Enhancing Fine-Grained Image Classifications via Cascaded Vision Language Models

Canshi Wei
Tencent Inc.
canshiwei@gmail.com

Abstract

Fine-grained image classification, especially in zero-/few-shot scenarios, poses a considerable challenge for vision-language models (VLMs) like CLIP, which often struggle to differentiate between semantically similar classes due to insufficient supervision for fine-grained tasks. On the other hand, Large Vision Language Models (LVLMs) have demonstrated remarkable capabilities in tasks like Visual Question Answering (VQA) but remain underexplored in the context of fine-grained image classification. This paper presents CascadeVLM, a novel framework that harnesses the complementary strengths of both CLIP-like and LVLMs VLMs to tackle these challenges. Using granular knowledge effectively in LVLMs and integrating a cascading approach, CascadeVLM dynamically allocates samples using an entropy threshold, balancing computational efficiency with classification accuracy. Experiments on multiple fine-grained datasets, particularly the Stanford Cars dataset, show that CascadeVLM outperforms existing models, achieving 92% accuracy. Our results highlight the potential of combining VLM and LVLM for robust, efficient and interpretable fine-grained image classification, offering new insights into their synergy.

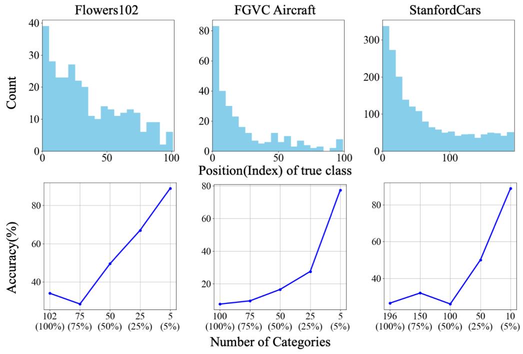


Figure 1: Distribution of true class rankings (top) and accuracies with varying category reductions (bottom) across datasets using QwenVL as the LVLM. The results show positional bias and accuracy improvements with fewer categories.

1 Introduction

The landscape of vision language models (VLMs) has evolved rapidly, with models such as CLIP (Radford et al., 2021) demonstrating remarkable zero- and few-shot classification capabilities (Zhou et al., 2022b). However, despite these advances, fine-grained image classification remains a significant challenge, particularly when distinguishing closely related subclasses (Ren et al., 2023a). To address this, recent efforts have focused on improving CLIP's fine-grained classification performance through advanced prompt engineering (Zhou et al., 2022b) and refinement of pre-training supervision (Li et al., 2023b; Singh et al., 2023). Furthermore, Menon and Vondrick (2022) introduced a method that uses GPT-3 to generate detailed class descriptions to improve the prompt context of CLIP. However, this approach struggles with visually similar classes, as the generated descriptions are often too similar, limiting its effectiveness in fine-grained classification tasks.

In this paper, we explore large vision-language models (LVLMs) to further harness their vast world knowledge for fine-grained classification. Our initial experiments involved directly querying LVLMs to classify images across multiple categories. Although the overall accuracy was low, a critical insight emerged: when LVLMs made correct predictions, the correct category frequently appeared early in the sequence of choices. Moreover, we found that reducing the number of categories significantly improved accuracy, indicating that LVLMs are susceptible to positional bias and face challenges with long-context modeling (Zhao et al., 2023), as illustrated in Figure 1.

Building on these insights, we hypothesize that CLIP and LVLMs have complementary strengths for fine-grained classification tasks. Specifically,

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 1857–1871
November 12-16, 2024 ©2024 Association for Computational Linguistics

CLIP's contrastive pre-training allows it to score all possible categories for an image, which can be used to optimally order the categories for LVLMs. This transforms LVLMs' positional bias into an advantage. Although CLIP's top-1 accuracy may be limited, its top-K accuracy is significantly higher, offering a broader set of correct options. For example, on the Flowers102 dataset, CLIP (ViT-B/32) achieves a 68.7% top-1 accuracy, which increases to 89.9% for top-10 accuracy. This characteristic motivates our approach to integrating CLIP with LVLMs to enhance classification performance.

We propose CascadeVLM, a novel framework that combines the strengths of CLIP-like models and LVLMs to achieve fine-grained image classification. The key innovation of CascadeVLM is using CLIP-like models to filter and order class options, thereby enabling LVLMs to perform more effectively. Additionally, we leverage LVLMs' in-context learning (Dong et al., 2022) for few-shot tasks, and introduce an entropy-based threshold mechanism to improve inference efficiency by dynamically determining when to invoke LVLMs, allowing for early exiting (Xin et al., 2020; Li et al., 2021b).

Our zero- and few-shot experiments on various fine-grained image datasets consistently demonstrate that CascadeVLM outperforms standalone models. For instance, CascadeVLM achieves 89.0% zero-shot and 92% few-shot accuracy on the Stanford Cars dataset. On the challenging iNaturalist dataset, CascadeVLM, which utilizes advanced CLIP-like models as a backbone in conjunction with LVLMs, delivers superior performance.

Further analysis reveals that the primary performance gains stem from resolving uncertain and misclassified samples in CLIP predictions (§4.1). Additionally, we explore the sensitivity of classification results to class order (§4.2) and examine the trade-offs between computational efficiency and performance with varying entropy thresholds (§4.3).

In summary, the key contributions of this paper are: (1) Demonstrating the potential of LVLMs for fine-grained image classification. (2) Introducing the CascadeVLM framework, which effectively integrates CLIP-like models and LVLMs for zero- and few-shot fine-grained classification, providing new insights into their synergistic potential.

2 Methodology

2.1 CLIP-based Candidate Selection

As a pivotal component of our CascadeVLM framework, CLIP's operational mechanism (Radford et al., 2021) allows it to effectively discern potential correct classes, making it an ideal choice for the initial phase of candidate filtering from an extensive array of class labels.

Specifically, the function $f_{\text{CLIP}}(x, c_i)$ denotes the score outputted by the CLIP model for a specific category c_i when given an image x. Upon acquiring raw scores from CLIP for each category in the label set C, we employ a softmax function to transform these scores into a probability distribution, as delineated by:

$$P(c_i \mid x) = \frac{\exp(f_{\text{CLIP}}(x, c_i))}{\sum_{c_j \in C} \exp(f_{\text{CLIP}}(x, c_j))}. \quad (1)$$

The resulting probabilities could reflect the relative confidence of the CLIP model in associating the given image with each category within the context of the entire set C.

Based on the probability computation $P(c_i \mid x)$ specified in Equation 1, we extract and sort the top-k categories from C in descending order of probability. This selection and sorting process, crucial for the framework's efficacy, is denoted as s_{topk}. Selecting the optimal k, which ensures the correct answer is included in the top-k options, involves a straightforward validation process to identify the point where the probability converges. This step condenses the pool of candidate classes and addresses the sensitivity of LVLMs to the sequence of categories. Our empirical results 3.2 affirm that sorting based on probability significantly enhances the predictive precision of LVLMs. The generalized representation of this procedure is as follows:

$$C^* = \{c'_1, c'_2, \ldots, c'_k\} = s_{\text{topk}}(P(c_i \mid x), C), \quad (2)$$

where C^* encapsulates the optimally sorted candidates, with c'_1, c'_2 through to c'_k representing the elements in descending order of their computed probabilities.

2.2 LVLMs Prediction with Candidate Set

In this section, we seek to leverage large vision-language models (LVLMs) in our CascadeVLM framework. Capitalizing on a subset of candidates

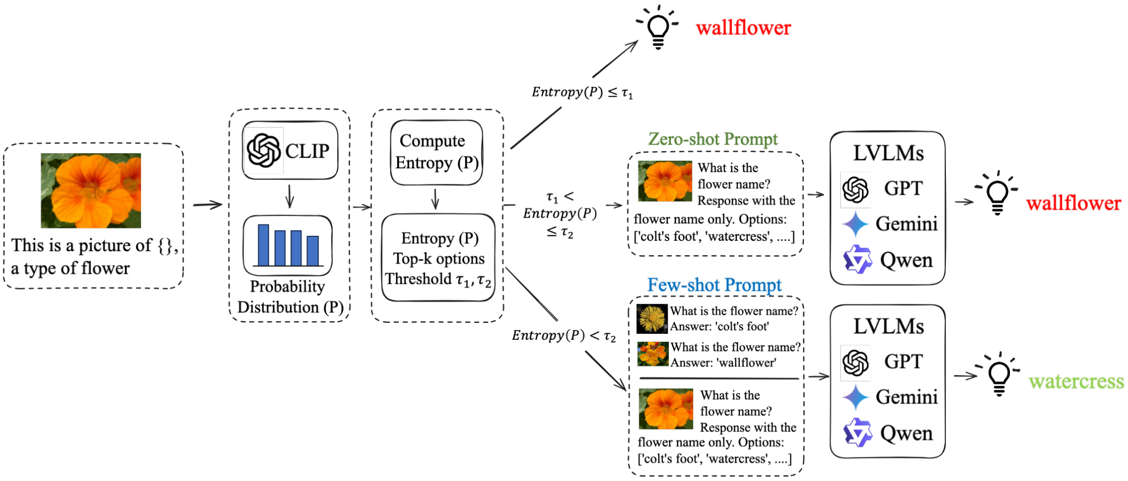

This is a picture of {}, a type of flower
CLIP
Probability Distribution (P)
Compute Entropy (P)
Entropy (P) Top-k options Threshold τ_1, τ_2
$Entropy(P) \leq \tau_1$
wallflower
$\tau_1 < Entropy(P) \leq \tau_2$
$Entropy(P) < \tau_2$
Zero-shot Prompt
What is the flower name? Response with the flower name only. Options: ['colt's foot', 'watercress', ...]
Few-shot Prompt
What is the flower name? Answer: 'colt's foot'
What is the flower name? Answer: 'wallflower'
What is the flower name? Response with the flower name only. Options: ['colt's foot', 'watercress', ...]
LVLMs
GPT
Gemini
Qwen
wallflower
LVLMs
GPT
Gemini
Qwen
watercress

Figure 2: CascadeVLM commences with CLIP for initial image analysis and probabilistic categorization, integrating an entropy threshold, τ, to balance efficiency and accuracy, culminating in LVLM's adaptive classification.

pre-selected by CLIP, LVLMs overcome the challenge of extensive context and improve prediction accuracy through adaptable zero-shot and few-shot learning strategies tailored to data-sparse environments.

Zero-Shot Prediction Zero-shot learning (Socher et al., 2013) enables models to predict unseen classes without specific training examples, leveraging pre-existing knowledge from broader contexts or related tasks. This method is particularly beneficial in data-scarce scenarios, which effectively infers new categories despite limited training data.

In the CascadeVLM framework, zero-shot prediction is executed after CLIP identifies the $top-k$ candidate classes. The LVLM then selects one candidate, $c*$, as the final prediction. Here, we generalize the process of LVLM prediction as function $f_{(LVLM)}$, given the input image x and the top-k candidate set C^*:

$$c^* = f_{\text{LVLM}}(x, C^*). \tag{3}$$

The zero-shot prediction phase in our CascadeVLM framework highlights LVLMs' proficiency in utilizing pre-trained knowledge for unseen data while adeptly managing contextual complexities.

Few-Shot Prediction In the Few-Shot Prediction phase of our CascadeVLM framework, we capitalize on LVLMs' in-context learning (Brown et al., 2020) ability, where additional relevant samples significantly enhance performance, allowing LVLMs to deepen their understanding and improve predictive accuracy.

In the integration of few-shot learning within our cascade framework, we undertake a two-step process for candidate categories set C^*:

Step 1: Context Generation: In this initial phase, for each category $c_i^{\prime}$ in C^*, we randomly select an example image $x_{c_i^{\prime}}$ from the training dataset, and design a prompt to contextualize the input image x for the LVLMs. Here, each candidate class $c_i^{\prime}$ and its corresponding example image $x_{c_i^{\prime}}$ are integrated with the prompt template to create a contextual basis. We denote this assemblage as E in the subsequent step. For instance, within the context of the GPT4-V scenario, the contextual basis denoted as E is formulated in Table 1.

<IMG: $x_{c_1^{\prime}}$>
Question: What is the class of the image? Answer: $c_1^{\prime}$

Table 1: Few-shot prompt used in our experiments.

Step 2 - Prediction with Contextual Information: In this step, the context E, embedded with rich contextual information is integrated with the input image x and fed into the LVLMs. The final classification outcome denoted as c^*, emerges from this enriched inferential framework. The process can be mathematically represented as:

$$c^* = f_{\text{LVLM}}(x, C^*, E), \tag{4}$$

where f_{LVLM} represents the LVLM prediction based on provided image x, the top-k candidate set C^* and the context set E.

Dataset	# of Class	# of Test
Flowers102	102	818
StanfordCars	196	8,041
FGVC Aircraft	100	3,333
BirdSnap	500	2,444
iNat18 (iNaturalist 2018)	8,142	24,426

Table 2: Statistics of the evaluated fine-grained image classification benchmarks.

2.3 Adaptive Entropy Threshold

In the CascadeVLM framework, we introduce an adaptive entropy-based approach to enhance inference speed and reduce the computational load on LVLMs. The entropy $H(x)$ of the probability distribution, a measure of uncertainty, is calculated as follows:

$$H(x) = -\sum_{c_i \in C} P(c_i \mid x) \log P(c_i \mid x). \quad (5)$$

This computation serves as a critical decision point. For instance, with a single entropy threshold H, samples with entropy below H are processed by CLIP alone, while LVLM handles others. With two thresholds, H_1 and H_2, samples with entropy below H_1 are processed by CLIP, those between H_1 and H_2 use the zero-shot method, and those above H_2 use the few-shot method. This method can be extended to apply different models based on various entropy thresholds.

We use a data-driven approach to determine the entropy threshold: (1) Pass the entire validation set through CLIP and compute the entropy for each sample. (2) Sort the entropy values in ascending order. (3) Set the threshold based on the desired percentage of samples to be processed by CLIP alone. For example, to have 20% of samples processed by CLIP, select the entropy value at the 20th percentile. This ensures that the entropy threshold is tailored to the specific characteristics of the validation data, enhancing the efficiency of the CascadeVLM framework.

3 Experiments

3.1 Experimental Settings

Models For experimental evaluation, we employed various CLIP models in combination with specific Large Vision-Language Models (LVLMs). The experiments utilized one of the CLIP variants CLIP ViT-B/32, ViT-B/16, or ViT-L/14 alongside QwenVL-Chat (Bai et al., 2023), Gemini-1.5-Pro (Google, 2024) or GPT-4V (OpenAI, 2023) as the LVLM. Additionally, we explore the framework's adaptability by integrating it with two robust CLIP-like models, MAWS-CLIP (Singh et al., 2023) and OpenCLIP (Ilharco et al., 2021) (ViT-G/14) pre-trained with Laion2B (Schuhmann et al., 2022) as backbones on iNaturalist(2018) (Horn et al., 2018).

Datasets We utilize a collection of datasets, each offering unique characteristics and significance for fine-grained image classification, as summarized in Table 2. These datasets include Flowers102 (Nilsback and Zisserman, 2008), StanfordCars (Krause et al., 2013), FGVC Aircraft (Maji et al., 2013), BirdSnap (Berg et al., 2014), and iNaturalist(2018) (Horn et al., 2018), collectively encompassing a wide range of categories.

3.2 Zero-shot Learning Results

Table 3 showcases the zero-shot prediction capabilities of CascadeVLM. Remarkably, CascadeVLM outperforms established methods such as CoOp, CoCoOp, and POMP without requiring any training. Moreover, it is comparable to and often outperforms the more strongly supervised method FLIP.

Applying the CascadeVLM framework, we observe that sorting candidate classes before feeding them to LVLM (Qwen Baseline vs. Qwen Cascade k=all) significantly improves performance. Further enhancement is achieved by limiting the number of candidates to the top k (Qwen Cascade k=all, Qwen Cascade k=k).

Additionally, we conducted experiments using an adaptive cross-entropy threshold, setting an entropy split point where LVLM processed 80% of samples, and the remaining 20% were handled solely by CLIP. This 20% threshold was a choice to illustrate the approach's feasibility, highlighting that users can customize this value to balance computational efficiency and performance.

3.3 Few-shot Learning Results

Our initial exploration assessed QwenVL's capacity for few-shot learning within fine-grained image classification domains. However, it became apparent that QwenVL struggled to utilize in-context demonstrations and instructions in this setting. Consequently, we focused on Gemini 1.5 Pro and GPT-4V, anticipating better alignment with our framework's requirements.

Gemini 1.5 Pro significantly improved few-shot

Model	Flower102	StanfordCars	FGVC Aricraft	BirdSnap	Avg.
Supervised	99.8 (2021)	96.3 (2021)	95.4 (2022)	90.1 (2020)	-
Qwen Baseline	37.9	23.5	9.0	6.3	19.2
CLIP ViT-B/32	68.7	59.3	19.1	51.7	49.7
Qwen Cascade (CLIP ViT-B/32, k=all)	73.0	75.1	24.0	41.7	53.5
Qwen Cascade (CLIP ViT-B/32, k=k)	74.6, k=5	**79.2**, k=10	**27.2**, k=10	**57.1**, k=3	**59.5**
Qwen Cascade (CLIP ViT-B/32, k=k w/ entropy)	**75.4**, k=5	79.0, k=10	25.0, k=10	56.9, k=3	59.1
CoOp ViT-B/16 (Zhou et al., 2022b)	68.7	64.5	18.5	-	-
CoCoOp ViT-B/16 (Zhou et al., 2022a)	71.9	65.3	22.9	-	-
POMP ViT-B/16 (Ren et al., 2023b)	72.4	66.8	25.6	-	-
CLIP ViT-B/16	73.0	64.4	24.5	52.5	53.6
Qwen Cascade (CLIP ViT-B/16, k=all)	70.7	74.9	27.4	39.4	53.1
Qwen Cascade (CLIP ViT-B/16, k=k)	73.3, k=3	**79.1**, k=10	**30.8**, k=10	**57.4**, k=3	**60.2**
Qwen Cascade (CLIP ViT-B/16, k=k w/ entropy)	**73.7**, k=3	**79.1**, k=10	29.6, k=10	**57.4**, k=3	60.0
FLIP ViT-L/14 (Li et al., 2023b)	75.0	**90.7**	29.1	63.0	64.5
CLIP ViT-L/14	**81.3**	76.2	30.9	62.2	62.7
Qwen Cascade (CLIP ViT-L/14, k=all)	76.2	79.4	31.6	44.3	57.9
Qwen Cascade (CLIP ViT-L/14, k=k)	78.5, k=3	85.6, k=10	**37.1**, k=5	63.5, k=3	66.2
Qwen Cascade (CLIP ViT-L/14, k=k w/ entropy)	78.7, k=3	85.6, k=10	36.8, k=5	**64.2**, k=3	**66.3**

Table 3: Zero-shot results comparison with different CLIP models as the backbone. The k is selected based on the validation set, and the entropy threshold allows 20% of samples to be handled solely by CLIP in each scenario. CascadeVLM achieves the best overall performance across four benchmarks.

Model	Flower102	StanfordCars	FGVC Aricraft	BirdSnap	Avg.
CLIP ViT-L/14	81.3	76.2	30.9	62.2	62.7
Gemini Baseline	77.1	80.9	57.2	44.1	64.6
Gemini Cascade (k=3, 0-shot)	84.6	87.0	51.4	69.4	73.1
Gemini Cascade (k=3, 1-shot)	**88.9**	**90.3**	**54.5**	**78.2**	**78.0**
Gemini Cascade (k=3, 1-shot, w/ entropy)	88.7	90.1	50.7	76.4	76.5
Gemini Cascade (k=5, 0-shot)	86.6	89.0	57.6	70.5	75.9
Gemini Cascade (k=5, 1-shot)	**91.6**	**92.0**	**63.9**	**80.8**	**82.1**
Gemini Cascade (k=5, 1-shot, w/ entropy)	91.5	91.8	59.4	79.1	80.5

Table 4: Few-shot learning results using Gemini-1.5-Pro as the LVLM. The performance improves when applying the cascade framework in the zero-shot setting and further improves with the one-shot setting.

learning performance across various fine-grained image classification datasets. As shown in Table 4, we maintained the same k value for both zero-shot and few-shot settings to observe the improvements better. The few-shot results consistently outperformed the zero-shot scenarios.

Further, we applied two entropy thresholds to explore the adaptive entropy threshold method. The first threshold allowed 20% of samples, which CLIP was most confident about, to be handled solely by CLIP. Another 20% of harder samples were processed using the zero-shot method, while the remaining most difficult samples, with entropy larger than the second threshold, were handled using the one-shot method. The choice of 20% was arbitrary and serves as an illustration, with users able to adjust these thresholds to balance efficiency and performance. This stratified approach enabled different strategies based on sample difficulty, saving time and cost while maintaining comparable performance.

For the best result, Gemini Cascade with CLIP ViT-L/14 (k=5, 1-shot) achieved the highest accuracy, such as 92.0% on StanfordCars. Moreover, detailed results for GPT-4V, which further validate the robustness of our approach, are provided in Table 5 of Appendix B.

3.4 Performance Evaluation for iNaturalist

To rigorously evaluate the performance of our CascadeVLM framework on highly complex and fine-grained tasks, we tested it on the INat18 dataset, which comprises 8,142 classes for detailed image classification. Due to the large number of classes, we set $k = 50$ for cascading, where the probability of the correct answer being within the top 50 options tends to converge.

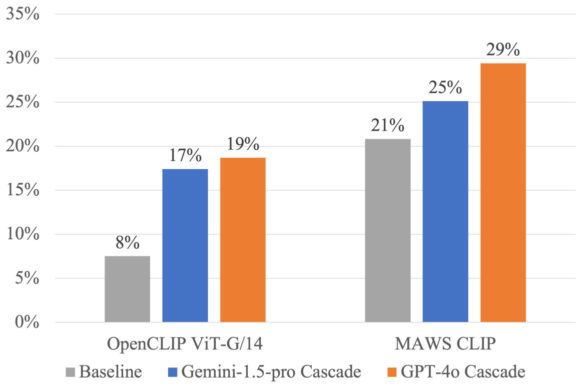


Figure 3: Performance on the challenging iNaturalist dataset with different cascaded models.

As illustrated in Figure 3, the MAWS CLIP model reached an accuracy of 20.8%, while OpenCLIP ViT-G/14 only achieved 7.5%, demonstrating the considerable difficulty of this dataset for vision-language models (VLMs). However, cascading with large vision-language models (LVLMs) resulted in significant performance improvements. Specifically, using OpenCLIP ViT-G/14 as the base model, cascading with Gemini-1.5-pro raised the accuracy to 17.4%, and with GPT-4o, the accuracy improved to 18.7%. Similarly, with MAWS CLIP as the base model, cascading with Gemini-1.5-pro increased the accuracy to 25.1%, and with GPT-4o, it further increased to 29.4%.

These results clearly demonstrate the effectiveness of the CascadeVLM framework in improving performance across various VLM and LVLM combinations on this highly challenging dataset. Additional results, including experiments with GPT-4V, are provided in Appendix B.

4 Analysis

In this section, we explore various aspects of CascadeVLM, highlighting the underlying reasons for its enhanced performance, the trade-off of the entropy threshold, and more. More investigations can be found in the Appendix C.

4.1 Performance Gain Analysis

This analysis seeks to demonstrate why cascading CLIP with an LVLM model leads to improved accuracy in classification tasks. Using the Flowers102 dataset as a case study, we assess the performance of CLIP and the enhancement brought by LVLM. The margin (Settles, 2009), i.e., the difference between the top1 and top2 probability scores from CLIP, serves as an indicator of the model's certainty about its prediction, where smaller margins suggest greater ambiguity in the image classification.

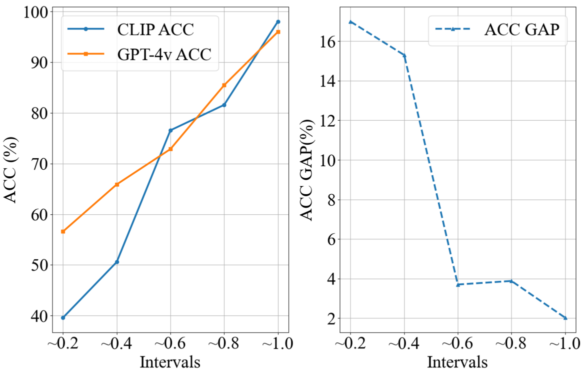


Figure 4: Comparative Analysis of ACC performance between CLIP and GPT-4V across different intervals of classification certainty. The left graph shows the ACC of both models across varying levels of margin. The right graph presents the ACC gap between the two models.

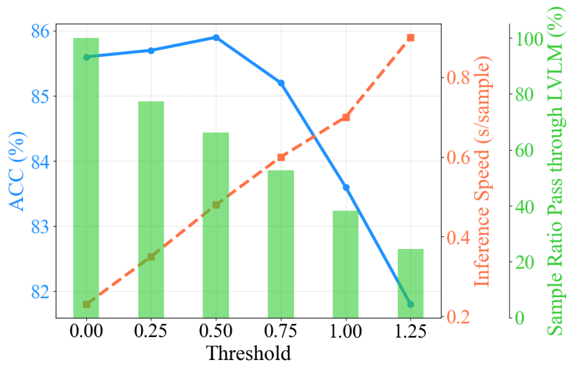
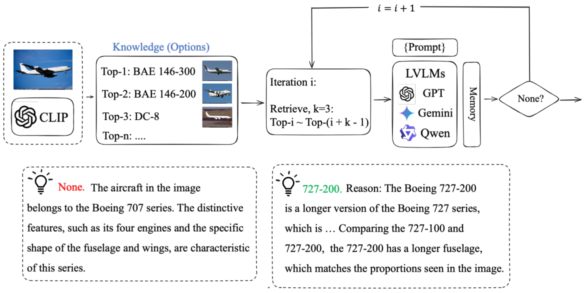


Figure 5: Performance variation in the StanfordCars dataset with varying entropy thresholds using CLIP-ViT-L/14 for cascading, set at top-k=10. An increase in entropy threshold results in decreased inference speed and reduced accuracy.

We divide the range of margins into five intervals, from 0 to 1, to analyze the effects systematically. The data reveals that GPT-4V, representing LVLM, significantly outperforms CLIP with margins less than 0.4, where CLIP experiences confusion. This is evident in the consistently high ACC for GPT-4V in these instances. When the margin exceeds 0.6, the ACC for CLIP improves, indicating that the model is more confident and accurate in its predictions, thus reducing the gap in performance between CLIP and LVLM. The accompanying Figure 4 illustrates this trend, with the ACC gap decreasing sharply as the margin increases. This pattern suggests that while LVLM provides a significant advantage in cases of high ambiguity, the benefit tapers off as CLIP's confidence in its classifications rises.



None. The aircraft in the image belongs to the Boeing 707 series. The distinctive features, such as its four engines and the specific shape of the fuselage and wings, are characteristic of this series.

727-200. Reason: The Boeing 727-200 is a longer version of the Boeing 727 series, which is … Comparing the 727-100 and 727-200, the 727-200 has a longer fuselage, which matches the proportions seen in the image.

Figure 6: Iterative refinement process in CascadeVLM. CLIP ranks all candidates with a 1-shot image as the knowledge base. If LVLM determines the correct answer is not within the top-k options, it iteratively retrieves additional top-k sets from the knowledge base.

4.2 Sensitivity of Option Orders

As delineated in Tables 3, we find, surprisingly, that the arrangement of options provided by CLIP plays a pivotal role in the efficacy of Language-Vision Language Models (LVLMs). While it may seem a minor detail to supply LVLMs with the entire class set from CLIP, this procedure is significantly impactful. For example, as shown in Table 3, presenting all classes in a random order to Qwen results in an average accuracy of only 19.2%. Conversely, when the classes are organized according to the probabilities assigned by CLIP, there is a notable enhancement in performance. Specifically, in the case where CLIP(ViT-L/14) cascades with Qwen, offering a fully ordered class set, there is a substantial accuracy increase of 38.7% across various datasets.

4.3 Inference Efficiency Analysis

This section critically evaluates the efficacy of implementing an entropy threshold within the CascadeVLM framework. To ensure clarity and simplicity, we apply the entropy threshold exclusively to the CLIP and zero-shot methods, using a 0.25-step increment for the threshold. In this analysis, no batch operation is applied, and each model, QwenVL and CLIP, is deployed in a single V100 GPU environment. As illustrated in Figure 5, the accuracy initially increases slightly but then decreases sharply as the entropy threshold is raised. Meanwhile, the inference speed increases as fewer samples pass through the LVLM. This result indicates a direct correlation between increasing the entropy threshold and heightened inference speed, albeit at the cost of reduced accuracy.

4.4 Iterative Cascading with Self-Refinement

As LLMs excel in combining reasoning and action for iterative learning and refinement (Yao et al., 2023), we are motivated to explore the error correction and explainability capabilities within the CascadeVLM framework. We conducted experiments on the FGVC Aircraft dataset with a small $k = 3$, where the CLIP's top-3 accuracy was 62.1%, limiting the CascadeVLM's few-shot accuracy to 54.5%. Figure 6 illustrates the overall process, where a knowledge base is constructed as in-context augmentation for iterative retrieval (Lewis et al., 2021) with the options sorted according to CLIP's prediction probability. The LVLM is then prompted to provide reasoning for its selections and to return a special result *None*, when it thinks that the top-k options do not contain the correct candidate. This step is performed iteratively until the LVLM stops to ask for more candidates, i.e., increasing k iteratively via self-refinement.

Our results on the FGVC dataset show that the LVLM (Gemini 1.5 Pro) asks for more candidates in 520 samples out of 3.3k evaluated cases. The iterative process successfully corrects predictions in 302 cases, boosting the few-shot accuracy from 54.5% to 60.5%.

This demonstrates the effectiveness of the iterative cascading approach in improving accuracy through error correction and indicates that our CascadeVLM has great potential to harness the unique capabilities of both VLMs.

4.5 Case Study

Our case study analysis examines four distinct scenarios in a $k = 3$ setting using Cascading CLIP ViT-L/14 and Gemini.

Case 1 presents a scenario where CLIP's top-1 prediction is incorrect, yet the ground truth is within its top-3 predictions. Leveraging the LVLM's discernment, the correct answer is selected.

Case 2 depicts a situation where, despite CLIP including the correct answer in its top-3 predictions, the LVLM fails to identify it correctly. This highlights potential areas for refinement in the LVLM's decision-making process.

Case 3 demonstrates a complete misalignment where both CLIP and the LVLM fail to recognize the correct class within the top-3 predictions, leading to a compounded error.

Case 4 shows that, even when CLIP's top-3 pre-

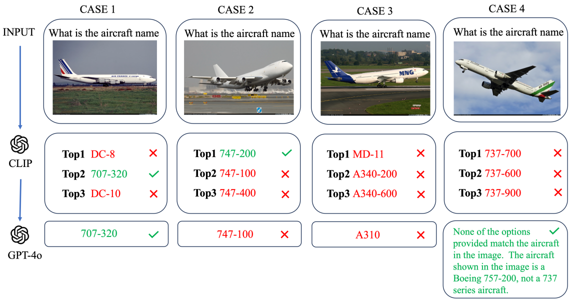

CASE 1
CASE 2
CASE 3
CASE 4
INPUT
What is the aircraft name
What is the aircraft name
What is the aircraft name
What is the aircraft name
CLIP
Top1 DC-8 ✗
Top2 707-320 ✓
Top3 DC-10 ✗
Top1 747-200 ✓
Top2 747-100 ✗
Top3 747-400 ✗
Top1 MD-11 ✗
Top2 A340-200 ✗
Top3 A340-600 ✗
Top1 737-700 ✗
Top2 737-600 ✗
Top3 737-900 ✗
GPT-4o
707-320 ✓
747-100 ✗
A310 ✗
None of the options provided match the aircraft in the image. The aircraft shown in the image is a Boeing 757-200, not a 737 series aircraft. ✓

Figure 7: Three case studies demonstrating the cascade process from CLIP predictions to LVLM refinement for bird species classification.

dictions fail, the LVLM can identify the correct answer based on its own knowledge.

These cases highlight the complexities of fine-grained image classification and reaffirm the need for integrated approaches like CascadeVLM to capitalize on the strengths of both CLIP and LVLMs.

5 Related Work

Vision Language Models Building vision language models (VLMs) for understanding the multi-modal world has been an active research area. Pilot studies leverage pre-training concepts from NLP (Devlin et al., 2019), learning shared representations across modalities from mixed visual and language inputs (Li et al., 2019; Tan and Bansal, 2019; Su et al., 2020; Chen et al., 2019; Li et al., 2020). Among these, Radford et al. (2021) introduced CLIP, a contrastive language-image pre-training framework that employs language as supervision, demonstrating potential for multi-modal tasks and inspiring subsequent variants for improvement (Jia et al., 2021; Li et al., 2022b, 2023b, 2021a, 2022a). The evolution of large language models like ChatGPT (OpenAI, 2022) has motivated the development of large vision language models (LVLMs), combining powerful vision encoders like CLIP with large language models such as LLaMa (Touvron et al., 2023) and Vicuna (Chiang et al., 2023). Achieved through large-scale modality alignment training on image-text pairs (Alayrac et al., 2022; Awadalla et al., 2023) and supervised fine-tuning on multi-modal instruction tuning datasets (Liu et al., 2023; Li et al., 2023a), resulting LVLMs like GPT-4V (OpenAI, 2023), QwenVL (Bai et al., 2023) and Gemini-1.5-pro (Google, 2024) exhibit promising perceptual and cognitive abilities (Yang et al., 2023) for engaging user queries.

Fine-grained Image Classification Fine-grained image recognition, involving categorization into subordinate classes within a broader category, such as cars (Krause et al., 2013) and aircraft models (Maji et al., 2013), demands fine-grained feature learning. Previous work explores diverse strategies, including local-global interaction modules with attention mechanisms (Fu et al., 2017; Zheng et al., 2017), end-to-end feature encoding with specialized training objectives (Dubey et al., 2018; Chang et al., 2020), and the incorporation of external knowledge bases or auxiliary datasets (Chen et al., 2018; Xu et al., 2018). These approaches offer potential enhancements similar to our CLIP model, which we identify as a future exploration for improved performance.

CLIP Enhancements for Fine-grained Image Classification Recent studies have enhanced the CLIP primarily via prompt engineering and pre-training techniques. In prompt engineering, CoOp (Zhou et al., 2022b) introduces an innovative method by learning context words as continuous vectors. Extending this idea, CoCoOp (Zhou et al., 2022a) incorporates a lightweight neural network to generate input-specific image tokens, further improving model performance. POMP (Li et al., 2023b) proposes pre-training a general soft prompt on the ImageNet-21K dataset for universal visual tasks. Besides, Menon and Vondrick (2022) instead employs GPT-4 to generate better descriptive prompts for classification, enriching the prompting context for CLIP models. In the realm of pre-training, MAWS (Singh et al., 2023) combines Masked Autoencoder (MAE) pre-training with weakly supervised learning, significantly enhancing the learning efficacy. Similarly, FLIP (Li et al., 2023b) increases prediction accuracy by masking substantial portions of image patches, facilitating processing more image-text pairs within the same timeframe and boosting performance across various tasks. Unlike previous studies, our CascadeVLM explores the integration of LVLMs for leveraging their world knowledge to handle similar classes effectively.

6 Conclusion

In this paper, we propose CascadeVLM, harnessing the advantages of CLIP and LVLMs for fine-grained image classification. By utilizing CLIP for selecting the potential candidate class, LVLM can make more accurate predictions for image classes with subtle differences. Experimental results on four benchmarks demonstrate the effectiveness of our proposed framework. Further extension to the few-shot setups showcases the great potential of the cascading framework to leverage the in-context learning ability of LVLMs.

Acknowledgments

We thank all anonymous reviewers for their constructive comments and suggestions, which helped us improve the quality of this work. We are particularly grateful to Lei Li for his valuable feedback in preparing the manuscript.

Limitations

The efficacy of our CascadeVLM framework hinges critically on the symbiotic interplay between the CLIP model and LVLMs. A key limitation emerges when CLIP's top-K accuracy is insufficient, failing to encompass correct options in LVLM's narrowed candidate set, thereby limiting the scope for enhanced accuracy. Moreover, if CLIP outperforms the LVLM in fine-grained classification, incorporating an LVLM with relatively inferior capabilities may inadvertently diminish overall accuracy. These dynamics underscore the imperative for meticulous selection and alignment of models, ensuring each component's strengths are effectively leveraged within the cascade architecture.

The CascadeVLM framework mainly utilizes the LVLM's extensive familiarity and common knowledge with the dataset. If lack of such knowledge, the accuracy would not be good enough. But we believe the integration of Agent or RAG approches (Lewis et al., 2020) may offer a solution to this limitation. Besides, augmenting the context for LVLM with additional information may also be a fact to be explored. Besides the few-shot learning we applied in experiments, we also tested sending the CLIP's prediction scores to LVLM. Unfortunately, this led to a reduction in accuracy, with a nearly 4% increase in erroneous predictions, attributed to LVLM's over-reliance on CLIP's scores. This highlights the need for a balanced approach to information feeding within the cascade framework.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. 2022. Flamingo: a visual language model for few-shot learning. *ArXiv preprint*, abs/2204.14198.

Anas Awadalla, Irena Gao, Josh Gardner, Jack Hessel, Yusuf Hanafy, Wanrong Zhu, Kalyani Marathe, Yonatan Bitton, Samir Gadre, Shiori Sagawa, Jenia Jitsev, Simon Kornblith, Pang Wei Koh, Gabriel Ilharco, Mitchell Wortsman, and Ludwig Schmidt. 2023. Openflamingo: An open-source framework for training large autoregressive vision-language models. *ArXiv preprint*, abs/2308.01390.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *ArXiv preprint*, abs/2308.12966.

Asish Bera, Zachary Wharton, Yonghuai Liu, Nik Bessis, and Ardhendu Behera. 2022. Sr-gnn: Spatial relation-aware graph neural network for fine-grained image categorization. *IEEE Transactions on Image Processing*, 31:6017–6031.

Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L. Alexander, David W. Jacobs, and Peter N. Belhumeur. 2014. Birdsnap: Large-scale fine-grained visual categorization of birds. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2019–2026.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Dongliang Chang, Yifeng Ding, Jiyang Xie, Ayan Kumar Bhunia, Xiaoxu Li, Zhanyu Ma, Ming Wu, Jun Guo, and Yi-Zhe Song. 2020. The devil is in the channels: Mutual-channel loss for fine-grained image classification. *IEEE Transactions on Image Processing*, 29:4683–4695.

Tianshui Chen, Liang Lin, Riquan Chen, Yang Wu, and Xiaonan Luo. 2018. Knowledge-embedded representation learning for fine-grained image recognition. In *International Joint Conference on Artificial Intelligence*.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019. Uniter: Learning universal image-text representations. *ArXiv*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2022. A survey for in-context learning.

Abhimanyu Dubey, Otkrist Gupta, Ramesh Raskar, and Nikhil Naik. 2018. Maximum-entropy fine-grained classification. *ArXiv*, abs/1809.05934.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. 2020. Sharpness-aware minimization for efficiently improving generalization. *CoRR*, abs/2010.01412.

Jianlong Fu, Heliang Zheng, and Tao Mei. 2017. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4476–4484.

Gemini Team Google. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context.

Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. 2021. Escaping the big data paradigm with compact transformers. *CoRR*, abs/2104.05704.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. 2018. The inaturalist species classification and detection dataset.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taori, Achal Dave, Vaishaal Shankar, Hongseok Namkoong, John Miller, Hannaneh Hajishirzi, Ali Farhadi, and Ludwig Schmidt. 2021. Openclip. If you use this software, please cite it as below.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 4904–4916.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 2013. 3d object representations for fine-grained categorization. In *4th IEEE Workshop on 3D Representation and Recognition, at ICCV 2013 (3dRR-13)*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *CoRR*, abs/2005.11401.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. 2022a. BLIP: bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900.

Junnan Li, Ramprasaath R. Selvaraju, Akhilesh Gotmare, Shafiq R. Joty, Caiming Xiong, and Steven Chu-Hong Hoi. 2021a. Align before fuse: Vision and language representation learning with momentum distillation. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 9694–9705.

Lei Li, Yankai Lin, Deli Chen, Shuhuai Ren, Peng Li, Jie Zhou, and Xu Sun. 2021b. Cascadebert: Accelerating inference of pre-trained language models via calibrated complete models cascade. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 475–486.

Lei Li, Yuwei Yin, Shicheng Li, Liang Chen, Peiyi Wang, Shuhuai Ren, Mukai Li, Yazheng Yang, Jingjing Xu, Xu Sun, Lingpeng Kong, and Qi Liu. 2023a. M^{3}IT: A large-scale dataset towards multi-modal multilingual instruction tuning. *ArXiv preprint*, abs/2306.04387.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *ArXiv*.

Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Proc. of ECCV*.

Yangguang Li, Feng Liang, Lichen Zhao, Yufeng Cui, Wanli Ouyang, Jing Shao, Fengwei Yu, and Junjie Yan. 2022b. Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*.

Yanghao Li, Haoqi Fan, Ronghang Hu, Christoph Feichtenhofer, and Kaiming He. 2023b. Scaling language-image pre-training via masking.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *ArXiv preprint*, abs/2304.08485.

Xinyu Ma, Xu Chu, Zhibang Yang, Yang Lin, Xin Gao, and Junfeng Zhao. 2024. Parameter efficient quasi-orthogonal fine-tuning via givens rotation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 33686–33729. PMLR.

S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. 2013. Fine-grained visual classification of aircraft. Technical report.

Sachit Menon and Carl Vondrick. 2022. Visual classification via description from large language models.

Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*.

OpenAI. 2022. Introducing chatgpt.

OpenAI. 2023. Gpt-4v(ision) system card.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763.

Shuhuai Ren, Lei Li, Xuancheng Ren, Guangxiang Zhao, and Xu Sun. 2023a. Delving into the openness of CLIP. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9587–9606, Toronto, Canada. Association for Computational Linguistics.

Shuhuai Ren, Aston Zhang, Yi Zhu, Shuai Zhang, Shuai Zheng, Mu Li, Alex Smola, and Xu Sun. 2023b. Prompt pre-training with twenty-thousand classes for open-vocabulary visual recognition.

Tal Ridnik, Emanuel Ben Baruch, Asaf Noy, and Lihi Zelnik-Manor. 2021. Imagenet-21k pretraining for the masses. *CoRR*, abs/2104.10972.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa R Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. 2022. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Burr Settles. 2009. Active learning literature survey.

Mannat Singh, Quentin Duval, Kalyan Vasudev Alwala, Haoqi Fan, Vaibhav Aggarwal, Aaron Adcock, Armand Joulin, Piotr Dollár, Christoph Feichtenhofer, Ross Girshick, Rohit Girdhar, and Ishan Misra. 2023. The effectiveness of mae pre-pretraining for billion-scale pretraining. In *ICCV*.

Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-shot learning through cross-modal transfer. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. VL-BERT: pre-training of generic visual-linguistic representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *ArXiv preprint*, abs/2302.13971.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. 2020. Deebert: Dynamic early exiting for accelerating bert inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2246–2251.

Zhe Xu, Shaoli Huang, Ya Zhang, and Dacheng Tao. 2018. Webly-supervised fine-grained visual categorization via deep domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:1100–1113.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Heliang Zheng, Jianlong Fu, Tao Mei, and Jiebo Luo. 2017. Learning multi-attention convolutional neural network for fine-grained image recognition. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5219–5227.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022a. Conditional prompt learning for vision-language models.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022b. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348.

Appendix

A Prompt Tuning of Qwen

A.1 Zero-shot Prompt Tunning of Qwen

We experimented with various prompt designs to optimize Qwen's performance in selecting the top-k categories. Two representative prompt styles were identified, each with distinct characteristics and performance implications.

The first prompt style, while intuitive, occasionally led to non-compliant responses. For example, Qwen would select a flower name not listed in the given options or use an alias instead of the specified name. This approach yielded suboptimal results.

Subsequently, we adapted our prompts to align more closely with the Qwen training data, where the keyword "options" was prevalent. This adaptation significantly improved compliance and accuracy in the model's responses. Thus, for the overall experiment, we use 'PROMPT2'. For GPT-4V, we applied a similar prompt style but followed the API requirement.

PROMPT 1:

```
Picture 1: <img>....jpg</img>
Please examine the flower image
    ↪ and identify the most
    ↪ suitable flower name
    ↪ corresponding to the image
    ↪ content from the list of
    ↪ flower names below.
    ↪ Remember to select only one
    ↪  flower name from the list,
    ↪  and respond with the
    ↪ flower name ONLY. Available
    ↪  flower names: [...]
```

PROMPT 2:

```
Picture 1: <img>...jpg</img>
Question: What is the flower's
    ↪ name? Remember to select
    ↪ only one flower name from
    ↪ the options and respond
    ↪ with the flower name only.
    ↪ Options: [...]
```

A.2 Few-Shot Prompt Tunning of Qwen

In the domain of few-shot learning, we conducted experiments with QwenVL and observed challenges in its ability to utilize in-context demonstrations and follow instructions effectively. Our experimentation involved different prompt structures in the context of the CLIP-ViT B/32 model with a top-$k = 10$ setting on the Flower102 dataset.

The initial two prompts led to moderate success, achieving an accuracy of approximately 50%. However, the implementation of the final prompt design demonstrated a notable improvement, yielding an accuracy close to 68%. This highlights the impact of prompt design on the model's ability to leverage few-shot learning effectively.

To corroborate the versatility of our Cascade-VLM framework, we conducted few-shot learning experiments with GPT-4V. These trials demonstrated the framework's adaptability across different LVLMs, reinforcing its effectiveness in diverse data-rich scenarios.

PROMPT 1:

```
<img>...jpg</img> Question: What
    ↪ is the flower name? Options
    ↪ : [...] Answer: ...
<img>...jpg</img> Question: What
    ↪ is the flower name? Options
    ↪ : [...] Answer: ...
```

```
<img>...jpg</img> Question: What
   ↪ is the flower name? Options
   ↪ : [...] Answer: ...
...
<img>...jpg</img> Question: What
   ↪ is the flower name? Answer:
   ↪  ...
```

PROMPT 2:

```
Picture 1: <img>...jpg</img>
   ↪ Question: What is the
   ↪ flower name? Options: [...]
   ↪  Answer: ...
Picture 2: <img>...jpg</img>
   ↪ Question: What is the
   ↪ flower name? Options: [...]
   ↪  Answer: ...
Picture 3: <img>...jpg</img>
   ↪ Question: What is the
   ↪ flower name? Options: [...]
   ↪  Answer: ...
...
Picture 4: <img>...jpg</img>
   ↪ Question: What is the
   ↪ flower name? Options: [...]
   ↪  Answer:
```

PROMPT 3:

```
Picture 1: <img>...jpg</img>
   ↪ Question: What is the
   ↪ flower name? Answer: ...
Picture 2: <img>...jpg</img>
   ↪ Question: What is the
   ↪ flower name? Answer: ...
Picture 3: <img>...jpg</img>
   ↪ Question: What is the
   ↪ flower name? Answer: ...
...
Picture 4: <img>...jpg</img>
   ↪ Question: What is the
   ↪ flower name? Options: [...]
   ↪  Answer:
```

PROMPT 4:

```
Picture 1: <img>...jpg</img>
   ↪ Answer: ...
Picture 2: <img>...jpg</img>
   ↪ Answer: ...
Picture 3: <img>...jpg</img>
   ↪ Answer: ...
...
Picture 4: <img>...jpg</img>
   ↪ Question: What is the
   ↪ flower name? Options: [...]
   ↪  Answer:
```

B Additioanl Experimental Results

B.1 Few-shot Experiments with GPT-4V

In Table 5, our experiments with GPT-4V were limited to a random subset of 200 samples per dataset due to budget constraints. To ensure fairness in our subset selection, we compare the subsample result of CLIP ViT-L/14 and the baseline result of CLIP ViT-L/14, demonstrating that our selection process was fair and the difference is negligible.

In this experiment, we utilize top-k (k=5) for few-shot experiments and yield even more pronounced improvements in predictive accuracy. For instance, with few-shot learning applied, the Flower102 dataset achieved an impressive 94.5% accuracy, while the StanfordCars dataset attained 88.5%. These results reaffirm the effectiveness of our cascade framework and highlight its adaptability and efficiency in leveraging few-shot learning for fine-grained classification tasks.

B.2 Performance Evaluation for iNaturalist and SUN397 with GPT-4V

Table 6, we explore the performance of cascading framework on challenging datasets iNaturalist(2018) (Horn et al., 2018) and SUN397 (Xiao et al., 2010). We utilize MAWS and OpenCLIP ViT-G/14 as VLM backbone, cascading with GPT-4V as LVLM. Because of the budget limitation, we explore 500 subsamples for each dataset. We observe that performance improves by applying the cascade framework in each scenario.

C Analysis

C.1 Influence of candidate classes number k

In the analysis of the influence of the number of candidate classes, k, on classification performance, two distinct configurations of the CLIP model, namely CLIP-ViT-B/16 and CLIP-ViT-L/14, as well as the integration of CLIP ViT-B/32 with Qwen in a cascade framework, have been explored. The investigation reveals dataset-specific optimal settings for k. Specifically, for the StanfordCars and FGVC Aircraft datasets, peak performance is observed at a top-10 setting across different configurations, with an interesting shift to top-5 for the FGVC Aircraft dataset when using the ViT-L/14 model, highlighting an enhancement in baseline

Model	Flower102	StanfordCars	FGVC Aricraft	BirdSnap	Avg.
CLIP ViT-L/14 (baseline)	81.3	76.2	30.9	62.2	62.7
CLIP ViT-L/14 (subsample)	82.0	75.0	30.0	60.5	61.9
GPT-4V Baseline (subsample)	67.5	74.0	61.5	46.0	62.3
GPT-4V Cascade (subsample, k=5)	86.5	85.5	56.0	62.0	72.5
GPT-4V Cascade (subsample, k=5) + 1-shot	**94.5**	**88.5**	**63.0**	**72.5**	**79.7**

Table 5: Few-shot learning results with GPT-4V as the LVLM. GPT-4V can better utilize the in-context demonstrations to achieve superior results for fine-grained classification. The result of CasecadeVLM is superior overall datasets.

Model	iNat18	SUN397
MAWS-CLIP	20	71.0
MAWS-CLIP Cascade	**26.8**	**75.4**
OpenCLIP (ViT-G/14)	6.6	74.4
OpenCLIP (ViT-G/14) Cascade	**11.8**	**77.6**

Table 6: Zero-shot prediction result comparison with MAWS and OpenCLIP ViT-G/14 as backbone, cascading GPT-4V.

performance. In contrast, the Flower102 and BirdSnap datasets exhibit optimal results at a top-3 setting, with the Flower102 dataset showing a superior accuracy with the CLIP-ViT-L/14 model, attributed to its intrinsic fine-grained image classification capability. This suggests that the CLIP-ViT-L/14 model's performance surpasses that of the Qwen LVLM in specific cases. Furthermore, the validation performance across different datasets demonstrates a dependency on the chosen value of k, indicating a nuanced behavior where the intrinsic properties of each dataset may favor a different range of candidate classes. This behavior underscores the importance of tailoring the cascade framework's parameters to the specific dataset at hand to achieve optimal performance, as evidenced by the gradual improvement in accuracy with a narrower focus in candidate classes for certain datasets, and a discernible peak before a decline in others, suggesting a balance between too few and too many options is crucial for maximizing classification accuracy.

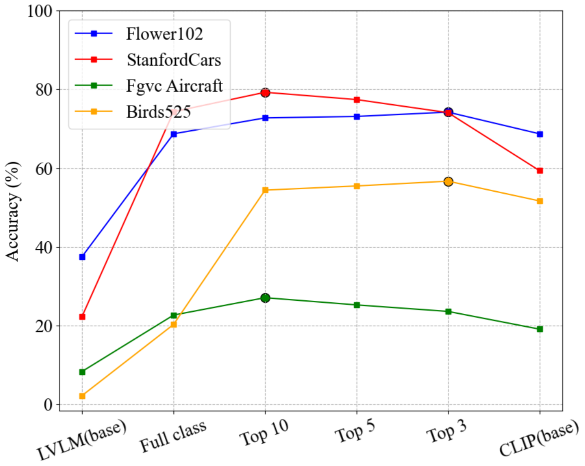


Figure 8: Performance changes with varied k with CLIP-ViT-B/32.

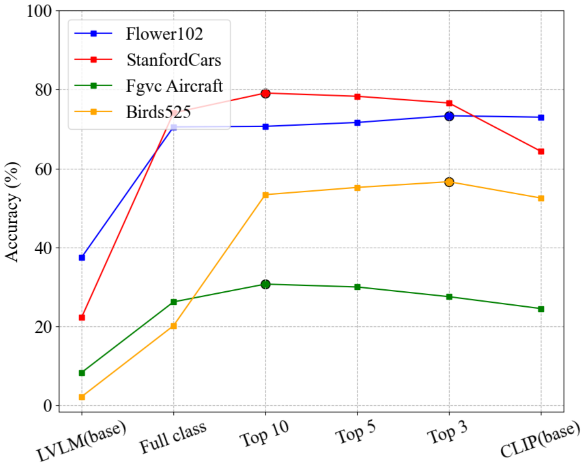


Figure 9: Performance changes with varied k with CLIP-ViT-B/16.

C.2 Error Analysis

An error analysis was conducted on the BirdSnap dataset using the cascade framework, which incorporates CLIP (ViT-L/14) for initial classification and Qwen as the LVLM for refined categorization with $k = 10$, as shown in Figure 11. When entropy is lower than the threshold, prediction is only processed by CLIP, in this case, 148 misclassifications were noted (*CLIP WRONG*). Otherwise, after the CLIP narrows down the options of classes, the LVLM Qwen would do the final classification. In this case, LVLM resulted in 812 misclassifications (*LVLM Wrong*), which further breaks down into two categories: 212 instances where the correct option was not present in the top-10 candidates given by CLIP(*LVLM Wrong not in Options*), and 600 instances where the correct option was present,

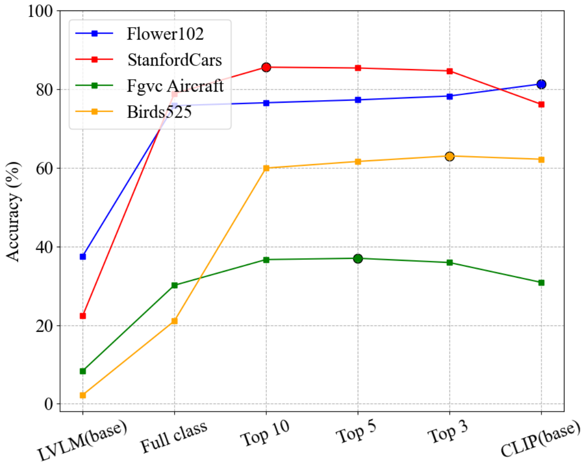


Figure 10: Performance changes with varied k with CLIP-ViT-L/14.

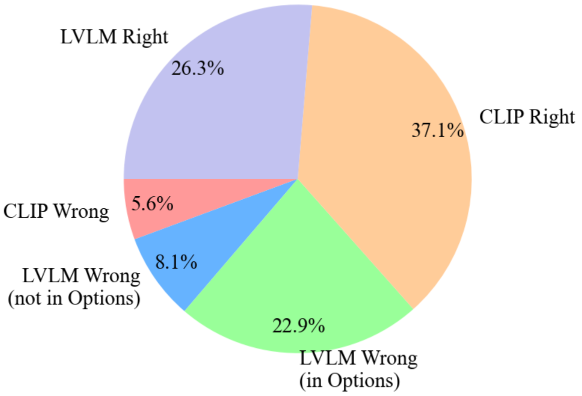


Figure 11: Error analysis of the BirdSnap dataset with an entropy threshold of 1.25 and top-k=10. The analysis reveals that despite CLIP including correct options, LVLM frequently misclassifies.

but the LVLM failed to identify it (*LVLM Wrong* $in\ Options$).