization}
8. $\widetilde{\mathbf{X}}_{q} = \mathbf{X}_{if} + \mathbf{X}_{df} + \mathbf{X}_{pf}$ {Fused feature}
9. $\mathbf{X}_{q} = \widetilde{\mathbf{X}}_{q} / \|\widetilde{\mathbf{X}}_{q}\|$ {Vector normalization}
10. $\mathbf{W} = \mathbf{X}_{q}^{\top} \mathbf{M}$ {Similarity scores}
11. $x \leftarrow \arg\max(\mathbf{W})$ {Predicted class index}
12. **Output:** Predicted class label l_x of input image

3 Methods

Instead of developing a novel zero-shot classification algorithm, our approach extends the experimental framework proposed by **Abdelrahman et al.** In their method, directly predicting image classes using MLLMs often results in semantically similar but incorrect labels. To address this, the approach employs a CLIP-based model to encode both visual and textual content. Additional descriptive information about the image is first generated by the MLLM and then passed through the encoder to enhance classification performance.

3.1 Datasets

We use the Oxford-IIIT Pets and Caltech-101 datasets for this study. The Pets dataset contains 37 categories, each with approximately 200 images, while Caltech-101 consists of 101 object categories, with each class containing between 40 and 800 images. These datasets offer a manageable number of classes for prompt techniques, where additional information is generated from class names in subsequent steps. We initially considered

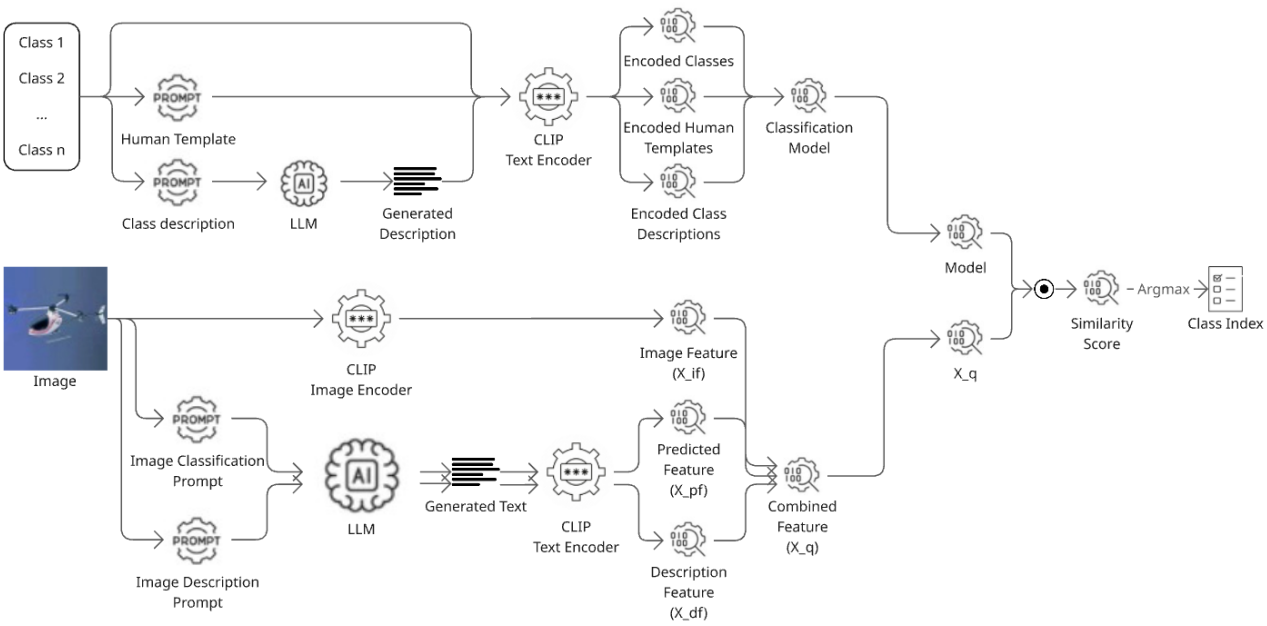

Figure 1: Classification steps.

ImageNet, which includes 1,000 classes, for evaluation. It was ultimately excluded from this study due to input token limitations in open-source models.

3.2 Classification

When using MLLMs to directly predict image labels, the output often consists of semantically similar but incorrect labels, leading to a decline in classification performance. To address this limitation, Abdelrahman et al. introduced a novel approach that encodes both visual and textual features and performs classification by matching image embeddings with a predefined set of label embeddings. This ensures that predictions are always restricted to the true label set, improving accuracy and consistency.

Label Model

The original class labels are first processed and encoded using a text encoder. For each label, three distinct textual representations are generated: (1) the original label itself, (2) a human-designed template such as *'This is an image of {label}'*, and (3) a descriptive sentence generated by an LLM that captures the characteristics of the label. These representations are individually encoded, and their embeddings are averaged to produce a final feature vector for each label. The resulting vectors are then stacked to construct the final label representation model.

Image Features

To enrich the input information, we do not use the image alone. Instead, an MLLM is used to generate both an initial prediction and a detailed description of the image. Specific prompts are designed for each task.

The prompt for generating the image description is: *What do you see? Describe any object precisely, including its type or class.*
The prompt for generating the initial prediction is: *You are given an image and a list of class labels. Classify the image given the class labels. Answer using a single word if possible. Here are the class labels: {classes}.*

Since this prompt includes all class labels, it increases the number of input tokens. The responses from the MLLM are text, which are then encoded using a CLIP-based model. At the same time, the original image is also encoded using the same model.

The following features are created:

1. (X_{if}): The image embedding produced by the CLIP-based model, representing how much the encoder model can capture the understanding of the input image.
2. (X_{df}): The embedding of the textual image description generated by the MLLM and encoded using the CLIP-based model, capturing additional semantic information about the image.
3. (X_{pf}): The embedding of the textual image prediction generated by the MLLM, also encoded using the CLIP-based model, reflecting the model's initial classification hypothesis.

Finally, the three feature vectors (image feature (X_{if}), image description feature (X_{df}), and initial prediction feature (X_{pf})) are normalized and averaged to create the final image feature representation (X_q).

Prediction

At the prediction stage, the similarity scores between the final image feature representation and each label representation are computed. The label corresponding to the highest similarity score, determined using the argmax function, is selected as the final prediction.

Evaluation

The evaluation metric used in this study is accuracy, defined as:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

A higher accuracy value indicates better classification performance. The evaluation is conducted on the 185 testing images of Pets dataset and 202 testing images of Caltech-101.

4 Experiments

Experiment settings: We use API services provided by Gemini and OpenAI to access proprietary models. For open-source models, we employ LLaVA-70B for image-based question answering and Llama-3-13B for text generation. CLIP-VIT-L14 is chosen as both the textual and visual encoders.

4.1 Experiment 1: Contribution of each component in classification

In image features generating step, the final image feature representation (X_q) is calculated by fusing the three components including image feature (X_{if}), image description feature (X_{df}), and initial prediction feature (X_{pf})). This experiment determines which is the most significant component by individually predict the label using each component.

In stead of calculating the similarity score using (X_q), this step is repeated for each others (X_{if}), (X_{df}), (X_{pf})

Table 1 presents the classification accuracies of individual feature components and their combination across the Pets and Caltech-101 datasets. On the Pets dataset, Gemini-Flash achieves the highest accuracy with X_{pf} (89.19%), while the combined feature X_q follows closely at 88.64%. For Caltech-101, X_q with Gemini-Flash reaches 92.08%, nearly matching the best individual score (93.56%) from X_{pf}. Among the individual components, X_{pf} (prediction-based feature) consistently delivers the highest accuracy, particularly with proprietary models like GPT-4o and Gemini-Flash. This suggests that MLLM-generated predictions are highly informative when token limits are not a constraint.

The image feature X_{if} also performs well, especially on Caltech-101, indicating the strong baseline capacity of CLIP-based image encoders. In contrast, X_{df} (description-based feature) lags behind, especially on Pets, likely due to the limited expressiveness or relevance of generated descriptions for fine-grained categories.

The results also suggest that the CLIP-based model effectively captures visual information even without textual context. However, the quality of descriptions and predictions varies depending on the specific MLLM used. When comparing between two datasets, the MLLMs captures better understanding of the Caltech dataset than the Pets dataset.

4.2 Experiment 2: Performance by models

Figure 2 compares the performance of models on image classification using Pets dataset. Gemini-Flash achieves the highest accuracy at 88.64%, followed by GPT-4o at 84.32%, and LLaVA-70B at a significantly lower 32.97%. This indicates that proprietary models, particularly Gemini-Flash and GPT-4o, are far more effective at handling fine-grained classification tasks such as pet breeds, while the open-source LLaVA-70B struggles in this domain.

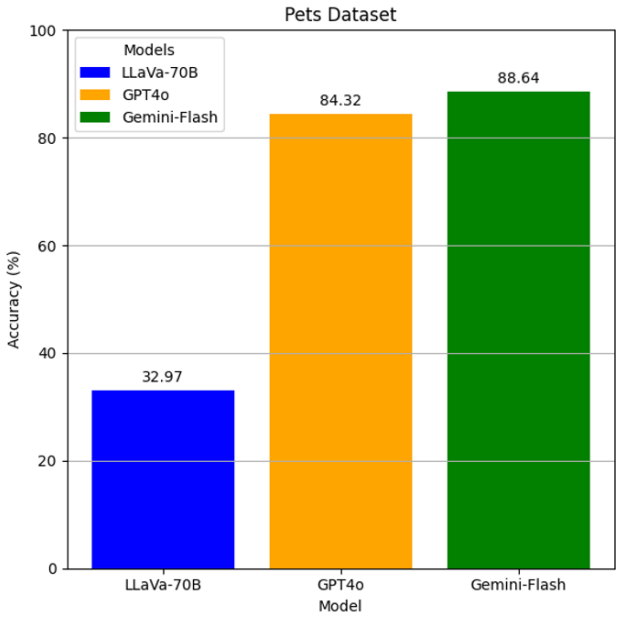

Figure 2: The accuracy of image classification task on Pets dataset by models.

Figure 3 shows the performance of models on image classification using Caltech-101 dataset. Gemini-Flash again leads with 92.08%, showing strong performance in general object classification. LLaVA-70B follows with a solid 79.70%, outperforming GPT-4o, which achieves

Dataset	Pets			Caltech-101		
Model	LLaVa-70B	GPT4o	Gemini-Flash	LLaVa-70B	GPT4o	Gemini-Flash
Image Feature X_{if}	73.51	74.59	83.78	92.08	91.09	90.09
Description Feature X_{df}	21.08	49.73	59.46	74.25	80.69	84.15
Prediction Feature X_{pf}	32.43	83.78	89.19	74.75	63.36	93.56
Combined Feature X_q	32.97	84.32	88.64	79.70	75.74	92.08

Table 1: Classification accuracies of components on Pets and Caltech-101 datasets.

Model	LLaVA		GPT4o		Gemini-Flash	
Prompt	Default	CoT	Default	CoT	Default	CoT
Description Feature X_{df}	74.25	68.31	80.69	77.22	84.15	84.15
Prediction Feature X_{pf}	74.75	71.78	63.36	63.36	93.56	94.55
Combined Feature X_q	79.70	75.24	75.74	76.73	92.87	92.57

Table 2: Benchmark results comparing between different prompt techniques.

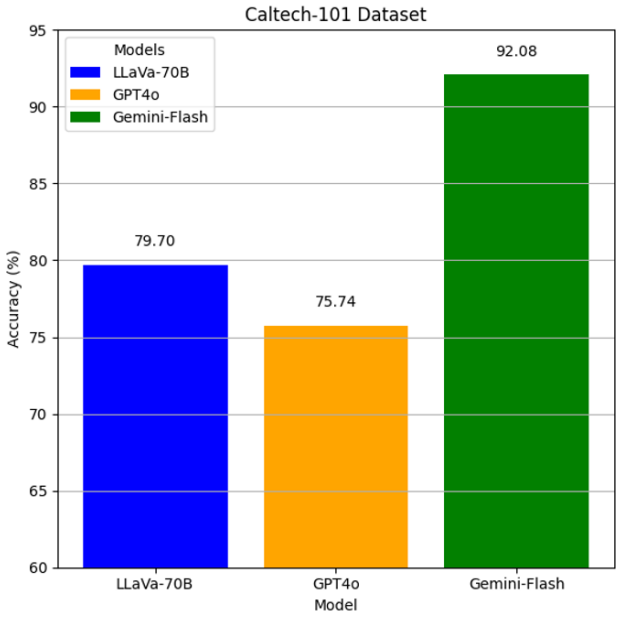

Figure 3: The accuracy of image classification task on Caltech-101 dataset by models.

75.74%. Unlike in the Pets dataset, LLaVA-70B performs competitively here, possibly due to the broader and more distinguishable category types in Caltech-101.

Overall, Gemini-Flash consistently outperforms both GPT-4o and LLaVA-70B, demonstrating its strength in both fine-grained and general image classification. GPT-4o performs well on Pets but less so on Caltech-101, whereas LLaVA-70B shows more balanced performance but lags significantly on fine-grained tasks. This highlights the superior robustness and adaptability of proprietary models in prompt-based classification.

4.3 Experiment 3: Performance by prompting techniques

This experiment is designed to evaluate the impact of different prompting strategies during the generation process using multimodal LLMs. Chain of Thought prompting is included due to its marginal improvements reported in prior studies. The prompt used for generating image descriptions is shown below.

- **Zero-shot prompt:** *What do you see? Describe any object precisely, including its type or class.*
- **Chain of Thought (CoT):** *First, identify the main object in the image. Then, describe its visual attributes such as shape, color, and size. Finally, infer the most likely category or class based on these observations.*

The following prompting strategies are designed for image classification tasks. Each prompt instructs the model to classify an image based on a predefined list of class labels.

- **Zero-shot prompt:** *You are given an image and a list of class labels. Classify the image given the class labels. Answer using a single word if possible. Here are the class labels: {classes}*
- **Chain of Thought (CoT):** *You are given an image and a list of class labels. First, describe key visual features of the image. Then, reason which class label best fits the visual characteristics. Answer using a single word if possible. Here are the class labels: {classes}*

The results in Table 2 show a comparison between the zero-shot prompt and CoT prompting techniques across three models.

For the description feature X_{df}, the default prompt slightly outperforms CoT for both LLaVA and GPT-4o, while Gemini-Flash shows no difference between the

two (both yielding 84.15%). This indicates that adding reasoning steps via CoT may not enhance the quality of image descriptions for classification purposes.

In the prediction feature X_{pf}, Gemini-Flash shows a marginal gain from CoT prompting (93.56% to 94.55%), but GPT-4o remains unaffected, and LLaVA shows only a small drop. These changes are within a narrow range and do not point to a consistent trend across models.

Similarly, for the combined feature X_q, which integrates all components, accuracy differences between default and CoT prompts are less than 1–4 percentage points. In some cases, such as with GPT-4o, CoT slightly improves performance (75.74% to 76.73%), while in others, like LLaVA and Gemini, it leads to a small decline or no improvement.

Generally, the performance differences between the two prompting strategies are minimal, suggesting that CoT prompting does not provide a consistent advantage in this classification task.

4.4 Experiment Limitations

While the experiments provided valuable insights, several important questions remain unanswered. Due to computational constraints, the study was limited to only two datasets, each with approximately 200 test images. This restricted scope limits our ability to generalize the findings across more diverse or large-scale benchmarks. For instance, Experiment 1 highlighted the significance of the image-encoded component X_{if}, but did not assess its performance on more complex datasets like ImageNet, which includes 1000 classes and a broader variety of images. Experiment 2 demonstrated that Gemini outperforms other methods. However, it did not explain why the performance gap varies across datasets. Experiment 3 found no significant difference between zero-shot and CoT prompting, yet it did not explore whether more sophisticated or task-specific CoT strategies could yield different outcomes.

To address these limitations, future work should evaluate the approach across a wider range of datasets, spanning from general-purpose to fine-grained tasks for a better understand in generalizability. Additionally, exploring alternative or more advanced prompting techniques, especially evolved versions of CoT prompting, could help uncover deeper reasoning capabilities and further improve performance.

5 Conclusions

In this project, we did the first experiment that demonstrates the performance among the individual components of a zero-shot image classification task using Multimodal Large Language Model. In which, encoded information using CLIP-based model contributes most significantly to the final prediction. In general, the combined feature X_q consistently provides more balanced and robust performance across datasets and models, validating the effectiveness of feature fusion. In term of model's performance, Gemini-Flash achieves the highest accuracy on both Pets and Caltech-101 datasets, followed by GPT-4o and LLaVA-70B. The comparison between zero-shot and Chain of Thought prompting reveals that while CoT occasionally improves performance marginally, the overall impact is minimal, suggesting that default prompting is generally sufficient for classification tasks using MLLMs.

This project is conducted on a small scale, which the fairness and transparency could be emerged when dealing with larger datasets. The benchmarking approach effectively evaluates model performance, but it does not account for the fairness or transparency of the responses generated by the Multimodal Large Language Model. In particular, the prediction decision relies on similarity scores, that could make a failure to capture the true semantic understanding of the input. When the distances between candidate responses are close, the model selects the nearest match without necessarily demonstrating meaningful comprehension. This raises concerns about the interpretability and reliability of the model's outputs.

All source code for this project is available at:
`https://github.com/tuanna712/Zero-shot-MLLM-Image-Classification`

References

[1] Abdelrahman Abdelhamed, Mahmoud Afifi, and Alec Go. *What Do You See? Enhancing Zero-Shot Image Classification with Multimodal Large Language Models*. 2025. arXiv: 2405.15668 [cs.CV]. URL: https://arxiv.org/abs/2405.15668.

[2] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV]. URL: https://arxiv.org/abs/2010.11929.

[3] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV]. URL: https://arxiv.org/abs/1512.03385.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks". In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'12. Lake Tahoe, Nevada: Curran Associates Inc., 2012, pp. 1097–1105.

[5] Haotian Liu et al. *Improved Baselines with Visual Instruction Tuning*. 2024. arXiv: 2310.03744 [cs.CV]. URL: https://arxiv.org/abs/2310.03744.

[6] Huan Liu et al. *Revisiting MLLMs: An In-Depth Analysis of Image Classification Abilities*. 2024. arXiv: 2412.16418 [cs.CV]. URL: https://arxiv.org/abs/2412.16418.

[7] OpenAI. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774.

[8] Yijiashun Qi et al. *Benchmarking Large Language Models for Image Classification of Marine Mammals*. 2024. arXiv: 2410.19848 [cs.CV]. URL: https://arxiv.org/abs/2410.19848.

[9] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV]. URL: https://arxiv.org/abs/2103.00020.

[10] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV]. URL: https://arxiv.org/abs/1409.1556.

[11] Mingxing Tan and Quoc V. Le. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. 2020. arXiv: 1905.11946 [cs.LG]. URL: https://arxiv.org/abs/1905.11946.

[12] Gemini Team. *Gemini: A Family of Highly Capable Multimodal Models*. 2024. arXiv: 2312.11805 [cs.CL]. URL: https://arxiv.org/abs/2312.11805.

[13] Jason Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. 2023. arXiv: 2201.11903 [cs.CL]. URL: https://arxiv.org/abs/2201.11903.

[14] Wenhao Wu et al. *GPT4Vis: What Can GPT-4 Do for Zero-shot Visual Recognition?* 2024. arXiv: 2311.15732 [cs.CV]. URL: https://arxiv.org/abs/2311.15732.

[15] Yuhui Zhang et al. *Why are Visually-Grounded Language Models Bad at Image Classification?* 2024. arXiv: 2405.18415 [cs.CV]. URL: https://arxiv.org/abs/2405.18415.

[16] Zhaoheng Zheng et al. *Large Language Models are Good Prompt Learners for Low-Shot Image Classification*. 2024. arXiv: 2312.04076 [cs.CV]. URL: https://arxiv.org/abs/2312.04076.