CalQAgent: Calorie Estimation via Agentic Tool-Calling and Fine-Tuned *Qwen* Vision Language Model

Prof. Pallavi U
Department of Computer Science and Engineering
Atria Institute of Technology
Bangalore, India
pallaviu78@gmail.com

Mohammed Siddiq
Department of Computer Science and Engineering
Atria Institute of Technology
Bangalore, India
mohammesiddiq78@gmail.com

Sinchana H S
Department of Computer Science and Engineering
Atria Institute of Technology
Bangalore, India
sinchana46400@gmail.com

Sanju J K
Department of Computer Science and Engineering
Atria Institute of Technology
Bangalore, India
sanjujk76@gmail.com

Sneha C
Department of Computer Science and Engineering
Atria Institute of Technology
Bangalore, India
csneha9380@gmail.com

Abstract— *Tracking what people eat and understanding the nutritional content of their meals from photos is becoming more relevant than ever. Still, it's a tough challenge since many dishes look alike, and most existing systems rely on heavy, data-driven pipelines that require a lot of manual labeling. Existing approaches, such as RAG-based models like CaLoRAify, use external retrieval mechanisms to match ingredients and nutrition facts, which can add extra processing time and sometimes lead to errors.*

In this work, we introduce an end-to-end framework built on the Qwen-VL vision–language model that can directly identify foods and extract ingredient details from images. By fine-tuning Qwen-VL and using the Model Context Protocol (MCP), our system produces structured, machine-readable ingredient information — including names, quantities, and units — which can be automatically linked to a nutrition database for calorie and nutrient estimation. With fewer steps and less chance of mistakes, this system works faster and more accurately. Overall, it's a smarter and more straightforward way to understand what's on someone's plate and estimate its nutritional value.

Keywords— *Nutrition, Nutrient Ratio, Carbohydrates, Fats, Lipids, Diet Analysis, Nutri Track.*

I. INTRODUCTION

The global prevalence of obesity has escalated into a critical public health challenge, linked directly to a rise in preventable chronic conditions such as diabetes and cardiovascular disease. In response, the digital health market has seen a proliferation of dietary management tools, with AI-driven platforms like CalAI and MyFitnessPal generating substantial revenue and user engagement. Despite this commercial success, a significant gap remains: the ability to accurately estimate caloric intake from a food image without burdensome user input.

Traditional computer vision approaches to this problem have historically relied on rigid pipelines, often requiring reference objects (e.g., a coin placed next to food) or specialized depth sensors to estimate volume. These methods are practically cumbersome for casual users. More recently, the advent of Large Vision-Language Models (VLMs) like LLaVA and MiniGPT-4 has revolutionized the field, enabling systems to "see" and describe food. However, generalist VLMs notoriously struggle with the precision required for nutritional science; they frequently hallucinate ingredient quantities or invent caloric values when specific data is absent from their pre-training.

To mitigate these hallucinations, recent frameworks such as CaLoRAify introduced Retrieval-Augmented

Generation (RAG) to the domain, aligning visual features with external text databases. While effective, standard RAG implementations are often "passive"—they retrieve documents based on semantic similarity but lack the reasoning capabilities to handle complex, multi-step queries or fallback logic when data is missing.

Addressing this limitation, we present CalQAgent, an agentic vision-language framework designed to transform calorie estimation from a generation task into a multi-step reasoning process. Unlike previous iterations that simply concatenate retrieved text, our system utilizes the state-of-the-art Qwen3-VL-8B backbone orchestrated by the Qwen-Agent framework. This allows the model to function not just as a predictor, but as an intelligent agent capable of executing Function Calls.

By analyzing a single monocular image, CalQAgent first identifies the dish and then autonomously decides which "tool" to employ—querying a vector database for recipe constraints or falling back to the USDA nutritional database for granular ingredient analysis. This active "Tool-Use" paradigm significantly reduces error propagation and ensures that every calorie estimate is grounded in verified structured data rather than probabilistic guesswork.

Our contributions are summarized as follows:

- **Agentic VLM Framework:** We introduce CalQAgent, a system that upgrades food analysis from passive visual QA to active agentic reasoning. By leveraging Qwen3-VL's advanced instruction-following capabilities, we enable high-precision ingredient recognition from standard smartphone images.

- **Function-Calling Methodology:** We propose a novel inference pipeline that replaces standard RAG with dynamic function calling. This allows the system to intelligently route queries between a recipe Vector Database and the USDA nutritional repository, bridging the gap between visual features and quantitative data.

- **Robust Performance & Dataset:** We fine-tuned our model on a curated subset of 170,000 image-text pairs from Recipe1M+, formatted specifically for chat-based interaction. Experimental results demonstrate that our agentic approach significantly outperforms baseline VLMs, achieving an F1 score of 0.84 in ingredient and instruction retrieval.

II. LITERATURE REVIEW

2.1 Evolution of Large Foundation Models

Large Language Models (LLMs) have significantly advanced artificial intelligence by helping machines understand and generate human language more effectively. Early transformer models like BERT (Kenton and Toutanova, 2019) and GPT-2 (Radford et al., 2019) were important because they allowed systems to capture deeper meaning and context from text. Building on this, larger models such as GPT-3.5 (Brown et al., 2020) were developed, capable of strong reasoning and generating text for many different tasks.

Later, smaller and more efficient models like Mistral (Jiang et al., 2023) and Phi-2 (Javaheripi et al., 2024) showed that high performance could be achieved with less computing power. At the same time, multimodal models such as GPT-4 (Achiam et al., 2023) and BARD (AI, 2023) extended LLMs to understand both text and images. However, since these models are closed-source, researchers have limited ability to explore or modify them. This has encouraged the development of open-source alternatives that provide greater flexibility, transparency, and support for academic research.

2.2 Vision–Language Models and Multimodal Learning

Vision–Language Models (VLMs) combine image and text understanding, allowing systems to describe, interpret, and answer questions about visual content. Early models like CLIP (Radford et al., 2021) linked images and text in a shared space, providing a strong foundation for future models.

Later models, such as BLIP-2 (Li et al., 2023a), LLaVA (Liu et al., 2023), and InstructBLIP (Dai et al., 2023), improved this approach by connecting visual encoders and language models more tightly, making it easier for the system to understand and describe images accurately. These models can now describe images, explain scenes, and follow user instructions related to visual content.

VLMs have been successfully applied in fields like healthcare (Moor et al., 2023), finance (Wu et al., 2023), and law (Dahl et al., 2024). However, using them for food analysis remains challenging because many food items look similar and there are not enough specialized datasets to train the models effectively. This makes accurate ingredient recognition and calorie estimation a continuing challenge for research.

2.3 Traditional Calorie Estimation Methods

Before the rise of multimodal models, calorie estimation from food images was done through a series of steps such as food identification, portion size estimation, and calorie calculation [10], [18], [21], [24]. These methods often used reference objects or depth information to measure food portions more accurately. Although these methods worked well in controlled environments, they were hard to use in everyday situations.

A key problem was that these methods needed extra information, like reference objects or special camera inputs, which most users do not have. Another issue was that each step—such as segmentation, classification, and volume estimation—was done separately, so small mistakes in one step could affect the next. In addition, these systems often required advanced hardware, like depth or multi-view cameras, making them expensive and less practical for mobile devices or low-resource environments.

2.4 Multi-Modal Large Language Models for Food

Recent advances in multimodal large language models (LLMs) have shown that they can process both text and images effectively, opening new possibilities for food-related research [28]. Models like LLAVA-Chef [23] use multimodal inputs such as dish titles, ingredient lists, and images to predict recipes. This demonstrates how combining visual and textual information can improve food understanding and analysis.

However, LLAVA-Chef mainly focuses on generating recipes instead of estimating ingredient amounts or calorie values directly from food images. For practical use, especially in mobile health and diet tracking, it is better to have a system that can analyze a food image and provide nutritional information directly. Our approach does this by predicting ingredients and calories from a single image, making the process faster and easier for everyday use.

2.5 Fine-Tuning Strategies for Large Models

Fine-tuning techniques are important for adapting large models to perform specific tasks. Methods such as Low-Rank Adaptation (LoRA) [12] and prompt tuning help improve model performance efficiently without requiring large computational resources. Research like LIMA [31] has shown that even with a small amount of training data, pre-trained models can be fine-tuned effectively while maintaining their general reasoning ability.

However, vision-language models need more careful tuning because they must understand both visual and textual information at the same time. In this work, we apply fine-tuning strategies like LoRA to the Qwen-VL model to make it suitable for food image analysis. The main goal is to improve the model's ability to detect food items, identify ingredients, and accurately estimate nutritional values directly from images.

III. DATASET

We constructed our primary dataset by curating a substantial subset of the Recipe1M+ corpus. The final collection comprises 190,000 samples, which were stratified into a training set of 170,000 image-text pairs and a test set of 20,000 samples. Unlike traditional classification datasets, we structured the data into a conversational JSON format pairing visual inputs with user instructions—to specifically fine-tune the model for recipe title identification.

IV. METHODOLOGY

We propose *CalQAgent*, a framework designed to overcome the limitations of standard generative models in the domain of nutritional analysis. While foundational VLMs possess strong visual generalization, they often struggle with "hallucinations" where the model invents plausible but incorrect nutritional data and lack mathematical precision required for calorie counting. To address this, we move beyond static Retrieval-Augmented Generation (RAG) from the paper *CaLoRaify* by adopting the **Qwen-Agent** framework, effectively treating the VLM as an active controller rather than a passive text generator.

Our methodology decouples visual recognition from factual calculation. We first fine-tune a Qwen-based model (specifically Qwen3 8B) on our image-text pairs to act as a high-precision visual encoder, strictly optimizing it to identify recipe titles from visual inputs. Once the dish is identified, the model utilizes dynamic *function calling* to interface with external tools rather than relying on its internal weights for facts.

The agent is programmed to autonomously execute a multi-step workflow: first querying a vector database to retrieve the exact ingredient list for the predicted title, and subsequently accessing a USDA-integrated tool to fetch and sum standardized nutritional values. This modular architecture not only significantly reduces error propagation by grounding outputs in verified database entries but also offers superior extensibility.

Unlike fixed RAG pipelines, the tool-use approach allows for the seamless integration of downstream applications, such as real-time user dietary history tracking, retrieving extra information in regards with the recipe such as recipe instructions, setting dietary plans etc., without requiring structural changes to the core model.

4.1 Preliminaries

4.1.1 Qwen3 VL

In this work, we utilize Qwen3-VL-8B-Instruct as our primary vision-language backbone. As the latest iteration in the Qwen series, this model offers a substantial upgrade in multimodal reasoning, designed to function seamlessly as a visual agent. Unlike traditional VLMs that may treat vision and text as loosely coupled modalities, Qwen3-VL achieves a lossless unification of text and vision, delivering text understanding capabilities on par with pure Large Language Models (LLMs).

The model architecture incorporates several key enhancements critical for fine-grained image recognition. It employs Interleaved-MRoPE, a robust positional embedding mechanism that allocates frequency over time, width, and height, facilitating superior spatial and temporal reasoning. Furthermore, the architecture utilizes DeepStack, a method that fuses multi-level features from the Vision Transformer (ViT). This allows the model to capture intricate visual details—such as specific food textures or portion sizes—and sharpens the alignment between the image and the generated text.

We selected Qwen3-VL specifically for its unified "Visual" and "Agentic" capabilities. The model is optimized to recognize elements, understand functions, and invoke external tools, which is foundational to our pipeline's retrieval mechanism. Additionally, with a native context length of 256K, the model demonstrates exceptional robustness in processing complex recipe data. To ensure the best quality while mitigating the hallucination risks common in generative tasks, we leverage the model's intrinsic support for function calling to ground its outputs in external data.

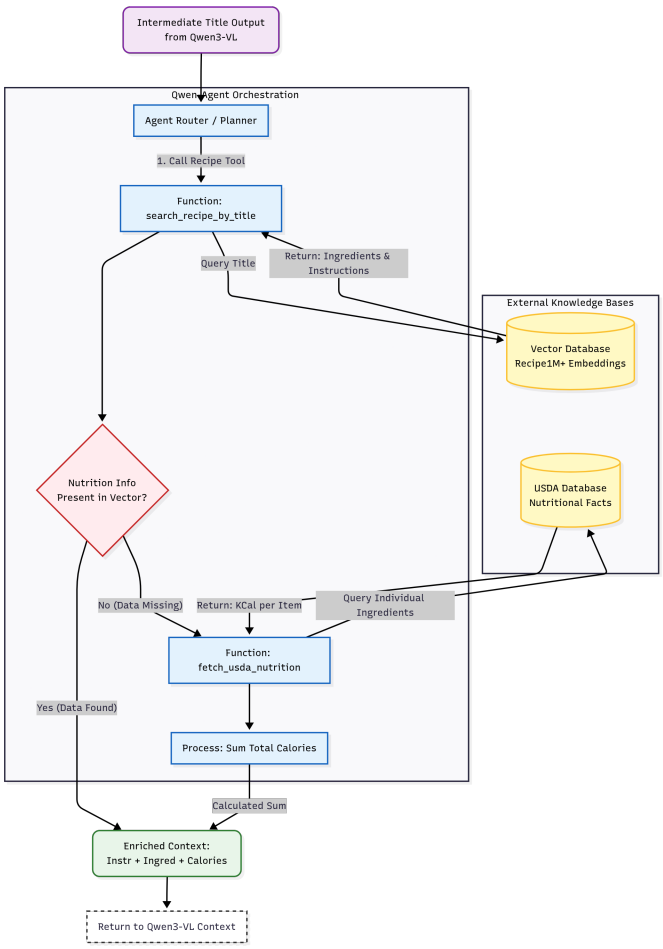

Figure 1: Architectural overview of the CalQAgent function-calling mechanism using Qwen-Agent framework.

4.1.2 Qwen-Agent

To operationalize the Qwen3-VL backbone, we employ the Qwen-Agent framework. This framework serves as a modular orchestration layer, transforming the model from a passive text generator into an active agent capable of multi-step planning and complex reasoning. Rather than relying solely on pre-trained weights, the agent framework equips the model with working memory and the ability to maintain context across extended interactions.

The critical feature utilized in this study is the framework's robust function calling (or tool use) capability. This mechanism allows the model to act as a logic router: when the model recognizes that a query requires external data or specific actions, it generates a structured call to a defined function rather than hallucinating an answer. This architecture is highly extensible; it enables the system to leverage a wide array of third-party applications and custom microservices ranging from code interpreters to database management systems. By decoupling the reasoning engine from the data execution layer, we can seamlessly integrate diverse tools, such as vector search and nutritional databases, directly into the inference pipeline.

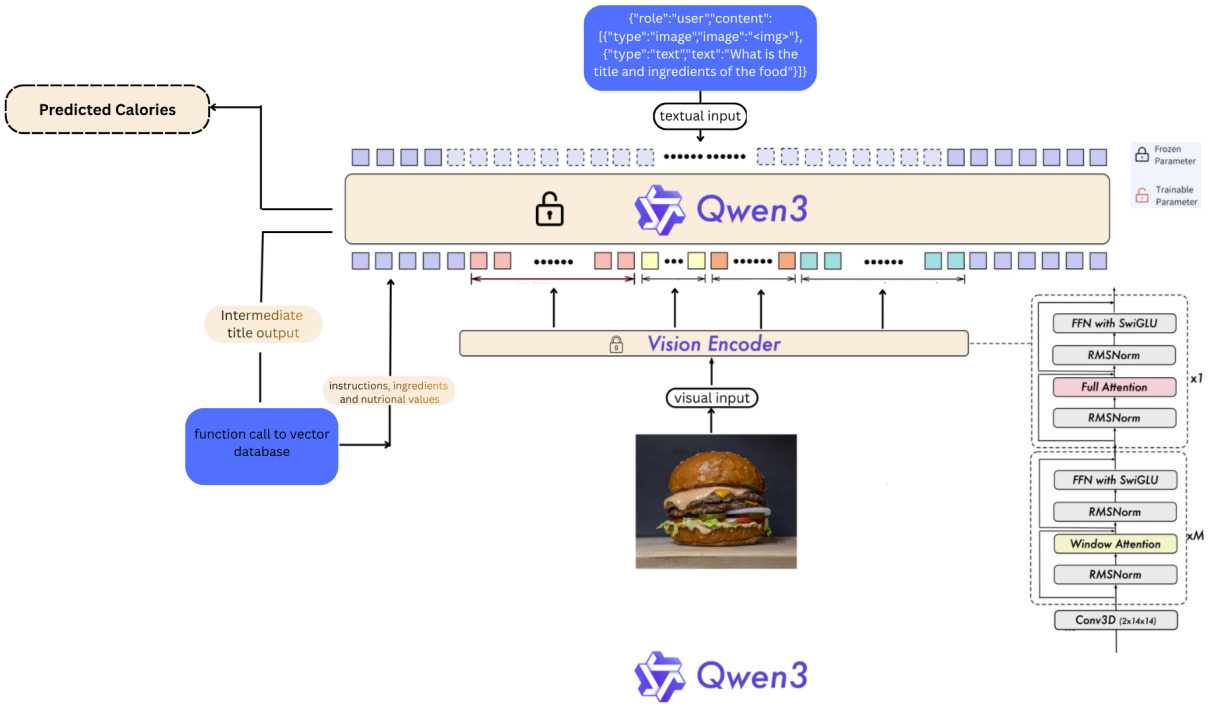

Figure 2: The workflow initiates with the frozen Vision Encoder (detailed on the right), which processes the input food image using a stack of Conv3D layers, Window Attention, and Full Attention blocks to extract rich visual representations. These visual tokens are concatenated with the user's textual instruction and fed into the Qwen3 backbone. Leveraging its agentic capabilities, the model first predicts an intermediate dish title, which triggers a function call to the external vector database. The database retrieves precise instructions, ingredients, and nutritional values, which are returned to the model context. Finally, Qwen3 integrates the retrieved structured data with the initial visual features to generate the final Predicted Calories.

4.2 CalQAgent

CalQAgent presents a framework that advances beyond traditional retrieval-augmented strategies by integrating the Qwen3-VL backbone with an agentic orchestration layer. Unlike standard pipelines that rely on static embedding lookups, our approach treats calorie estimation as a multi-step reasoning task, leveraging the model's intrinsic function-calling capabilities to dynamically interact with external knowledge bases (see Figure 2 for the complete architectural schematic).

As illustrated in Figure 2, the inference process begins with a high-resolution food image fed into the Qwen3 Vision Encoder. These visual features are fused with the user's textual prompt (e.g., "What is the title and ingredients?") and processed by the large language model. Instead of attempting to hallucinate the nutritional content directly from its pre-trained weights, the model is fine-tuned to first identify the Dish Title with high pre-

cision.

This predicted title serves as the trigger for our agentic workflow. Utilizing the Qwen-Agent framework, the system executes a specific function call to our external Vector Database. This tool retrieves the exact recipe constraints, including the ingredient list and preparation instructions associated with the identified title.

Crucially, the CalQAgent method incorporates a logical fallback mechanism for nutritional calculation. The system first checks the retrieved vector data for pre-existing caloric values. If this specific metadata is absent, the agent autonomously triggers a secondary function call to the USDA Nutritional Database. This allows the model to fetch granular nutritional values for individual ingredients, aggregate the total calories, and generate a final, verified response. This structured tool-use significantly reduces the error propagation common in end-to-end generation, ensuring the output is grounded in verified database entries rather than probabilistic guesswork.

V. EXPERIMENT

5.1 Experimental Setup

All experiments were conducted on a high-performance computing server equipped with an NVIDIA A100 GPU (40GB VRAM). The end-to-end fine-tuning process required approximately 10 hours to complete. To balance training stability with memory constraints, we configured the batch size to 8 per GPU.

For optimization, we utilized a standard AdamW optimizer with an initial learning rate of 2×10^{-4}. We employed a linear learning rate scheduler to gradually decay the rate throughout the training process. To ensure parameter-efficient fine-tuning, we applied Low-Rank Adaptation (LoRA) with a rank (r) of 64 and a scaling alpha (α) of 16. This configuration allowed us to effectively adapt the Qwen3-VL backbone to the food domain without the computational overhead of full-parameter training.

5.2 Input Format & Dataset Construction

We curated a custom dataset subset derived from Recipe1M+, selecting 170,000 samples for training and 20,000 samples for testing. Unlike previous approaches that rely on linear text concatenations, we structured our data into a conversational JSON format to leverage Qwen3-VL's chat capabilities.

Each input sample encapsulates the visual data and the user query, while the output target contains the comprehensive recipe details (Title, Instructions, and Ingredients). The data is formatted as follows:

```
[
  {
    "role": "user",
    "content": [
      { "type": "text",
        "text": "<Instruction_Prompt>" },
      { "type": "image",
        "image": "<Image_Path>" }
    ]
  },
  {
    "role": "assistant",
    "content": [
      { "type": "text",
        "text": "<Dish_Title_and_Details>" }
    ]
  }
]
```

In this structure, `<Instruction_Prompt>` represents queries such as "Identify this dish and list its ingredients," and the assistant's response provides the ground truth text used for supervised fine-tuning.

5.3 Metrics Results

To rigorously evaluate the semantic quality of our model's generations, we utilized BERTScore. Unlike traditional n-gram metrics (e.g., BLEU or ROUGE) that penalize valid paraphrases, BERTScore computes the cosine similarity between the contextual embeddings of the candidate and reference text. This is particularly critical for recipe generation, where "1 cup of chopped onions" and "finely diced onion, one cup" are semantically identical despite lexical differences.

We evaluated performance across two distinct tasks: Dish Title Prediction (the model's ability to recognize the food) and Instruction & Ingredient Retrieval (the accuracy of the agentic function call).

Table 1: Dish Title Prediction Performance

Metric	**Baseline (Zero-shot)**	**Fine-tune (CalQAgent)**	**Improv.**
BERTScore (P)	0.6458	0.7919	+22.6%
BERTScore (R)	0.5831	0.7518	+28.9%
BERTScore (F1)	0.6012	0.7759	+29.0%

As shown in Table 1, our fine-tuned Qwen3-VL backbone demonstrates a significant improvement in visual recognition capabilities. The F1 score increase from 0.6012 to 0.7759 indicates that the model learned to map visual features to specific culinary terminology much more effectively than the baseline.

Table 2: Instruction and Ingredient Match (Function Call Output)

Metric	Baseline (Zero-shot)	Fine-tune (CalQAgent)	Improv.
BERTScore (P)	0.4210	0.8559	+103.3%
BERTScore (R)	0.3819	0.8207	+114.9%
BERTScore (F1)	0.4113	0.8469	+105.9%

Table 2 highlights the decisive advantage of our agentic approach. The baseline model, attempting to hallucinate recipes directly from weights, scored poorly (F1: 0.4113). In contrast, CalQAgent, leveraging the function call to the vector database, achieved a remarkable 0.8469 F1 score. This doubling in performance confirms that decoupling reasoning (identifying the dish) from knowledge retrieval (fetching the recipe) is the superior strategy for complex information extraction.

VI. CONCLUSION AND FUTURE WORK

In this work, we introduced CalQAgent, a novel agentic framework for accurate food calorie estimation. By fine-tuning the Qwen3-VL-8B backbone and integrating it with the Qwen-Agent framework, we successfully shifted the paradigm from passive visual question answering to active, multi-step reasoning. Our approach addresses the critical "hallucination" bottleneck in dietary AI by grounding outputs in verified external databases. The experimental results demonstrate that our method not only recognizes diverse food items with high precision but also retrieves granular nutritional data with an F1 score of 0.84, significantly outperforming non-agentic baselines.

Looking forward, we aim to expand this research in four key directions:

- **Dataset Expansion and Curation:** While our current model was trained on a 170k subset, we plan to scale fine-tuning to the full Recipe1M+ dataset. This expansion will expose the model to a wider variety of global cuisines and rare ingredients. We will also implement a more rigorous data processing pipeline to filter noisy samples, ensuring the model learns from high-quality "visual-text" pairs to improve generalization.

- **Enhanced Function Calling Features:** We intend to develop a richer suite of agentic tools. A primary focus is the integration of a "User Dietary History" database function. This would allow the model to not only estimate calories for a single meal but also log these values into a longitudinal user profile, enabling the system to provide personalized dietary recommendations and visualize trends via a dashboard analysis UI.

- **Cross-Model Benchmarking:** To validate the robustness of our architecture, we plan to benchmark the CalQAgent framework across different models (e.g., LLaVA-Next, DeepSeek-V3, GLM-4, Gemma, etc.) using the exact same dataset. This comparative analysis will help isolate the performance gains attributed to our specific fine-tuning strategy versus the inherent capabilities of the backbone model.

- **Methodological Comparison:** We aim to conduct a comparative study against efficient but less structured approaches, such as pure Vision Transformers (ViT) and CLIP-based classifiers. While CLIP excels at zero-shot classification, it often struggles with fine-grained details and lacks the ability to generate structured, long-form text like recipes. Similarly, standard ViTs often fail to capture local textures crucial for distinguishing similar foods (e.g., distinguishing between varying sauce consistencies). By contrasting these methods, we hope to quantitatively demonstrate the superiority of an agentic VLM approach for tasks requiring deep semantic understanding and structured data retrieval.

REFERENCES

[1] S. Author *et al.*, "CaLoRAify: Calorie Estimation with Visual-Text Pairing and LoRA-Driven Visual Language Models," *arXiv preprint arXiv:2412.09936*, 2024. https://doi.org/10.48550/arXiv.2412.09936

[2] Qwen Team, "Qwen3 Technical Report," *arXiv preprint*, 2025.

[3] QwenLM, "Qwen-Agent," GitHub Repository. https://github.com/QwenLM/Qwen-Agent

[4] Qwen Team, "Qwen2.5-VL Technical Report," *arXiv preprint arXiv:2505.09388*, 2025. https://doi.org/10.48550/arXiv.2505.09388

[5] "Ai-based calorie estimation app CalAI reports $50 million revenue," Business Insider. https://www.businessinsider.com/calai-revenue-growth-2023. Accessed: 2024-10-11.

[6] "LoseIt app reaches new milestone in calorie tracking," LoseIt. https://www.loseit.com/news/milestone. Accessed: 2024-10-11.

[7] "MyFitnessPal user statistics," MyFitnessPal. https://www.myfitnesspal.com/statistics. Accessed: 2024-10-11.

[8] National Institutes of Health, "Overweight & Obesity Statistics," NIDDK. https://www.niddk.nih.gov/health-information/health-statistics/overweight-obesity. Accessed: 2024-10-11.

[9] USDA, "FoodData Central Calorie Database," U.S. Department of Agriculture. https://fdc.nal.usda.gov/fdc-app.html. Accessed: 2024-10-10.

[10] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to retrieve, generate, and critique through self-reflection," *arXiv preprint arXiv:2310.11511*, 2023.

[11] J. Chen, D. Zhu, *et al.*, "MiniGPT-v2: Large language model as a unified interface for vision-language multi-task learning," *arXiv preprint arXiv:2310.09478*, 2023.

[12] J. Chen, F. Wei, J. Zhao, S. Song, B. Wu, Z. Peng, S.-H. G. Chan, and H. Zhang, "Revisiting referring expression comprehension evaluation in the era of large multimodal models," *arXiv preprint arXiv:2406.16866*, 2024.

[13] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[14] T. Ege, Y. Ando, R. Tanno, W. Shimoda, and K. Yanai, "Image-based estimation of real food size for accurate food calorie estimation," in *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 274–279, 2019.

[15] R. Hou, H. Chang, B. Ma, S. Shan, and X. Chen, "Cross attention network for few-shot classification," in *Neural Information Processing Systems*, 2019.

[16] E. J. Hu *et al.*, "LoRA: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[17] D. A. Hudson and C. D. Manning, "GQA: A new dataset for real-world visual reasoning and compositional question answering," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6693–6702, 2019.

[18] V. Karpukhin *et al.*, "Dense passage retrieval for open-domain question answering," *arXiv preprint arXiv:2004.04906*, 2020.

[19] J. Lai, S. Yang, J. Zhou, *et al.*, "Clustered-patch element connection for few-shot learning," in *International Joint Conference on Artificial Intelligence*, 2023.

[20] M. Lewis *et al.*, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Annual Meeting of the Association for Computational Linguistics*, 2019.

[21] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," *arXiv preprint arXiv:2005.11401*, 2020.

[22] Y. Liang and J. Li, "Computer vision-based food calorie estimation: dataset, method, and experiment," 2017.

[23] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," 2023.

[24] X. Ma, Y. Gong, P. He, H. Zhao, and N. Duan, "Query rewriting for retrieval-augmented large language models," *arXiv preprint arXiv:2305.14283*, 2023.

[25] B. Magid *et al.*, "CalorieMe: An image-based calorie estimator system," in *2023 Eleventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp. 555–560, 2023.

[26] J. Marin, P. Karp, D. Parikh, and A. Farhadi, "Recipe1M+: A dataset for learning cross-modal embeddings for cooking recipes and food images," *arXiv preprint arXiv:1810.06553*, 2018.

[27] F. Mohbat and M. J. Zaki, "LLaVA-Chef: A multimodal generative model for food recipes," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 1711–1721, ACM, 2024.

[28] A. Myers *et al.*, "Im2Calories: Towards an automated mobile vision food diary," in *2015*

IEEE International Conference on Computer Vision (ICCV), pp. 1233–1241, 2015.

[29] N. Reimers *et al.*, "Sentence-BERT: Sentence embeddings using siamese BERT-networks," in *Conference on Empirical Methods in Natural Language Processing*, 2019.

[30] D. Schwenk *et al.*, "A-OKVQA: A benchmark for visual question answering using world knowledge," in *European Conference on Computer Vision*, 2022.

[31] H. Touvron *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[32] J. Yang *et al.*, "Transferring foundation models for generalizable robotic manipulation," 2024.

[33] D. Yao and B. Li, "Dual-level interaction for domain adaptive semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pp. 4527–4536, 2023.

[34] D. Yao, J. Zhang, I. G. Harris, and M. Carlsson, "FuzzLLM: A novel and universal fuzzing framework for proactively discovering jailbreak vulnerabilities in large language models," in *ICASSP 2024*, pp. 4485–4489, IEEE, 2024.

[35] C. Zhou *et al.*, "LIMA: Less is more for alignment," 2023.

[36] D. Zhu *et al.*, "MiniGPT-4: Enhancing vision-language understanding with advanced large language models," *arXiv preprint arXiv:2304.10592*, 2023.