ALMA MATER STUDIORUM – UNIVERSITA' DI BOLOGNA

Scuola di Ingegneria
Corso di Laurea Triennale in Ingegneria Informatica

Vision Language Models as Image Classifiers: an Experimental Study

Relatore
Prof. Luigi di Stefano

Candidato:
Andrea Ritossa

Sessione: 8 ottobre 2024

Anno accademico: 2023/24

Abstract

This study explores the application of Vision Language Models (VLMs) in the context of image classification, using a subset of the ImageNet dataset and leveraging the BART model to maintain the final output in the closed set of candidate labels. The following research sheds light on challenges and trade-offs in adapting VLMs for image classification, performing three experiments of different complexity: 20-way, 100-way and 1000-way classification. Open-source models, while performing reasonably well in the 20-way and 100-way, fall behind closed-source models in the 1000-way classification. Gwen2-VL-7B achieves 25% accuracy while GPT-4o obtains an impressive 79%, a result on par with the first ResNet. Furthermore, by integrating the results in the pre-existing research the following thesis highlights the importance of optimizing vision encoders within VLMs architectures.

1 Introduction

Background of the Study

The advancements in the field of artificial intelligence (AI) in the last two decades reshaped various sectors and industries. The evolution of AI resulted in a proliferation of algorithms that brought machine learning and deep learning into the field of Natural Language Processing (NLP) and Computer Vision (CV). Although distinct, these two areas have begun to intersect with the introduction of multimodality in large language models, known as Multimodal LLMs (MLLMs). MLLMs are a type of deep learning models that extend the capabilities of LLMs by enabling them to process multimodal information such as texts, images, audios and videos. As shown in figure one the proliferation of Multimodal LLMs has been exponential during the last two years:

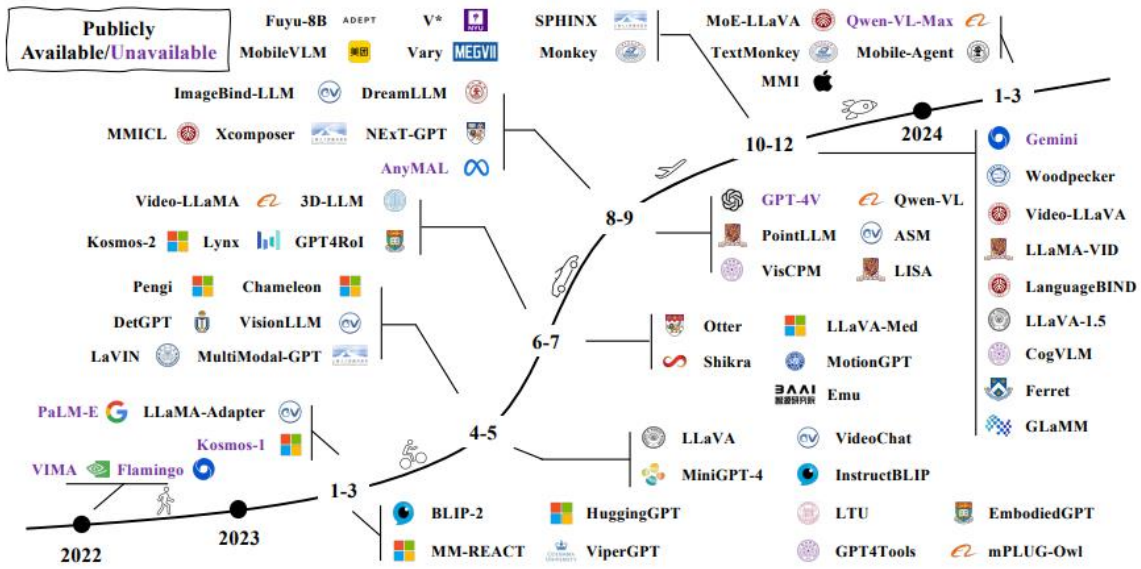


Figure 1: A timeline of representative MLLMs [1].

In context of the intersection of MLLMs with CV this study focuses on models capable of handling vision input: Vision Language Models (VLMs). Recent research has demonstrated the potential of VLMs in various applications, from image captioning to visual question answering. There is a need to understand how these models can be effectively utilized, the challenges that might arise, and the solutions to address these challenges.

Statement of the Problem

Consequently, the objective of this thesis is to explore the capabilities of VLMs by applying them in the domain of *image classification*. While VLMs have demonstrated success in a variety of tasks, their ability to perform well in image classification remains an open question. Therefore, this study seeks to answer:

How well can VLMs, typically perceived as "chatbots", perform in image classification?

By applying VLMs to a classical CV problem, we aim to quantify their ability to understand and categorize images. Image classification involves classifying images into one of several classes, a task that's narrow compared to the broader and more complex capabilities of VLMs. To draw a meaningful comparison, we will evaluate the performance of the VLMs against traditional computer vision methods to fit the results into the broader context of computer vision research.

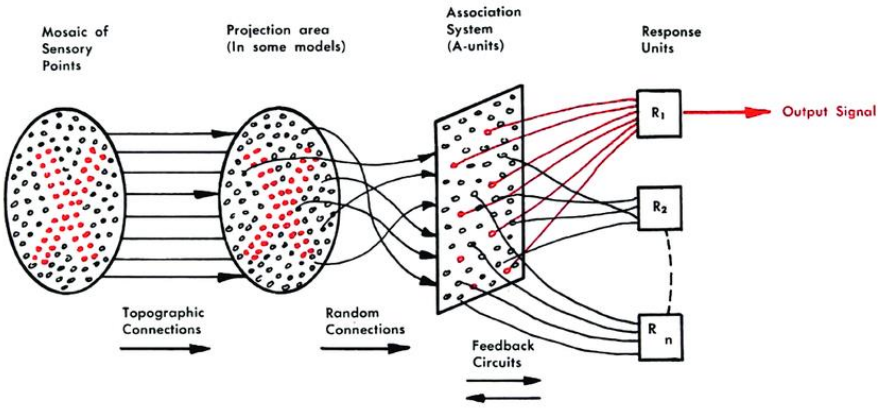
2 Literature Review of Related work

Computer Vision

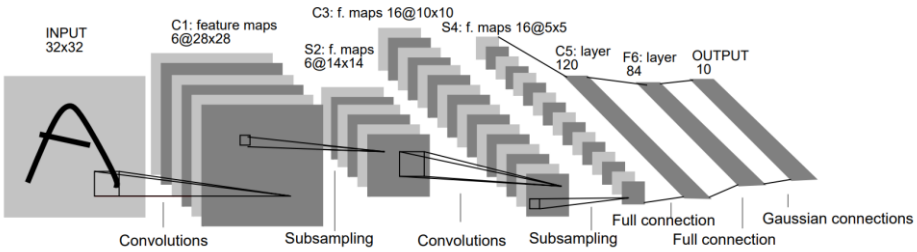
Computer vision has long been focused on image classification tasks, evolving through various methods driven by technological advancements. While neural networks have been the dominant architecture in the past fifteen years, the foundations for these systems were laid decades earlier.

The concept of Neural Networks can be traced back to 1943 with the publication of "*A Logical Calculus of the Ideas immanent in Nervous Activity*" by McCullch and Pitt, where they introduced the first mathematical model of a neural network and established the idea that a network of simple units could perform complex tasks. [21].

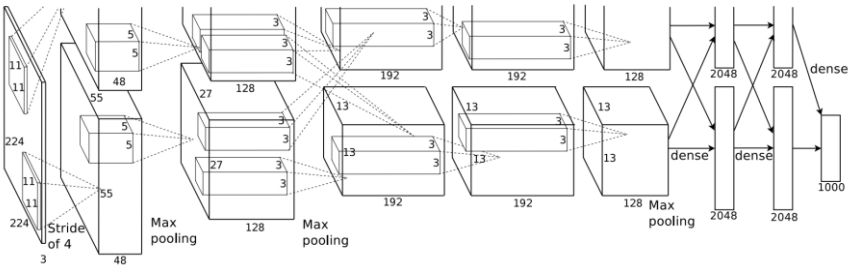
Another foundational contribution came from the Rosenblatt's research around the 1960s with the theorization of the Perceptron model. A neural network that simulated human thought processes. Subsequently the Mark I Perceptron machine was developed, one of the earliest realizations of machines with learning capabilities acquired by trial and error. Therefore it demonstrated the potential for machines to learn and adapt without explicit programming. The image on the right is a representation of Rosenblatt's Perceptron from the "*Design of an Intelligent Automation*", summer 1958. [22]



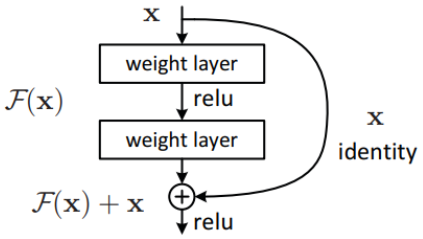
A significant leap forward occurred with Yann LeCun's work at Bell Laboratories in the late 1980s. One of his early innovations was applying the backpropagation algorithm to recognize handwritten digits with the paper "*Backpropagation Applied to Handwritten Zip Code Recognition"* [23]. Furthermore, the 1998 paper *"Gradient-Based Learning Applied to Document Recognition"* introduced Graph Transformer Network (GTN) for handwritten character recognition. The architecture in the paper is a Convolutional Neural Network (CNN), denominated LeNet-5. The research importance relies on its analysis on gradient-descent techniques for training." [24].



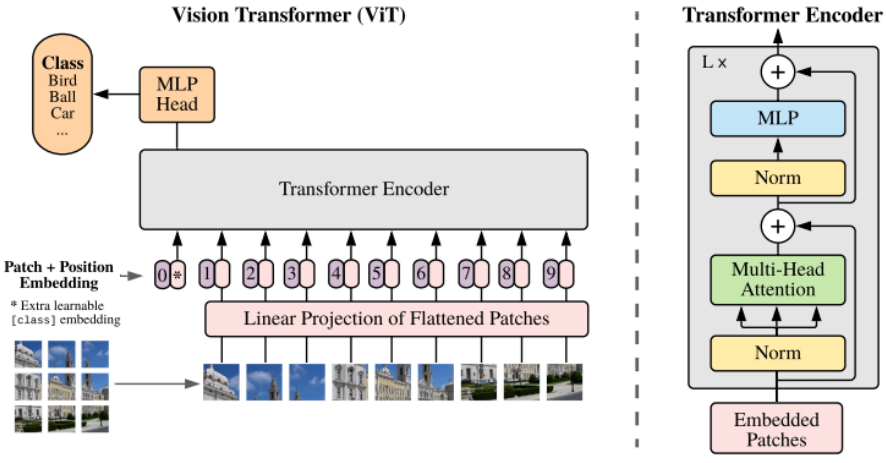
During the early 2000s advances in GPUs allowed for order of magnitude of improvement in the speed of computations for neural network. This period saw the rising of CNNs for image recognition and its most representative moment was with "AlexNet" in 2012; its architecture is described in the following image taken from the AlexNet paper. This deep CNN won for different image competitions and achieved the state of the art in multiple image databases. Notables are its results in the ImageNet [2] in the LSVRC-2010 contest with a great margin from the previously best performant model. It achieved top-1 37.5% and top-5 17.0% error rates, demonstrating such a superior ability in the task that further catalysed the widespread adoption of deep learning models in computer vision. [25]



The next milestone achieved for vision through deep learning came with the introduction of Residual Networks (ResNet) in 2016. The breakthrough relies in addressing the degradation problem during training of deeper neural networks with the introduction of a deep residual learning framework based on explicitly fitting a residual mapping into the layers. The building block is depicted on the paper's figure in the right. The model obtained with the first place on the ILSVRC 2015 classification task and by achieving a 3.57% error on the ImageNet test set. [17]



The Transformers in Vision (ViT), introduced in the paper *"An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale"* a new architecture that successfully operated for vision. Transformers were introduced for Natural Language Processing (NLP) and emerged later in computer vision. Instead of using convolutions to detect patterns ViT split an image into fixed-size non-overlapping patches and treats each patch as a token. The paper's image above describes the model overview and gives a visual representation on how patches are linearly embedded and fed to a standard Transformer encoder with its position embeddings. The Transformer processes these inputs using multi-head self-attention, which enables the model to learn global relationships across the image. After pretraining on a large-scale dataset, the model improved several benchmark results, for example reaching 88.55% accuracy on ImageNet [18]

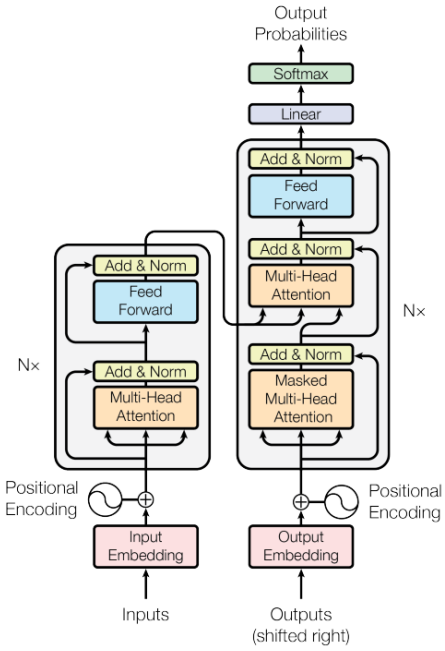


Large Language Models

Large Language Models (LLMs) are deep learning models designed to address complex NLP tasks by training on vast amounts of text data. These algorithms aim is to model language with human fluency for a various array of tasks such as understanding, generation, translation and interaction. LLMs significantly advanced the field of NLP by allowing machines to comprehend and generate language with increasing accuracy.

Before the introduction of the Transformer architecture, the dominant models in NLP were RNNs and long short-term memory networks (LSTMs). However these previous models struggled with several key limitations. RNNs suffered from vanishing gradient problems which made it difficult to capture long-range dependencies. LSTMs, while improving RNNs, still struggled with efficiently capturing long term distances relationships due to the sequential nature of their processing.

In 2017, the paper "*Attention is All You Need*" [26] introduced a new architecture based solely on attention mechanisms: the Transformer. Dispensing recurrence and convolutions, the model relies entirely on attention mechanisms to model relationships between words in a sentence. The multi-head attention mechanism allowed the model to capture long-term dependencies by dynamical weighting of the importance of words in relation to one other. Additionally, positional encoding was introduced to preserve the order of words in the sequence. Importantly, the Transformer's parallelization abilities during processing enabled computational scalability and efficiency. These advantages led to SOTA performance in machine translation tasks, such as the WMT 2014 English-to-German and English-to-French. The paper's picture on the right describes the model architecture.



Building on the Transformer, in 2018 OpenAI introduced GPT-1 in the paper "*Improving Language Understanding by Generative Pre-Training*" [27]. GPT-1 employed a two-stage training process: first unsupervised pre-training on a large amount of unlabelled text and secondly supervised fine-tuning on task-specific datasets. The unsupervised pre-training enabled the model to learn a broad understanding of language while the fine-tuning stage optimized for downstream tasks such as textual entailment, question answering, semantic similarity assessment, and document classification. GPT-1 with its 117 million parameters proved that generative models could generalize across multiple tasks by achieving SOTA on multiple NLP benchmarks.

In 2019, OpenAI released "*Language Models are Unsupervised Multitask Learners*" [28], showcasing the power of unsupervised learning on large scale. With 1.5 billion parameters, the model GPT-2 exhibited zero-shot learning shot capabilities, meaning it could perform tasks it had not explicitly

trained for, just based on understanding natural language prompts. The paradigm shifted again, from creating task-specific models for different NLP tasks towards the creation of a single general-purpose model, that could perform well on a diverse range of tasks without any specific fine-tuning.

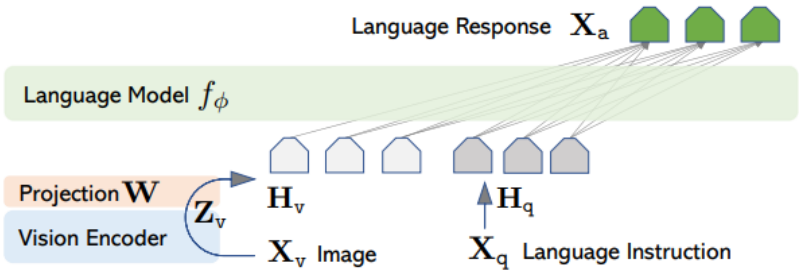
In 2020 OpenAI extended this approach in the paper "*Language Models are Few-Shot Learners*" [29] with GPT-3 (175 billion parameters). The result was a model with increased abilities in language generation, understanding and reasoning capabilities setting a new standard for LLMs. In addition to traditional NLP tasks, GPT-3 demonstrated capabilities in code generation, question answering and even creative writing.

Again in 2023 OpenAI introduced GPT-4, but without disclosing key architectural detail, including model size, hardware specifications, training compute, dataset construction, or specific training methods. [4] The race to enhance LLMs has intensified, with notable contributions to from open-source initiatives such as Meta's LLaMA (Large Language Model Meta AI). [30]

Vision Large Language Model

Vision Language Models (VLMs) represent a new class of models that leverage both visual and language capabilities to process both images or videos and text.

A foundational paper in the space is "*Learning Transferable VIsual Model From Natural Language Supervision*" (2021) [19] which addressed the challenge of integrating images and text by learning joint embeddings. The thesis was to overcome the fixed set of predetermined object categories (typical of computer vision) by exploring learning from raw text about images. In the model CLIP, both an image encoder and a text encoder are trained to predict the correct image-text pairings from a batch of data, using contrastive learning. The image encoder transforms images into feature vectors, while the text encoder processes natural language descriptions into text embeddings, accounting for a dynamical set of object categories. The breakthrough lies in the effectiveness of using natural language as supervision for visual task, achieving some zero-shot capabilities. But the paper also underlines the model limitations, such as weak zero-shot capabilities on tasks such as classification of model of cars, species of flowers and more abstract tasks.



In 2023 the paper "*Visual Instruction Tuning*" introduced the model Large Language and Vision Assistant (LLaVA) that combined visual understanding with language processing capabilities with an architecture that integrates a pre-trained frozen CLIP ViT-L/14 vision encoder with a Vicuna language model using a simple projection matrix as shown in the figure above. The paper considers a two-stage instruction tuning procedure: pre-training the projection layer for alignment of features and then fine-tuning the projection matrix and the LLM. This architecture succeeds in a wide range of multimodal tasks excelling in benchmarks such as visual question answering and visual dialogue and has been applied in many fields that need conversational interfaces with visual understanding, such as education and healthcare. [6]

The Qwen-VL model, introduced in the 2023 paper "*Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond*" [31] that with proposal to improve integration between the vision and language components: a different cross-modal alignment

strategy and a training procedure end-to-end that included the vision encoder, differently from LLaVA.

3 Methodology

The task of image classification consists in labelling input images. The elements involved for such a task are the dataset, consisting of images paired with labels and a computer vision algorithm that connects the visual input the candidate labels with the actual inferred prediction of the image's label. Where the candidate labels are all the possible categories fed to the model along with the image to be classified. Each of these elements is subject to several trade-offs and the one made for the study will be discussed in this chapter. ImageNet was the chosen dataset for its general-purpose scope. Though the study was limited to a subset of the validation split of 10000 images due to computational constraints. Both closed-source models, like GPT-4 and Claude 3 Haiku, and open-source alternatives, such as LLaVA and Phi-3Vision, were evaluated to provide a balanced comparison, with trade-offs made between model performance, cost, and accessibility.

Dataset Selection

As the study focusses on the ability of image understanding of a model, the aspiration for the dataset is to be diverse and to contain various real-world subjects, so to evaluate how well VLMs understand context and can classify a broad range of objects. After investigating several datasets, including Caltech 101, I selected ILSVRC 2012 [2], commonly known as 'ImageNet' as the most appropriate dataset for this task. This dataset spans 1000 object classes such as animals, vehicles, objects, and more, making it a widely used benchmark in computer vision. The Image Net trade-off can be explained in the following table:

Pros	**Cons**
Diverse classes it includes a wide array of real-world objects, which tests the model's understanding across various contexts.	Size The full dataset is computationally expensive to process, requiring significant resources for both storage and inference.
High-quality all images are carefully labelled and curated	Overhead with 1000 categories and a million images running inference is highly resource intensive.
Widespread use given its established role in computer vision research, it provides a robust baseline for comparing results	

Given the economic limitations of this study, I opted to work with a subset of 10,000 images, from the validation split of the dataset, distributed across the whole 1,000 categories for open-source model and a subset of 1600 images for the closed source models. This limited size is a weakness as it might introduce misrepresentation of certain labels, but it grants computational feasibility and maintains enough elements across all the classes of the dataset.

Note that all the images used for evaluation of the models belong to the validation split of the ImageNet 1k dataset and not to the test split due to policy of not divulgation of the test split classes enforced by the dataset creators.

The dataset was accessed through Hugging Face's interface using the following code snippet: [3]

```
from datasets import load_dataset # HF library

imageNetValidation = load_dataset("imagenet-1k", split='validation', streaming=True,
token=hf_token, trust_remote_code=True)

int2str = imageNetValidation.features["label"].int2str
```

The ImageNet dataset includes multiple terms for each class. Due to the limited context window, to ensure that the models could process the labels I simplified them by using only the first item in the full annotation. It's important to note that the simplification was consistent throughout the study across all the different tests.

The code snipped below demonstrates the label simplification process:

```
full_annotation = "grey whale, gray whale, Eschrichtius gibbosus, Eschrichtius robustus"

label = full_annotation.split(',')[0]

# Output: "grey whale"
```

Model Selection

The field of Vision Language Models is fast-evolving, and new state-of-the-art models emerged throughout the study. I continuously assessed the performance of closed-source and open-source models to provide a balanced evaluation.

Closed-source models:

- **GPT-4 Vision** (OpenAI) [4], **GPT-4o** and **GPT-4o-mini** (OpenAI) [32]: Although known for its high accuracy, I limited its usage to 1,600 images due to the high cost of running inferences.
- **Claude 3 Haiku** (Anthropic) [5]: Selected for its availability and cost. The other Claude models, such as Sonnet and Opus, were not used due to financial constraints.
- **Gemini** (Google): Not available in Europe during the time of this study, so it was excluded.

Open-source models:

- **LLaVA v1.6 7B and 13B** (based on Vicuna) [7]
- **Phi-3Vision 4.2B** [8]
- **Gwen2-VL 2B Instruct and 7B Instruct** [9]

The closed-source models have been the innovation forefront, and these models typically perform better presumably due to greater resources invested in their development. However, their proprietary nature also introduces limitations in terms of accessibility and transparency. Therefore, access to the closed source models was performed through the API of the organization and inference using the model GPT-4 Vision was restricted to a subset of 1,600 images due to economic restrictions

of personal funding.
In contrast open-source models were hosted using Kaggle's free GPU workspace, specifically a P100 GPU. This provided a cost-effective way to perform inference with these models, although it introduced constraints on computation time due weekly usage limits. The models marked with (*) were run using an 8-bit quantization of the model. While this may have impacted performance slightly, it was a necessary trade-off for practical feasibility.

Finally, open-source models have the potential to be fine-tuned for this specific task, which is beyond the scope of this study but could be relevant for future work focused on optimizing VLMs.

Prompt Engineering and Output Processing

One challenge was adapting these VLMs for image classification tasks. Unlike traditional classification models that output a label directly, the models used here are ***Image-text-to-text***, meaning their outputs are natural language descriptions rather than discrete labels. Therefore, I had to employ prompt engineering and design strategies to extract a single class from the models' responses.

Through an iterative approach I shaped the prompt structure. I fine-tuned it to maximize classification accuracy, ultimately following the format:

```
introduction = "This image represents an object that belongs to one of the following classes"
labels_string = ", ".join(random_labels)
question = "What is the most probable class that the object in the image belongs to, given the provided classes?"
prompt = f"{introduction}: {labels_string}. {question}"
```

Generally, the output structure adhered to a standard format, an example from the phi3 Vision model:

Visual Input and True Label	Text Output	Label
 True Label = *Italian greyhound* Candidate Labels (20) = *[..., Italian greyhound, fountain, bow tie, valley, ...]*	*The most probable class that the object in the image belongs to is "Italian greyhound."*	*Italian greyhound*

While some models adhered strictly an output's structure (e.g., GPT-4), others, such as LLaVA 7B, exhibited more variability in their responses. When the model's output did not conform to a standard format, I employed **BART** [10] to constrain the output to the closed set of candidate labels. BART, specifically, the *bart-large-mli* [11] model variant, was integrated using an image classification pipeline. By utilizing natural language inference (NLI), BART could classify a given text by calculating the probability of entailment across multiple candidate labels.

The model's prediction was constrained by selecting the most probable label, as demonstrated below:

```
from transformers import pipeline

bart_model = pipeline("zero-shot-classification", model="facebook/bart-large-mnli", device='cuda')

def bart_classify_string(text, candidate_labels):
    result = bart_model(text, candidate_labels=candidate_labels)
    predicted_label = result['labels'][0]
    return predicted_label
```

For instance, when the output from the model "LLaVA Vicuna 7B" prompts a label that do not belong to the ones provided, such as "bird", BART manages to fit the text back into the closed set of classes provided:

Visual Input and True Label	VLM Text Output	BART Output Label
 True Label = sulphur-crested cockatoo Candidate Labels (100) = *[... sulphur-crested cockatoo, ..., great grey owl, koala, Leonberg, lipstick, pelican, eft, ...]*	*The most probable class that the object in the image belongs to is "bird." The image shows two birds perched on a fence, which is a common behavior for birds.*	*eft*

Evaluation Metrics and Performance Assessment

In the evaluation of this multi-class classification task, I employed several metrics that use the following notation [12]:

- ***TP*** - True Positive are the elements labelled as positive by the model and they are positive.
- ***TN*** - True Negative are the elements labelled as negative by the model and they are negative.
- ***FP -*** False Positive are the elements labelled as positive by the model, but they are negative.
- ***FN*** - False Negative are the elements labelled as positive by the model, but they are negative.

Precision measures the proportion of true positive predictions out of all instances predicted as positive, capturing how well the model avoids false positives:

$$precison = \frac{TP}{TP + FP}$$

Recall (or sensitivity) measures the model's ability to identify all relevant instances, focusing on false negatives:

$$recall = \frac{TP}{TP + FN}$$

Accuracy is a commonly used metric that measures the overall correctness of the model's predictions across all classes:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

F1 Score that combines Precision and Recall into a single metric using the harmonic mean, which is especially useful when Precision and Recall have different values. This provides a balanced measure that accounts for both false positives and false negatives:

$$F1 = 2\,\frac{precision \cdot recall}{precision + recall}$$

4 Results

This section presents the finding from various experiments conducted to assess the performance of VLMs with the methods previously described. Varying the number of candidate labels fed into the models these experiments evaluate VLMs efficacy across scenarios with different complexity. The database sizes were selected based on the available resources.

Vision Language Models Performance

20-way Classification: The first experiment utilized a dataset of 10000 entries with 20 candidate labels. Close source models were not included in this experiment for resource optimization.

Model	Precision	Recall	F1-score	Accuracy
LLaVA v1.6 13b vicuna *	0.85	0.85	0.84	0.85
LLaVA v1.6 7b vicuna *	0.79	0.78	0.78	0.78
phi-3-vision 4.2b	0.79	0.77	0.76	0.77
Gwen2-VL-7b-Instruct *	0.92	0.89	0.89	0.89
Gwen2-VL-2b-Instruct	0.93	0.92	0.92	0.92

100-way Classification: In the second experiment a dataset of 1600 entries was used to evaluate different VLMs in a 100 ways classification task:

Model	Precision	Recall	F1-score	Accuracy
LLaVA v1.6 13b vicuna *	0.58	0.53	0.52	0.53
LLaVA v1.6 7b vicuna *	0.44	0.46	0.42	0.49
Gwen2-VL-7b-Instruct *	0.75	0.74	0.73	0.77
Gwen2 -VL-2b-Instruct	0.81	0.75	0.76	0.75
GPT-4V	0.97	0.93	0.95	0.93
Claude 3 Haiku	0.67	0.63	0.65	0.63

1000-way Classification: For these final settings, its computational cost excluded GPT4-Vision, but the experiment was conducted using GPT-4o. The Gwen2-VL family of models was selected for this experiment due to their superior performance in previous classification tasks. The dataset consisted of 1600 entries.

Model	Precision	Recall	F1-score	Accuracy
Gwen2-VL-7b-Instruct *	0.26	0.24	0.24	0.25

Gwen2-VL-2b-Instruct	0.26	0.21	0.22	0.22
GPT-4o	0.82	0.79	0.79	0.79
GPT-4o-mini	0.67	0.60	0.61	0.60

Note: models marked with (*) were quantized to int 8-bit precision

Computer Vision Methods

To provide a fair comparison, experiments were made using traditional computer vision methods against the same dataset and label set as the VLM experiments.

Model	**Ways**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**Support**
openai/clip-vit-large-patch14 [14]	100	0.94	0.89	0.90	0.89	10000

The code to execute image classification with this model can be found at [13]

Model	**Ways**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**Support**
microsoft/resnet-50	1000	0.96	0.94	0.95	0.94	10000
google/vit-large-patch16	1000	0.82	0.82	0.81	0.81	10000

The code for classification using these models can be found at: resnet-50 [15] | vit-L 16 [16]

Benchmark Results from Literature

To contextualize the study findings, here are summarized benchmark results from previous literature on image classification using the ImageNet dataset.

ResNet - Table from the paper Deep Residual Learning for Image Recognition [17]

	Top-1 error	*Accuracy*
ResNet-34 B	21.84%	78.16%
ResNet-50	20.74%	79.26%
ResNet-101	19.87%	80.17%
ResNet-152	19.38%	80.62%

Error rates % of single-model results on the ImageNet validation set

Vision Transformer - Table from the paper: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale [18]

	ViT-H/14	ViT-L/16	ViT-L/32	ViT-B/16	ViT-B/32

ImageNet Accuracy	88.08%	87.12%	84.37%	84.15%	80.73%

Top1 accuracy (in %) of Vision Transformer on ImageNet

CLIP – Table from the paper: Learning Transferable Visual Models From Natural Language Supervision [19]

CLIP vision encoder	Vit L/14	Vit B/32	Vit B/16	RN101	RN50
ImageNet Accuracy	75.3%	68.6%	63.2%	62.2%	59.6%

Zero-shot performance of CLIP models over ImageNet dataset

5 Discussion

These results highlight a trend: as the complexity of the classification task increases, the performance of all evaluated VLMs decreases significantly.

Results Summary

In the 20-way classification task the open-source models showed strong a performance, suggesting that for simple image classification tasks current VLMs can achieve high accuracy levels. The Qwen model family achieved the highest accuracy with Qwen2-2b-Instruct at 92%, followed by Qwen2-7b-Instryct at 89%.

As the classification task scaled to 100-way classification a noticeable decline occurred across all models with Gwen2-VL stood out for suffering a minimal loss of accuracy compared to the other open-source models. GPT-4V stood out with its 93% accuracy, outperforming all other VLMs. While Claude 3 Haiku lagged significantly behind GPT-4 with a 63% accuracy.

In the 1000-way classification task the Gwen2-VL models achieved only around 25% accuracy, with a drastic drop of their performance. This suggests that that the models struggle to navigate across such a large set of labels. On the other hand, GPT-4o demonstrated a robust performance with an accuracy of 79%, showing resilience to the complexity. With a 60% accuracy GPT-4o-mini fell short compared to its larger counterpart but still managed to overperform the open-source models.

Since GPT-4o, GPT-4V, and Claude 3 are closed-source, their architecture and training details remain undisclosed. Therefore, their opacity limits the analysis of their performance and the possible discussion.[4][5]

Open-Source models: LLaVA and Gwen2-VL

Instead, the comparison between the open-source models LLaVA and Qwen2-VL offers valuable insights. Both are cutting-edge VLMs built on the Vision Transformer (ViT) architecture, yet they demonstrate divergent performance. The phenomenon might be explained by differences in training strategies and architecture.

Although the significant size difference between Qwen2 (675M parameters) and LLaVA (307M parameters) the Vision Transformer (ViT) research suggests that this is not a determinant factor because larger models like ViT-Huge (607M parameters) provide only marginal gains in accuracy compared to smaller models like ViT-Large (307M parameters) on classification tasks such as ImageNet [18].

LLaVA uses a frozen CLIP ViT-L/14 model for visual encoding. In standalone evaluation on a 100-way classification task., CLIP achieves 89% accuracy but when integrated with the Vicuna 13b LLM in LLaVA, performance drops to 53% accuracy. This performance reflects a limitation of LLaVA design, that is consistent with the findings in the LLaVA paper: the frozen CLIP encoder and the use of a simple two-layer MLP projection to connect the vision encoder to the LLM. [6]

Note that CLIP's 89% performance is based on the 100-way classification task performed in this study and presented in the results above. It was chosen to perform this computation to provide a benchmark for fair evaluation.

In contrast, Qwen2-VL represents a more advanced approach. With a larger, trainable vision encoder (675M parameters) Qwen2-VL 7B achieved better results than LLaVA on every experiment performed. The key advantage lies in Qwen2-VL's end-to-end training where the vision encoder is fine-tuned with the rest of the model. As noted in the Qwen2-VL paper [9], fine-tuning all parameters ensures that "the vision encoder can learn more task-specific features", which is crucial for achieving higher accuracy in complex classification task. Since image classification was not explicitly part of the fine-tuning dataset, this research suggests that the model's training also enhanced its general-purpose capabilities.

6 Conclusions

This study investigated the performance of VLMs in image classification tasks of varying complexity, focusing on identifying their strengths and limitations. By evaluating both closed-source and open-source models through 20-way, 100-way, and 1000-way classification tasks, this study provides a deeper understanding of how these models handle classification.

Summary of Findings

The findings from this study demonstrate a key trend: as the complexity of the task increases, most VLMs experience a significant drop in performance. However, models like GPT-4o performed exceptionally well, even in the 1000-way classification task, achieving results comparable to traditional networks like ResNets. GPT-4o's performance in complex classification tasks is noteworthy: it suggests that newer VLMs are closing the gap between vision-language models and traditional computer vision models. This achievement has not been reached by the open-source models, nonetheless the strength of Qwen2-VL over LLaVA suggests that open-source models are catching up with their counterparts.

Recommendations for future research

To improve scalability of VLMs in complex classification tasks future research should focus on how to enhance integration between visual and language information. Promising directions are the research of more sophisticated projection mechanisms and additionally end-to-end training of both vision encoder and the language model as it has shown to enhance performance.

Expanding VLM capabilities into object detection and other vision tasks offers another promising direction. Transfer learning across different vision tasks and from multiple datasets could strengthen the model's generalization capabilities.

This study also opens several research questions that could provide insights into the potential of VLMs:

- How does fine-tuning VLMs specifically for image classification affect performance on image classification benchmarks? And on other benchmarks, such as object detection or multimodal tasks?
- Can fine-tuning the entire model on the same dataset used for training the vision encoder lead to improved synergy between the vision encoder and the LLM?

References

1. A Survey on Multimodal Large Language Models https://arxiv.org/pdf/2306.13549v2.pdf
2. ImageNet website- https://www.image-net.org/
3. ImageNet ILSVRC 2012 hugging face- https://huggingface.co/datasets/ILSVRC/imagenet-1k
4. GPT4 Technical Report- https://arxiv.org/abs/2303.08774
5. The Claude 3 Model Family- https://paperswithcode.com/paper/the-claude-3-model-family-opus-sonnet-haiku
6. Visual Instruction Tuning https://llava-vl.github.io/ https://arxiv.org/abs/2304.08485
7. LLaVA-Next: Improved Reasoning, OCR and world knowledge https://llava-vl.github.io/blog/2024-01-30-llava-next/
8. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone https://arxiv.org/abs/2404.14219
9. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution https://arxiv.org/abs/2409.12191
10. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation and Comprehension https://arxiv.org/abs/1910.13461
11. Hugging Face- facebook/bart-large-mnli https://huggingface.co/facebook/bart-large-mnli
12. Metrics for Multi-Class Classification: an Overview https://arxiv.org/abs/2008.05756
13. Image Classification using CLIP through hugging face https://huggingface.co/tasks/zero-shot-image-classification
14. Hugging Face – openai/clip-vit-large-patch14- https://huggingface.co/openai/clip-vit-large-patch14
15. Hugging Face – microsoft/resnet-50 https://huggingface.co/microsoft/resnet-50
16. Hugging Face – google/vit-large-patch16-224 https://huggingface.co/google/vit-large-patch16-224
17. Deep Residual Learning for Image Recognition https://arxiv.org/abs/1512.03385
18. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale https://arxiv.org/abs/2010.11929
19. Learning Transferable Visual Models From Natural Language Supervision https://arxiv.org/abs/2103.00020 https://openai.com/index/clip/
20. Image Classification Top1 accuracy on ImageNet timeline https://paperswithcode.com/sota/image-classification-on-imagenet
21. A Logical Calculus of the Ideas immanent in Nervous Activity https://www.historyofinformation.com/detail.php?entryid=782
22. McCulloch-Pitts neuron https://www.historyofinformation.com/detail.php?entryid=782
23. Backpropagation Applied to Handwritten Zip Code Recognition https://yann.lecun.com/exdb/publis/pdf/lecun-89e.pdf
24. Gradient-Based Learning Applied to Document Recognition http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf
25. ImageNet Classification with Deep Convolutional Neural Networks https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html
26. Attention is All You Need https://arxiv.org/abs/1706.03762
27. Improving Language Understanding by Generative Pre-Training https://openai.com/index/language-unsupervised/
28. Language Models are Unsupervised Multitask Learners https://openai.com/index/better-language-models/
29. Language Models are Few-Shot Learners https://arxiv.org/abs/2005.14165
30. LLaMA: Open and Efficient Foundation Language Models https://arxiv.org/abs/2302.13971
31. Qwen-VL: A Versatile Vision-Language Model for Understanding, Localization, Text Reading, and Beyond https://arxiv.org/abs/2308.12966
32. OpenAI on GPT-4o- https://openai.com/index/hello-gpt-4o/