Abstract

Image classification has emerged as a pivotal task within the realm of computer vision. With the recent development of immensely powerful visual large language models, can they handle the task with more skill than previously developed vision models like CLIP? We investigate the zero-shot capabilities of a 13-billion parameter LLaVA model and find that although it lacks competence at its original state, it can be enhanced significantly with a combination of carefully crafted prompts, hierarchical classification strategies, and adjusted model temperatures. Compared to its initial state, we achieve an average increase of 10 percentage points in classification accuracy across all datasets without any fine-tuning. Though more improvement is necessary to outperform CLIP, these findings underscore the potential of visual large language models on image classification.

1. Introduction

In the field of computer vision, zero-shot image classification has emerged as a dynamic area of research, driven by the remarkable performance of pre-trained vision models like CLIP (Contrastive Language-Image Pre-training). CLIP has showcased impressive accuracy, yet it is bound by a limitation - it can only select the closest text label from a given list of options, rendering it unable to perform classification without predefined choices.

On the other hand, a visual large language model would be able to transcend this constraint and generate novel text labels on the spot for any given image. Taking in an image and textual prompt as input, it is able to generate the appropriate output without a predetermined list of outputs to select from. Specifically, LLaVA: Large Language and Visual Assistant is a recently released multimodal model tailored specifically for vision-language tasks. With its fusion of CLIP’s vision encoder and LLaMA (Large Language Model Meta AI), a transformer-based language model, LLaVA boasts an enormous scale of 13 billion parameters, in contrast to CLIP’s 63 million. This allows LLaVA not only to classify the primary object in an image but to weave a tapestry of rich descriptions, capturing every aspect of the visual content. Figure 1 provides a glimpse into LLaVA’s basic architecture.

Figure 1: LLaVA’s vision encoder is derived from CLIP, while the language model is derived from LLaMA. Once the image features are converted into language tokens, LLaVA’s final output is constructed within the transformer architecture.

Since LLaVA has not been tested with zero-shot classification, our focus centers on evaluating this task, specifically in comparison to CLIP. And if it fails to outperform CLIP, how can we improve it?

We start by evaluating LLaVA off-the-shelf with a simple prompt, then experimenting with further prompt engineering, inference styles, and model temperature to raise the classification accuracy. We explore pros and cons of different prompting methods and establish an evaluation procedure that can be applied to test other visual large language models as well.

2. Related Work

2.1. CLIP Model

CLIP was released by OpenAI in 2021, and it has shown that zero-shot classification can be done quite accurately with its retrieval method when pre-trained on millions of (image, text) pairs (et al. [6]). Nevertheless, CLIP is limited by its reliance on predefined class labels; what if we seek a more flexible model that can classify objects without a list of options to choose from? Or what if we desire a richer, more descriptive answer, rather than a simple class label?
2.2. LLaVA Model

This is what draws us to a visual large language model such as LLaVA. With its vision encoder derived from CLIP and a transformer-based language model derived from LLaMA, LLaVA has the power to not only perform simple classification tasks but also generate a rich set of descriptions and analyses for all aspects of the given image (et al. [4]). The model’s prior evaluations have revolved around its responses to conversational and scientific inquiries, leaving a void in the exploration of its image classification capabilities. It is within this space that our work focuses on. We investigate whether LLaVA’s performance in zero-shot classification is robust enough to surpass the efficacy of visual-only models like CLIP.

2.3. Hierarchical Classification

Evaluating a complex model like LLaVA in the context of image classification presents some challenges. Notably, datasets such as ImageNet ([1] with their plethora of hyper-specific class labels demand a thoughtful approach. To navigate this, we draw inspiration from Wen et al. [226], who propose a hierarchical classification strategy in their work “What can we learn from misclassified ImageNet images?”. In their two-stage approach, they first discern the broad superclass of an image, followed by usage of a separate network to pinpoint the subclass. Similarly, our work confirms that misclassifications occur primarily within the correct superclass, rather than across superclasses. The main point of distinction of our work, however, is that we introduce a method to achieve the same hierarchical classification with a single inference round. By employing this approach, we optimize the efficiency for the evaluation process.

2.4. Using Descriptors

Furthermore, we aim to take advantage of LLaVA’s rich vocabulary to compete against CLIP. In alignment with the findings of Menon and Vondrick [5] in their work “Visual Classification via Description from Large Language Models,” our experiments also shed light on the advantages of harnessing the descriptive insights generated by language models. Instead of relying on external models such as GPT-3, however, we opt to leverage LLaVA’s capabilities for both description generation and classification. This further highlights LLaVA’s multimodality and ability to perform a wide variety of vision and language tasks.

3. Methods

Our evaluation procedure for LLaVA consists of: inference, extraction, and matching. We query the model with an image and prompt during inference, then extract the relevant elements from the model’s output which is matched to the closest class label for accuracy calculation. We describe the steps in detail below, and the overall flow is illustrated in Figure 2.

![Figure 2](image.png)

**Figure 2**: We begin by querying LLaVA with an image and textual prompt, then the extracted part of the output is embedded via CLIP’s text encoder. All the possible labels in the dataset (e.g. ”dog”, ”cat”, etc.) are each embedded as well. This allows LLaVA’s output to be matched to a particular class label, enabling us to evaluate whether the classification is accurate or not.

3.1. Inference

The first step is to query LLaVA with an image and textual prompt. We initiate this with a default prompt that reads: “Give me a one-word label in quotation marks for the foreground object in this image”. By requesting a “one-word label”, we sought to retrieve concise responses from LLaVA, facilitating straightforward comparisons with CLIP. The incorporation of quotation marks in the prompt also streamlines the subsequent extraction process, making it easier to identify and isolate the essential information. Finally, specifying for the ”foreground object” minimized irrelevant answers pertaining to the image background. We present more details with prompt engineering in the Experiments section.

3.2. Extraction

After LLaVA generates its responses, we need to extract the relevant parts of the output that would enable us to evaluate the classification accurately. With our default prompt above, we extract only the tokens contained within quotation marks. However, when the quotation marks are absent in the output, we utilize the NLTK library to identify relevant sequences of adjectives and nouns that pertain to the object being classified. Different extraction methods adopted for different prompts are described in the Experiments section.
3.3. Matching

After extracting LLaVA’s raw label, we find its closest match to the respective datasets’ set of labels. The “closeness” is measured between each text embedding pair with the cosine similarity function. This enables us to gauge the likeness between LLaVA’s generated label and the available dataset labels ([1], [2], [3]).

4. Experimental Results

Experiments encompassed a variety of image classification datasets, including CIFAR-10, CIFAR-100, ImageNet, ImageNet100, and Open Images. For CIFAR-100 and ImageNet100, we extended our evaluation to include superclass accuracy as well.

4.1. Baseline

Our initial assessment centered on evaluating LLaVA’s zero-shot classification performance without any modifications. The prompt: “Give me a one-word label in quotation marks for the foreground object in this image” was determined to be an appropriate starting point, and the rest of the procedure was followed as outlined in Figure 2. The results are presented in Table 1. With even the smallest difference between the two models’ accuracy exceeding 10%, it is clear that alternative approaches need to be considered.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CLIP Accuracy</th>
<th>LLaVA Accuracy</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>92%</td>
<td>78%</td>
<td>14%</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>73%</td>
<td>45%</td>
<td>28%</td>
</tr>
<tr>
<td>CIFAR-100 Superclass</td>
<td>53%</td>
<td>39%</td>
<td>14%</td>
</tr>
<tr>
<td>ImageNet</td>
<td>46%</td>
<td>21%</td>
<td>25%</td>
</tr>
<tr>
<td>ImageNet100</td>
<td>69%</td>
<td>17%</td>
<td>52%</td>
</tr>
<tr>
<td>ImageNet100 Superclass</td>
<td>78%</td>
<td>66%</td>
<td>12%</td>
</tr>
<tr>
<td>Open Images</td>
<td>48%</td>
<td>36%</td>
<td>12%</td>
</tr>
</tbody>
</table>

Table 1: An initial comparison of CLIP vs. LLaVA on zero-shot classification accuracy.

The most common source of error was when LLaVA’s classification was not specific enough for the particular dataset. See Section 4.3 and Figure 7 for a comprehensive discussion of these errors and strategies to resolve them.

Another common source of error was when LLaVA was focusing on the wrong object in the image. Figure 3 shows an example of this, where the model should be correct but is mismeasured as incorrect. We tackled this issue by adding the Open Images dataset, which provides multiple labels per image. As seen in Table 1, LLaVA’s performance in Open Images is relatively competent against CLIP.

Figure 3: An example of a misclassification case where LLaVA focuses on an object different from what we desire.

After establishing the initial state, we explored several ways to improve performance, listed below.

4.2. Prompt Engineering

Simply phrasing the input prompt in different ways affected LLaVA’s classification performance quite drastically. Figure 4 lists some of the prompts we experimented with.

- Prompt 1: "Give me a one-word label in quotation marks for the foreground object in this image"
- Prompt 2: "Give me a one-word label in quotation marks for the foreground object in this image from this list: [label1, label2, ...]"
- Prompt 3: "Give me a one-sentence detailed description about the foreground object in this image"
- Prompt 4: "Fill in the blank: this is a photo of a {}"

Figure 4: Examples of prompts used for inference.

Prompt 1, as previously mentioned in Section 3.2, represents the initial prompt used in our evaluation.

Prompt 2, on the other hand, shares the same structure as Prompt 1, with the addition of explicitly listing all the class names for LLaVA to consider during classification. This approach was implementable for datasets featuring a relatively small number of classes. For these datasets, we observed a 1-2% increase in accuracy. For datasets with a
large number of classes, however, including the list of options seemed to worsen performance by overwhelming the model and triggering behaviors of hallucination.

After running experiments with Prompts 1 and 2, we realized that the “one-word” constraint was actually harming performance by restricting the model from utilizing its description ability advantageously. Taking this into account, we came up with Prompt 3, which asks for a description that would provide enough detail while also keeping the output concise as to not exceed the text encoder’s maximum input token length. Moreover, we modified the extraction procedure described in Section 3.2 by encoding the entirety of LLaVA’s output rather than selecting certain tokens. This led to a significant improvement in accuracy across many datasets.

Finally, Prompt 4 was inspired by a successful strategy employed by CLIP to achieve state-of-the-art accuracies. Throughout their experiments with CLIP, Radford et al. [6] discover that prompting CLIP with “a photo of a {class name}” helped boost its performance. We adopted a similar approach with our 4th prompt, asking LLaVA to fill in the blank with a description of the object. We retained the same extraction method as Prompt 3, i.e. embedded the entire output generated by LLaVA. Figure 5 illustrates an example of how this approach helped LLaVA produce a more precise answer.

![Figure 5](a) Prompt 1  (b) Prompt 4)

Figure 5: LLaVA’s output tended to be more precise with Prompt 4 compared to the initial Prompt 1, leading to a higher accuracy overall.

While prompt engineering demonstrated undeniable potential for elevating LLaVA’s performance, we recognize that it may represent a limited avenue for growth. Nevertheless, the impact was notable, with many datasets witnessing performance gains exceeding 10 percentage points. Figure 6 displays the effect of prompt engineering on classification accuracy with CIFAR-100, which exhibited the most sensitivity to prompt variation.

![CIFAR-100 Prompt Engineering](Figure 6: We observe a 15% point increase in accuracy due to prompt engineering in CIFAR-100. Note that Prompt 2 is excluded since it did not prove helpful for this dataset.

4.3. Hierarchical Classification

The ImageNet dataset presented a unique challenge due to the intricacies of its 1000 class labels. The labels encompassed a multitude of distinctive species, with even a seemingly straightforward object like a snake warranting differentiation across more than 10 types—ranging from thunder snake to ring snake to hognose snake. Aligned with the findings of Wen et al. [7], our own analysis echoed the recognition that while LLaVA performed decently in identifying broader animal types, it encountered difficulties when tasked with classifying the precise species as required by the ImageNet dataset. Figure 7 shows one example of such a case.

![Figure 7](LLaVA Label: large snake  Closest ImageNet label: night snake  Correct Label: sea snake)

Figure 7: An example of a misclassification case where LLaVA struggles with fine-grained classification of ImageNet animals.

After our initial exploration of ImageNet, we shifted our focus to a smaller dataset called ImageNet100. ImageNet100 is a subset of 100 classes selected from ImageNet, all of which are animals. Since there were no pre-
defined superclasses provided for ImageNet100, we crafted our own list of 12 superclasses to categorize the 100 fine-grained classes. Table 1 includes results on these superclasses. While the fine-grained accuracy for ImageNet100 is only 17%, we observe that the superclass accuracy soars to 66%. This stark contrast in performance reiterates the need for an evaluation method tailored to handle these hyperspecific labels.

With this in mind, we devised a 2-stage classification procedure. In this method, LLaVA identifies the broad superclass during the initial inference round. Building upon this, we perform a second inference round to pinpoint the specific subclass within the identified superclass. This yielded a better accuracy of 31% on ImageNet100, an almost twofold increase from the original state of 17% (Table 1).

However, the higher accuracy came with a loss in efficiency. The need for two full inference rounds, coupled with the intermediary steps of superclass matching and question editing, resulted in more than twice the original compute power and time.

To address this challenge, we devised an approach to achieve hierarchical classification in a single inference round. We introduced the prompt: "Consider this list: [list of superclasses], and give me the specific species name in quotation marks for the foreground object in this image". We successfully obtained an accuracy of 30%, on par with the results obtained from the 2-round procedure. This improvement in ImageNet100 classification due to an integration of hierarchical classification and prompt engineering is visualized in Figure 8.

4.4. Model Temperature

The final set of experiments we conducted revolved around the model temperature. Temperature is one of LLaVA's hyperparameters, used to control the amount of randomness and creativity the generated output can have. We suspected that lowering the temperature would mitigate the issue of hallucination and thus improve the accuracy. After experimentation with 3 datasets: CIFAR-10, CIFAR-100, and ImageNet100, our hypothesis was shown to be true. Figure 9 summarizes the results.

![Figure 8: We observe an almost doubled accuracy rate in ImageNet100 due to prompt engineering incorporating hierarchical classification.](image8)

![Figure 9: Though thresholds for exact temperatures vary, lower temperature generally improves accuracy across all datasets.](image9)

5. Conclusion and Future Work

We have investigated the zero-shot capabilities of LLaVA in comparison against CLIP. Although the initial evaluation revealed LLaVA's performance to be significantly lacking, trials with prompt engineering, hierarchical classification, and model temperature paved the way for improvement. See Table 2 for a summary of the LLaVA accuracies obtained from the best combination of prompt and temperature for each dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CLIP Accuracy</th>
<th>LLaVA Accuracy</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>92%</td>
<td>86%</td>
<td>6%</td>
</tr>
<tr>
<td>CIFAR-100</td>
<td>73%</td>
<td>60%</td>
<td>13%</td>
</tr>
<tr>
<td>CIFAR-100 Superclass</td>
<td>53%</td>
<td>50%</td>
<td>3%</td>
</tr>
<tr>
<td>ImageNet</td>
<td>46%</td>
<td>23%</td>
<td>23%</td>
</tr>
<tr>
<td>ImageNet100</td>
<td>69%</td>
<td>30%</td>
<td>39%</td>
</tr>
<tr>
<td>ImageNet100 Superclass</td>
<td>78%</td>
<td>72%</td>
<td>6%</td>
</tr>
<tr>
<td>Open Images</td>
<td>48%</td>
<td>47%</td>
<td>1%</td>
</tr>
</tbody>
</table>

Table 2: The current state of CLIP vs. LLaVA on zero-shot classification accuracy. Though still unable to overperform CLIP, LLaVA has grown since its initial state as shown in Table 1.
In order to further improve LLaVA’s classification performance, we have more ideas for different prompts that would better utilize the model’s fine-grained vocabulary. After all, LLaVA’s unparalleled strength lies in its ability to process and generate knowledge of remarkable depth.

Beyond prompt engineering, we also plan to delve deeper into the raw architecture of LLaVA. This way, we can gain a better perception of the root causes of hallucination and other errors that may be blocking the model from producing desirable output.

Our ultimate goal with this work is to showcase the strengths of visual large language models and explore whether simple vision models like CLIP could be replaced. Beyond the realm of classification tasks, our curiosity extends to a diverse array of vision tasks for a comprehensive understanding of LLaVA’s capabilities.

References


