*electronics*

MDPI

*Article*

# Edge-Constrained Food Retrieval System

**Andrea Spagnolo [1], Simone Bianco [1], Raimondo Schettini [1] and Danilo Pau [2,*]**

[1] Department of Informatics, Systems and Communication, University of Milano-Bicocca, 20126 Milan, Italy; andreaspagnolo99@gmail.com (A.S.); simone.bianco@unimib.it (S.B.); raimondo.schettini@unimib.it (R.S.)
[2] STMicroelectronics, 20864 Agrate Brianza, Italy
* Correspondence: danilo.pau@st.com

**Abstract**

Fine-grained food image and text retrieval requires models that can distinguish subtle differences in ingredients, preparation methods, and visual appearance. While vision language models provide strong multimodal reasoning capabilities, their computational requirements can make direct deployment on edge devices impractical. This paper proposes a training free retrieval framework that combines efficient candidate retrieval with selective vision language reasoning. For ingredient retrieval, Qwen3-VL performs closed choice selection over the retrieved candidates, while dense caption retrieval uses Qwen3-VL-Reranker as a second stage multimodal reranker. An efficiency oriented configuration, Stage 4-E, further introduces confidence based execution, allowing the VLM components to be skipped when the initial retrieval is sufficiently confident. The proposed approach is evaluated on the Dishcovery challenge and on auxiliary food retrieval datasets, with particular attention to the tradeoff between retrieval quality and computational cost. The final pipeline is also deployed on two different edge AI platforms, the NVIDIA Jetson AGX Orin and the Qualcomm Dragonwing IQ-9075 EVK, using platform specific model representations and execution backends. The results show that selective multimodal reasoning, confidence based execution, and hardware aware optimization can substantially reduce inference cost while preserving most of the retrieval performance, making the proposed architecture suitable for practical edge deployment.

**Keywords:** food image–text retrieval; edge AI; vision–language models; training-free retrieval; SigLIP2; Qwen3-VL; multimodal reranking

Check for updates

Academic Editors: Galina Ilieva, George A. Tsihrintzis and Maria Virvou

Received: 18 September 2026
Revised: 27 October 2026
Accepted: 29 October 2026
Published: 1 November 2026

**Copyright:** © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

## 1. Introduction

Fine-grained food image and text retrieval requires matching food images with structured ingredient labels or rich textual descriptions. Although recent vision language models have significantly improved cross modal retrieval, food remains a difficult domain because many dishes share similar visual appearances while differing in small but important details. Retrieving the correct ingredient set or dense textual description therefore requires models to capture subtle visual and semantic cues.

The CVPR Dishcovery VLM MetaFood Challenge [1] provides a realistic benchmark for this problem by evaluating fine grained food retrieval across two complementary tasks: multi label ingredient retrieval and single label dense caption retrieval. High performing multimodal systems often rely on large VLMs [2], model ensembles, or expensive inference pipelines, which can be difficult to deploy when memory, latency, and power consumption are constrained. This is particularly relevant for edge AI platforms, where inference must be adapted to the available compute budget rather than assuming server grade hardware.

This work studies fine grained food retrieval in a power constrained edge AI setting using a pipeline based entirely on frozen pretrained models. The proposed approach combines efficient SigLIP2 candidate retrieval with Qwen3-VL based reasoning applied only to a restricted set of candidates. In addition to evaluating retrieval performance, the paper investigates the practical deployment of the resulting Stage 4-E pipeline on two different edge AI platforms: the NVIDIA Jetson AGX Orin 64GB and the Qualcomm Dragonwing IQ-9075 EVK. On the Jetson platform, the analysis includes different power modes and an optimized deployment using reduced precision models and TensorRT based execution. The same retrieval architecture is then adapted to the Qualcomm platform using its available NPU, GPU, and CPU execution backends. This allows the analysis to consider not only retrieval accuracy, but also latency, memory usage, numerical precision, and the effect of different hardware and software environments on the execution of the same multimodal pipeline.

The paper also provides a more detailed analysis of the complete retrieval process. This includes the normalization of the ingredient vocabulary, the selection of candidate pool sizes and confidence thresholds, the prompts used by the vision language models, and separate evaluations of the two retrieval tasks on MM-Food-100K and Food500-Cap. These additions make it possible to analyze the contribution of the individual pipeline components and provide a more complete description of the experimental procedure.

This paper is organized as follows. Section 2 reviews previous work on food retrieval, image and text retrieval, and multimodal retrieval methods. Section 3 introduces the Dishcovery challenge, the auxiliary evaluation datasets, and the ingredient vocabulary normalization procedure. Section 4 defines the problem and the main system requirements. Section 5 describes the progressive development of the proposed pipeline together with the final experimental configuration, candidate selection strategy, confidence thresholds, and VLM prompts. Section 6 reports the retrieval outcomes. Section 7 analyzes the computational complexity of the proposed methods and their deployment on the NVIDIA Jetson AGX Orin and Qualcomm Dragonwing IQ-9075 EVK. Section 8 discusses the main experimental findings, including the accuracy and computational cost tradeoff, the comparison between the two edge platforms, the effect of ingredient vocabulary normalization, and the main sources of retrieval error. Finally, Section 9 summarizes the main conclusions and discusses possible directions for future work.

## 2. Background and Related Work

Fine grained food image and text retrieval lies at the intersection of food computing and multimodal information retrieval. Recent approaches have achieved strong retrieval performance by combining increasingly powerful visual encoders, vision language models (VLMs), and multiple stages of retrieval, generation, or reranking. However, improvements in retrieval accuracy are often accompanied by higher computational and memory requirements, especially when large multimodal models are repeatedly invoked at query time. This tradeoff becomes particularly relevant when these systems are intended for deployment on hardware with limited memory, computational capacity, and power availability.

The literature relevant to this work can be organized into three complementary directions. First, research on food retrieval has progressively moved from visual food recognition toward multimodal representations that jointly model images, ingredients, recipes, and other semantic information. Second, general image and text retrieval research has developed different mechanisms for cross modal matching, ranging from detailed cross attention to scalable dual encoder architectures and multimodal reranking. Finally, the methods developed for the CVPR Dishcovery VLM MetaFood Challenge provide the closest compar-

ison with the tasks addressed in this paper. Together, these research directions provide the context for the training free retrieval pipeline proposed in the following sections.

### *2.1. Food Specific Multimodal Retrieval*

Food image retrieval belongs to the broader area of food computing, which studies computational methods for acquiring and analyzing heterogeneous food related information such as images, recipes, ingredients, cooking videos, and dietary records. Min et al. [3] provide a comprehensive overview of this research area, identifying food recognition, retrieval, recommendation, perception, and monitoring as closely related computational tasks. Within this broader field, retrieval is particularly challenging because food semantics are not determined by visual appearance alone. Visually similar dishes may contain different ingredients or correspond to different preparation methods, while the same dish can exhibit considerable variation in presentation, portion size, and cooking style. Consequently, food retrieval provides a natural setting for multimodal approaches in which visual evidence is complemented by textual or structured culinary information.

Earlier food retrieval systems primarily relied on visual representations extracted from food images [4–6]. Ciocca et al. [5], for example, investigated CNN-based representations for food classification and retrieval and showed the importance of learning features from sufficiently representative food datasets. Their work also introduced the Food 475 collection, obtained by combining multiple existing datasets, and demonstrated that representations learned from food specific data provide more robust features for food retrieval than representations learned from less representative domains. Although these methods rely only on visual information, they represent an important step toward later multimodal systems because they established the need for representations capable of capturing subtle differences among food categories.

Subsequent research increasingly incorporated semantic information associated with food. Min et al. [7] jointly modeled food images, textual ingredients, and higher level attributes such as cuisine and course information for recipe retrieval and exploration. Their approach learned correlated representations of visual and ingredient information and addressed tasks including food image retrieval and ingredient inference [7,8]. This line of work is particularly relevant to ingredient retrieval because it treats ingredients as an important semantic representation of a food image rather than merely as additional recipe metadata.

Large scale image and recipe datasets further encouraged the development of shared multimodal representation spaces. Marin et al. [9] introduced Recipe1M+, a large corpus containing more than one million structured recipes and approximately 13 million food images. Their approach learns a common embedding space in which corresponding food images and recipes are represented close to one another, enabling retrieval between visual and textual information. Recipe1M+ demonstrated the value of representing food images and structured textual descriptions within the same semantic space and became an important reference for subsequent multimodal food retrieval research.

Building on this paradigm, Wang et al. [10] explicitly addressed two characteristics that make food retrieval difficult: the large visual and semantic variation among examples of the same type of food and the comparatively small differences between related dishes. Their Semantic Consistent and Attention based Networks learn a common representation for food images and recipes while encouraging semantic consistency between the two modalities. A self attention mechanism is also used to obtain more discriminative recipe representations. Evaluations on Recipe1M showed that explicitly modeling semantic information can improve matching compared with approaches that rely only on direct global correspondence between image and recipe embeddings.

Taken together, these works show a progressive transition from visual food retrieval toward richer multimodal representations in which ingredients, recipes, and semantic attributes help resolve ambiguities that cannot always be determined from global visual appearance alone. This progression is directly relevant to the two tasks considered in this paper. Ingredient retrieval requires identifying multiple fine grained semantic concepts that may be only partially observable in an image, while dense caption retrieval requires distinguishing between textual descriptions of visually similar dishes.

### *2.2. Image and Text Retrieval Methods*

Beyond the food specific literature, Image Text Retrieval (ITR) has evolved through several architectural paradigms that provide different tradeoffs between detailed multimodal interaction, scalability, and inference cost. Three directions are particularly relevant to the proposed system: visual semantic alignment through cross attention, scalable dual encoder retrieval, and training free multimodal pipelines that combine efficient retrieval with more expressive reasoning or reranking modules.

#### 2.2.1. Visual Semantic Embedding and Cross Attention

Early image and text retrieval methods focused on explicitly modeling detailed interactions between visual regions and textual tokens. SCAN [11], for example, uses stacked cross attention to infer latent alignments between image regions and words, improving matching accuracy on descriptive image sentence retrieval benchmarks. Such approaches are effective when detailed interactions between modalities are required, since individual textual concepts can be related to localized visual evidence.

Their computational cost, however, makes exhaustive application to large candidate collections less attractive. When image and text representations must interact for every possible pair of query and candidate, inference cost grows with the size of the retrieval database. This limitation motivates architectures in which an inexpensive retrieval model first produces a compact candidate set and a more expressive multimodal model is then applied only to those candidates [12,13]. Modern multimodal rerankers such as Qwen3-VL-Reranker [14] follow this principle by performing joint image and text reasoning as a final refinement step rather than over the complete retrieval collection.

#### 2.2.2. Dual Encoder Models

A more scalable retrieval paradigm is represented by dual encoder models such as CLIP [15] and SigLIP2 [16]. These models encode images and texts independently into a shared embedding space. Text representations from the database can therefore be computed in advance and stored, while a query image needs to be encoded only once during inference. Retrieval can then be performed efficiently through cosine similarity or approximate nearest neighbor search without evaluating deep interactions between the query and every candidate.

This separation between image and text encoding makes dual encoders particularly attractive for large scale retrieval and for systems with limited computational resources. Nevertheless, the same property also limits the amount of detailed interaction that can occur between the two modalities. A single global similarity score may be insufficient when candidate descriptions differ only in subtle visual or semantic details. This limitation is particularly relevant in the food domain, where two dishes may share most of their appearance while differing in a small ingredient, garnish, sauce, or preparation method. Dual encoders therefore provide an efficient mechanism for retrieving a set of plausible candidates, while difficult examples can benefit from a more expressive second stage model.

#### 2.2.3. Training Free Multimodal Retrieval Pipelines

A recent research direction investigates how pretrained vision language and language models can be combined into retrieval pipelines without adapting their parameters to the target task. Rather than fine tuning the underlying foundation models, these approaches perform task specialization during inference through prompt design, generated textual representations, similarity based retrieval, and deterministic or model based reranking.

CIReVL [17], proposed by Karthik et al., is a "Vision by Language" pipeline for training free compositional image retrieval. Given a reference image and a textual modification, the method first uses a generative VLM to describe the reference image and then asks a language model to incorporate the requested modification into a new target description. The generated description is subsequently encoded using a frozen CLIP text encoder and used for retrieval. The approach demonstrates that generative multimodal reasoning can be combined with a frozen contrastive retrieval backbone without introducing task specific training.

Sun et al.'s training free local concept reranking method [18] follows a complementary strategy for zero shot composed image retrieval. A frozen CLIP style dual encoder first retrieves an initial candidate set, after which the ranking is refined by explicitly modeling local visual and textual concepts associated with the query and candidate images. By restricting the more expensive refinement stage to a reduced candidate pool, the method shows how retrieval quality can be improved without modifying the parameters of the underlying encoder.

These approaches are closely related to the design philosophy adopted in this paper. Pretrained models are treated as reusable components, while adaptation to the task is achieved through modular operations performed during inference rather than through supervised fine tuning. In particular, using a frozen dual encoder for efficient candidate generation and applying more expressive multimodal reasoning only to a compact candidate set provides a natural way to balance retrieval scalability and fine grained matching accuracy. The proposed pipeline follows this general principle by using SigLIP2 for candidate retrieval and Qwen3-VL based modules for constrained selection or final reranking.

### *2.3. Works Related to the CVPR Dishcovery VLM MetaFood Challenge*

The methods developed specifically for the CVPR Dishcovery VLM MetaFood Challenge provide the closest comparison with the system proposed in this paper because they address the same ingredient retrieval and dense caption retrieval tasks.

F4-ITS (Fine-Grained Feature Fusion for Food Image-Text Search) [19] is one of the most relevant training free retrieval frameworks proposed for the challenge. The method improves standard contrastive retrieval by enriching visual query representations with textual descriptions generated by a VLM and by applying feature level fusion before similarity based ranking. In the caption retrieval setting, F4-ITS combines visual embeddings with generated dense descriptions to obtain a more expressive query representation, which is subsequently compared with the textual index using cosine similarity. For ingredient retrieval, food components predicted by the VLM are used to guide reranking. The approach therefore demonstrates that textual information generated from the query image can improve detailed food retrieval without adapting model parameters to the target task.

LXlearning [20] adopts a different strategy based on supervised recall and reranking. A SigLIP family dual encoder first retrieves a compact set of candidate ingredients or captions from the fixed challenge database. A Qwen2.5-VL-7B-Instruct model [21] adapted with LoRA [22] is then used to rerank the retrieved candidates and determine the final prediction. Restricting the VLM to a reduced candidate set limits its inference cost compared with exhaustive multimodal evaluation. However, the approach introduces additional offline

steps, including pseudo label generation, synthetic data construction, LoRA fine tuning, and checkpoint merging. These operations are less aligned with the strict training free setting considered in this paper.

The comparison between these systems illustrates an important design tradeoff. Training free approaches such as F4-ITS avoid task specific optimization and are therefore naturally compatible with frozen model deployment, but their performance can depend on the reliability of unrestricted VLM generation and subsequent retrieval heuristics. Supervised reranking approaches such as LXlearning can adapt the multimodal model more directly to the target task, but require an additional training stage and the computational resources associated with it.

The proposed method follows the training free direction but differs from both approaches in the way multimodal reasoning is integrated into the retrieval process. SigLIP2 is used as an efficient initial retriever, while ingredient prediction is formulated as constrained closed choice selection rather than unrestricted generation. For dense caption retrieval, Qwen3-VL-Reranker is applied only after SigLIP2 has reduced the complete collection to a small set of plausible captions. The more expensive multimodal models therefore operate on limited candidate sets rather than on the complete retrieval collection. This design preserves the training free setting while allowing the computational cost of multimodal reasoning to remain bounded and suitable for the edge oriented evaluation considered in this paper.

## 3. Problem Formulation and Datasets

This section introduces the two retrieval tasks considered in the paper and describes the official and auxiliary datasets used for their evaluation. It also presents the normalization procedure applied to the ingredient vocabulary for the auxiliary Task 1 experiments.

### *3.1. Datasets and Challenge Setting*

This study is conducted in the context of the CVPR Dishcovery VLM MetaFood Challenge [1], which focuses on fine-grained food image-text retrieval. The challenge evaluates vision-language pipelines on hidden test sets and is structured around two complementary tasks:

- Task 1: Multi-label Ingredient Retrieval: Given a query food image, the system must retrieve all matching ingredient-level labels from a sparse vocabulary of 399 dish components. Since a dish can contain multiple ingredients, this task is formulated as multi-label retrieval and is evaluated on 6756 test instances.
- Task 2: Single-label Caption Retrieval: Given a query food image, the system must identify the single correct dense narrative caption from an indexed corpus of 4650 textual descriptions. Since each image is associated with exactly one description, this task is formulated as single-label retrieval and is evaluated on 2892 test instances.

While the leaderboard provides an end-to-end measure of performance, its hidden evaluation protocol limits detailed analysis. In particular, the public score is reported as the harmonic mean of the two task scores, making it difficult to determine whether a given improvement comes from ingredient retrieval, caption retrieval, or both. Let $S_1$ be the Task 1 score and $S_2$ be the Task 2 score. The final challenge score $S$ is computed as:

$$S = \frac{2S_1S_2}{S_1 + S_2}. \quad (1)$$

In addition, the hidden ground truth prevents direct inspection of failure cases and limits the study of preprocessing choices.

This limitation is especially relevant for Task 1, where the ingredient vocabulary contains lexical variants and near-duplicates, including alternative spellings, synonyms, and singular and plural forms. For example, semantically equivalent labels such as "barbecue sauce" and "bbq sauce" may be treated as distinct entries. Such distinctions are often not meaningful in practical retrieval settings, but can artificially penalize systems that retrieve the correct ingredient using a different surface form.

To complement the hidden challenge evaluation, we use two auxiliary datasets with accessible annotations: MM-Food-100K [23] for ingredient retrieval and Food500-Cap [24] for dense-caption retrieval.

For Task 1, we first filtered MM-Food-100K to retain only images whose ground-truth ingredients belonged to our cleaned 295-label vocabulary, obtaining a pool of 14,119 images. From this pool, we selected 6756 images to approximate the ingredient-count distribution of Dishcovery. Since the official Dishcovery ground truth was unavailable, we used the ingredient counts predicted by our previous Dishcovery Task 1 submission as a proxy for this distribution. We then performed stratified random sampling based on the number of ingredients per image. The resulting distribution had a total variation distance of 0.0352 from the prediction-derived reference. This alignment concerns the number of ingredients per image, not individual ingredient frequencies or equivalent visual difficulty.

For Task 2, we used a class-balanced subset of Food500-Cap comprising 2820 query images from 470 food classes, with six images per class. Each query was evaluated against the complete bank of 4940 candidate captions.

Finally, to enable a direct quantitative comparison with published solutions from the previous edition of the competition [19,20], we also use the CVPR 2025 Dishcovery benchmark [25] test set. It follows the same two-task structure on 3823 Task 1 images with a larger 495-label ingredient vocabulary and 3253 Task 2 images over the 4650-caption corpus, using the geometric mean of Task 1 F1 and Task 2 Top-1 accuracy as the official evaluation metric.

### *3.2. Ingredient Vocabulary Normalization*

For the auxiliary Task 1 experiments on MM-Food-100K, the original 399-label ingredient vocabulary was manually inspected and normalized before generating the corresponding text embeddings. This preprocessing step was introduced to study how lexical redundancy and label ambiguity affect ingredient retrieval evaluation. While the official Dishcovery benchmark retains its fixed vocabulary, the auxiliary dataset allows comparing the original and normalized label spaces using accessible ground-truth annotations.

The original vocabulary contained 399 ingredient captions formatted as "A plate of food containing <ingredient>". Through a manual, author-reviewed semantic normalization policy, the vocabulary was reduced to 295 entries (a reduction of 104 labels, or 26.1%), while keeping the caption template unchanged. Specifically, across the 399 source labels, 205 were kept unchanged, 112 underwent lexical normalization, 26 were consolidated through task-specific semantic merges, and 56 ambiguous or non-reusable entries were removed. The complete 399-to-295 mapping table and decision criteria are provided in the public repository (see Data Availability Statement). The normalization was organized around the following main operations.

#### 3.2.1. Singular and Plural Normalization

The most frequent source of redundancy was the simultaneous presence of singular and plural forms. These variants were mapped to a single canonical label, generally expressed in the singular form. For example, *almonds* was converted to *almond*, while pairs such as *apple*/*apples*, *carrot*/*carrots*, and *tomato*/*tomatoes* were represented using only their singular

form. The same procedure was applied to multi-word expressions, such as *baked beans* → *baked bean*, *bean sprouts* → *bean sprout*, and *pumpkin seeds* → *pumpkin seed*.

This operation prevents the retrieval model from treating grammatical-number variations as independent semantic classes. Such distinctions are not useful for the considered task, since the objective is to identify the ingredient category rather than estimate the number of visible instances.

#### 3.2.2. Synonym, Abbreviation, and Spelling Consolidation

Different lexical expressions referring to the same ingredient were merged into a single preferred form. For example, *bbq sauce* was consolidated into *barbecue sauce*. Similarly, *cashews* and *cashew nuts* were represented by the canonical label *cashew*, while the tokenization variants *flax seeds* and *flaxseeds* were merged into *flax seed*.

Semantically equivalent culinary terms were also consolidated when the distinction was not relevant to visual recognition. Examples include retaining *sea urchin* instead of the alternative culinary name *uni*, and retaining *roast duck* rather than maintaining both *roast duck* and *roasted duck*. These transformations reduce lexical fragmentation and ensure that visually equivalent examples contribute to the same textual representation.

#### 3.2.3. Removal of Explanatory Text

Some original labels included parenthetical descriptions containing ingredients, preparation details, or short definitions. These descriptions were removed so that every class was represented by a concise noun phrase. For instance, *chokha* (*mashed vegetables*, *spices*) was reduced to *chokha*, while *litti* (*wheat flour*, *ghee*, *stuffed with sattu*) was reduced to *litti*.

Although these explanations can be informative to a human reader, they introduce additional words into the text embedding that are not part of the class name. They may therefore bias the similarity calculation towards the ingredients mentioned in the explanation rather than towards the visual appearance of the target food.

#### 3.2.4. Consolidation of Preparation and Presentation Variants

Preparation-related modifiers were removed or merged when they did not define a sufficiently distinct visual category. For example, the original vocabulary contained generic *egg* and *eggs* labels together with variants such as *boiled eggs* and *poached eggs*. The normalized vocabulary retains the general label *egg*. Similarly, *white rice* was absorbed into the broader *rice* category.

This rule was not applied indiscriminately. Preparation terms were retained when they described a visually meaningful and consistently recognizable category. Labels such as *fried chicken*, *grilled eel*, *fried rice*, and *tomato sauce*, for example, remain separate from their corresponding base ingredients. The cleaning process was therefore based on semantic and visual judgement rather than on the automatic removal of every preparation adjective.

#### 3.2.5. Reduction of Overly Specific or Overlapping Categories

Some labels represented highly specific subtypes that substantially overlapped with broader classes already included in the vocabulary. Where the visual distinction was considered unreliable or unnecessary for the retrieval task, the more specific entry was eliminated as a standalone class and consolidated into a broader category (`SEMANTIC_MERGE` in the mapping table).

For example, specific pasta descriptions such as *black pasta*, *fettuccine*, and *lasagna noodles* were consolidated into general classes such as *pasta*, *noodle*, and *spaghetti*. Similarly, expressions such as *various fish*, *whole fish*, and certain preparation-specific meat labels were merged into existing categories such as *fish*, *pork*, or *duck*.

The purpose of this operation was not to construct a strict food taxonomy, but to obtain a vocabulary whose classes were sufficiently distinct to be used as retrieval targets.

#### 3.2.6. Removal of Ambiguous and Unsuitable Labels

The original vocabulary also contained 56 entries (`REMOVE` in the mapping table) that did not clearly identify an ingredient, referred to decorative or structural elements, or were too ambiguous to define a stable retrieval class. Examples include labels such as *decorations*, *crispy cups*, and *cake layers*. These entries were discarded rather than mapped to another class.

Several complete-dish names and highly dataset-specific food terms were also excluded when they could not be interpreted as reusable ingredient categories or when their visual content was already covered by more general labels. However, culturally specific terms were retained when they represented a sufficiently coherent and recognizable category. Consequently, labels such as *dosa*, *idli*, *fufu*, *rasgulla*, *chokha*, and *litti* remain in the normalized vocabulary. This again indicates that the cleaning process was based on manual semantic validation rather than on a rule that automatically removed all dish-level terms.

Overall, the normalization process aimed to reduce semantic overlap among candidate labels while preserving distinctions that are useful for visual recognition. The resulting vocabulary is therefore more compact and internally consistent, but it still contains both basic ingredients and selected prepared-food categories when these correspond to visually distinctive concepts.

This preprocessing is particularly important in a text–image retrieval setting. Each label is independently encoded into the shared embedding space, meaning that lexical duplicates can compete against one another even when they describe the same visual content. By assigning a single textual representation to equivalent concepts, vocabulary normalization reduces this competition and makes the similarity scores easier to interpret. It also improves the consistency of the evaluation, since predictions are less likely to be penalized solely because of pluralization, spelling variation, or the use of an equivalent culinary term.

The vocabulary normalization and the formal 399-to-295 mapping policy were conducted by the primary author and reviewed by the co-authors (no formal multi-annotator agreement study was performed). In constructing the auxiliary MM-Food-100K evaluation set, ground-truth annotations were curated into the 295-label space prior to image filtering (including targeted image-level disambiguation where needed), so that all retained evaluation instances contain only valid 295-space ground-truth labels. To separate the effect of evaluating on this consistent label space from the effect of running retrieval directly on a smaller candidate pool, we also defined a post-hoc *Harmonized* evaluation setting. In this setting, the archived predictions originally generated using the 399-label vocabulary are deterministically translated into the 295-label space via the global mapping table without re-running the models or altering the ground truth: lexical and semantic-merge variants are mapped to their canonical target, resulting duplicate predictions within the same image are merged, and predictions corresponding to the 56 removed labels are discarded. Both this *Harmonized* baseline and the *Cleaned retrieval* configuration are evaluated against the exact same 295-label ground truth.

## 4. System Requirements

The central problem addressed by this work is the design of an image and text retrieval pipeline for the CVPR Dishcovery VLM MetaFood Challenge that achieves competitive retrieval performance while remaining suitable for deployment on resource constrained edge AI platforms. The target scenario is characterized by limited computational resources, memory, inference latency, and power availability. Consequently, the proposed system

should avoid server scale hardware, ensembles of very large models, repeated expensive computation at query time, and retrieval strategies whose cost becomes prohibitive as the candidate database grows.

A second requirement is that the pipeline operates in a training free setting and relies exclusively on frozen pretrained models. No task specific fine tuning, LoRA adaptation, pseudo label training, supervised calibration, or other offline learning procedure is performed. Avoiding these operations reduces the computational and hardware requirements associated with adapting large vision language models, which would typically require resources considerably beyond those available on the considered edge platforms.

However, computational cost is not the only motivation for this choice. The increasing capabilities of general-purpose foundation models have made training-free adaptation a practical alternative for a growing number of downstream tasks. Rather than learning a separate set of model parameters for every application, pretrained models can be adapted at inference time through prompting, retrieval, candidate restriction, and other forms of constrained reasoning. Fine tuning remains valuable when maximum specialization is required, but it can also make a solution more dependent on a particular dataset, task formulation, or label space.

This distinction is particularly relevant for the retrieval problems considered in this work. Since the proposed pipeline does not encode the ingredient vocabulary into task-specific learned parameters, the candidate ingredient list can be modified without retraining the underlying models. New textual candidates only require the corresponding embeddings to be computed. Similarly, the modular architecture facilitates the replacement of individual pretrained components when newer or more capable models become available. Such a replacement may still require changes to preprocessing, prompts, quantization, or deployment configuration, but it does not require repeating task-specific training.

A further requirement is that the high level retrieval architecture should not depend on a specific hardware platform or inference runtime. The individual models may require different numerical representations, execution backends, or deployment configurations depending on the target device, but the retrieval logic, candidate selection strategy, and confidence based execution should remain unchanged. This separation between the retrieval method and its hardware implementation makes it possible to adapt the same pipeline to different edge accelerators without redesigning the complete system.

This property is especially useful in the current vision-language model ecosystem, where model capabilities and architectures evolve rapidly. Designing the pipeline around frozen and interchangeable components therefore provides an additional form of flexibility: improvements in the underlying foundation models can potentially be incorporated without redesigning or retraining the complete retrieval system.

The objective is therefore not only to investigate how far a fully frozen vision language retrieval pipeline can be pushed in terms of fine grained food retrieval performance, but also to develop a system that remains computationally practical, modular, and adaptable to different candidate vocabularies, pretrained models, and edge computing platforms.

## 5. Methods

The proposed final pipeline was developed through a multi-stage iterative process, progressively evolving from simple visual retrieval toward a hybrid architecture that balanced efficient dual-encoder recall with bounded VLM reasoning for deployment on edge systems.

### *5.1. Stage 1: SigLIP2-Based Visual Retrieval*

An initial version of the pipeline was based exclusively on SigLIP2 [16]. In this stage, both tasks were treated as visual-textual retrieval problems. Query images were encoded

with the SigLIP2 image encoder, while candidate ingredients or captions were represented through the corresponding text encoder. The final predictions were achieved by computing the similarity scores between images and text in the shared embedding space.

This baseline provided a strong and efficient starting point, since SigLIP2 offered robust contrastive representations while keeping the inference complexity low. However, the model relied entirely on embedding similarity and therefore had limited ability to reason explicitly about ambiguous visual evidence, fine-grained ingredients, or semantically close captions. These limitations motivated the introduction of a vision-language model as an additional reasoning component.

### *5.2. Stage 2: Qwen Open-Choice*

An improved version introduced Qwen3-VL [26] as a complementary semantic module. In this version, the model was prompted in an open-choice setting: given the food image, it generated either a free-form textual description or a list of visible ingredients without being constrained to the official label vocabulary.

This strategy exploited the descriptive capabilities of the VLM. In particular, Qwen3-VL could identify high-level dish properties, cooking methods, sauces, garnishes, and visual details that were not always captured by the SigLIP2 embedding space alone. For Task 2, this open descriptive signal was especially useful, since the objective was to retrieve a dense narrative caption rather than a sparse set of labels.

The textual output generated by Qwen3-VL was not used directly as the final prediction. Instead, it was treated as an additional semantic representation of the query image. The generated description or ingredient list was encoded with the text branch of the contrastive model and combined with the original image embedding. Cosine similarity was then computed between the fused query representation and the target embeddings. For Task 1, the top-$k$ ingredient candidates were selected from the ranked vocabulary, while for Task 2 the top-1 caption was retrieved from the caption database.

However, open-choice generation also introduced several issues for Task 1. Since the model was not constrained to the official ingredient vocabulary, it could produce synonyms, overly generic food terms, or ingredients that were plausible but not actually present in the image. This made the output difficult to map reliably to the required multi-label format and increased the risk of hallucinated ingredients. To address this issue, the open-choice Qwen3-VL ingredient predictions were mapped back to the official ingredient vocabulary. This remapping step slightly improved performance, suggesting that the VLM outputs contained useful semantic information. Nevertheless, the approach remained indirect and error-prone, since free-form predictions still had to be normalized and aligned with the target labels. This observation motivated the shift toward a closed-choice setting, where the VLM directly selects visible ingredients from a constrained candidate list.

### *5.3. Stage 3: Qwen Closed-Choice*

To address the limitations of open-choice generation, an improved pipeline reformulated the use of Qwen3-VL for Task 1 as a closed-choice selection problem, while preserving the same embedding-based fusion and retrieval framework. Instead of freely generating ingredients, SigLIP2 was first used to retrieve a limited set of visually plausible candidate ingredients. Then Qwen3-VL was prompted to select only the visible ingredients from this candidate list. The selected labels were then embedded and fused with the image-based retrieval scores before computing the final ranked predictions.

This change made the VLM output directly compatible with the official Task 1 vocabulary. By constraining the answer space, the closed-choice setting reduced hallucinations, avoided synonym-matching errors, and allowed Qwen3-VL to act as a semantic verifier over

a candidate set proposed by the contrastive retriever. The resulting design combined the efficiency of SigLIP2 retrieval with the stronger visual reasoning capabilities of Qwen3-VL. A closed-choice formulation was also tested for Task 2, where Qwen3-VL was asked to choose among a restricted set of candidate captions. However, this modification did not improve top-1 accuracy. While closed-choice reasoning was effective for Task 1 due to the fixed ingredient vocabulary, Task 2 requires retrieving a dense narrative caption, for which the open-choice description generated by the VLM provided a more useful semantic signal. As a result, the open-choice strategy remained the most effective configuration for Task 2 at this stage.

### *5.4. Stage 4: Qwen Reranker*

This new step adopted two task-specific pipelines. For Task 1, the Stage 4 configuration kept the SigLIP2 retrieval backbone combined with the closed-choice Qwen3-VL selection strategy introduced in the previous stage.

For Task 2, the stage 4 pipeline replaced the previous open-choice VLM description strategy with a two-stage retrieval and reranking architecture based on SigLIP2 and Qwen3-VL-Reranker [14]. Since the objective was to select a single dense caption from a large caption database, SigLIP2 was first used as an efficient high-recall retriever. The query image was encoded with the SigLIP2 image encoder, while all candidate captions were encoded in advance with the corresponding text encoder. Image-text similarity scores were then computed over the full caption collection, producing a compact set of top candidate captions.

These candidates were subsequently passed to Qwen3-VL-Reranker, which acted as a multimodal cross-encoder. Unlike the dual-encoder retrieval stage, where image and text were embedded independently, the reranker evaluates each image-caption pair jointly. This allows a direct comparison between the visual content of the food image and the proposed menu-style description, with particular attention to fine-grained details such as ingredients, preparation method, sauces, garnishes, and plating style.

To keep inference computationally feasible, the reranker was applied only to the most promising captions retrieved by SigLIP2. The final Task 2 prediction was selected from this reranked candidate set. This design combined the scalability and recall of SigLIP2 with the stronger cross-modal reasoning of Qwen3-VL-Reranker, which was particularly useful when several captions were visually similar but differed in subtle semantic details.

For both tasks, retrieval uses L2-normalized SigLIP2 embeddings. The implementation multiplies the cosine similarity by a fixed factor of 100, so the retrieval scores are

$$s_i = 100 \cos(h, z_i), \qquad r_j = 100 \cos(h, u_j). \tag{2}$$

This fixed scaling does not change the Top-$K$ ranking, but it defines the numerical scale used by the confidence thresholds described in Section 5.7. It is a constant applied to cosine similarity and is not the learned SigLIP2 logit scale or bias.

In Algorithm 1 (Task 1) and Algorithm 2 (Task 2), we summarize the complete Stage 4 inference procedures. In Algorithm 1, Fuse combines the visual retrieval scores with the information provided by Qwen3-VL. The ingredients selected as present by Qwen3-VL are used to obtain a VLM-derived semantic score vector over the ingredient vocabulary. This vector and the SigLIP2 retrieval scores are normalized separately and combined with weights 0.75 and 0.25, respectively. The final selection thresholds are specified in Section 5.7.

**Algorithm 1:** Stage 4 pipeline for Task 1: multi-label ingredient retrieval

**Data:** Query image $x$, ingredient vocabulary $\mathcal{V}$, candidate size $K_I$
**Data:** SigLIP2 image encoder $E_I$, SigLIP2 text encoder $E_T$, Qwen3-VL selector $Q_{\text{sel}}$
**Result:** Final ingredient set $\mathcal{Y}^*$

`// Offline preprocessing: performed once`
**foreach** *ingredient label* $v_i \in \mathcal{V}$ **do**
  $z_i \leftarrow E_T(v_i)$;
Store ingredient embeddings $\mathcal{Z} = \{z_i\}$;

`// Online inference: performed for each query image`
$h \leftarrow E_I(x)$;
**foreach** *ingredient label* $v_i \in \mathcal{V}$ **do**
  $s_i \leftarrow 100 \cos(h, z_i)$;
$\mathcal{V}_K \leftarrow \text{TopK}_{K_I}(\mathcal{V}, s)$;
$\hat{\mathcal{Y}} \leftarrow Q_{\text{sel}}(x, \mathcal{V}_K)$;
$f \leftarrow \text{Fuse}(s, \hat{\mathcal{Y}})$;
$\mathcal{Y}^* \leftarrow \text{Select}(f)$;
**return** $\mathcal{Y}^*$;

**Algorithm 2:** Stage 4 pipeline for Task 2: dense caption retrieval

**Data:** Query image $x$, caption database $C$, candidate size $K_C$
**Data:** SigLIP2 image encoder $E_I$, SigLIP2 text encoder $E_T$, Qwen3-VL-Reranker $Q_{\text{rank}}$
**Result:** Final caption $c^*$
`// Offline preprocessing: performed once`
**foreach** *caption* $c_j \in C$ **do**
  $u_j \leftarrow E_T(c_j)$;
Store caption embeddings $\mathcal{U} = \{u_j\}$;

`// Online inference: performed for each query image`
$h \leftarrow E_I(x)$;
**foreach** *caption* $c_j \in C$ **do**
  $r_j \leftarrow 100 \cos(h, u_j)$;
$C_K \leftarrow \text{TopK}_{K_C}(C, r)$;
**foreach** *candidate caption* $c_j \in C_K$ **do**
  $\rho_j \leftarrow Q_{\text{rank}}(x, c_j)$;
$c^* \leftarrow \arg\max_{c_j \in C_K} \rho_j$;
**return** $c^*$;

### *5.5. Stage 4-E: Edge-Optimized Dual-Task Pipeline*

Although Stage 4 improved both subtasks, its Qwen-based components remained the most expensive inference steps. Task 1 used Qwen3-VL as a closed-choice selector over the top SigLIP2 ingredient candidates, while Task 2 used the 8B Qwen3-VL-Reranker to rerank the top caption candidates.

Stage 4-E reduces this cost by applying the Qwen modules only when the SigLIP2 retrieval stage is uncertain. For Task 1, Qwen3-VL closed-choice selection is skipped when the SigLIP2 ingredient scores are confident enough to produce the final ingredient set directly. Otherwise, Qwen3-VL is used as a verifier over the retrieved ingredient candidates.

For Task 2, the same guarded strategy is applied: confident SigLIP2 caption predictions are returned directly, while ambiguous cases are passed to a smaller 2B Qwen3-VL-Reranker instead of the 8B reranker used in Stage 4. This reduces both the number of reranker calls and the parameter footprint of the caption-retrieval branch.

Therefore, Stage 4-E keeps the same SigLIP2 retrieval backbone as Stage 4, but replaces unconditional Qwen execution with confidence-gated execution for both subtasks. This

reduces the average number of expensive VLM calls while preserving multimodal reasoning for difficult images. With scores sorted in descending order, Task 1 uses the relative gap

$$\gamma_I = \frac{s_{(1)} - s_{(2)}}{|s_{(1)}|}, \tag{3}$$

and bypasses Qwen3-VL when $\gamma_I \geq 0.25$. Task 2 uses the absolute scaled-score gap

$$\gamma_C = r_{(1)} - r_{(2)}, \tag{4}$$

and bypasses the reranker when $\gamma_C \geq 3.0$. Because of the fixed ×100 scaling in Equation (2), the Task 2 threshold is equivalent to a cosine-similarity margin of 0.03.

Figures 1 and 2 summarize the two branches of the Stage 4-E pipeline.

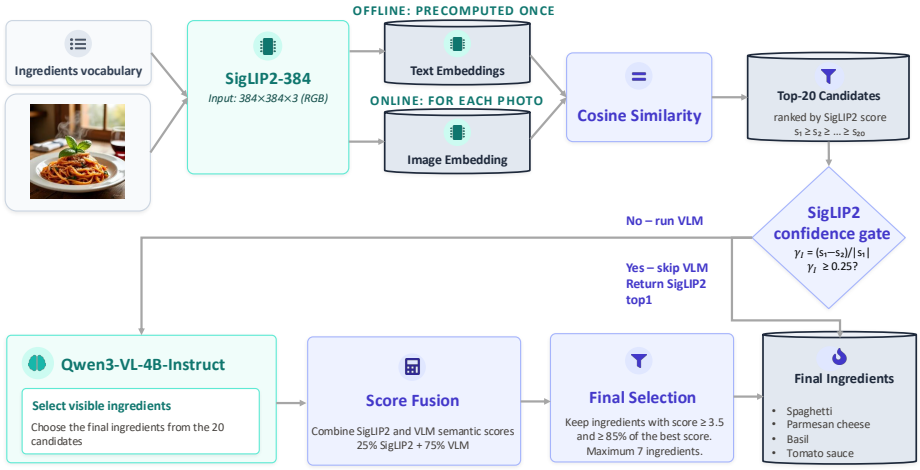



**Figure 1.** Stage 4-E ingredient recognition pipeline for Task 1 (arrows indicate the data flow; green blocks denote neural models, blue blocks denote scoring and selection stages, and grey cylinders denote data/embedding stores). Similarity values are fixed ×100 scaled cosine scores. SigLIP2 operates at 384 × 384; platform-specific Qwen visual preprocessing is described in Section 7.3.

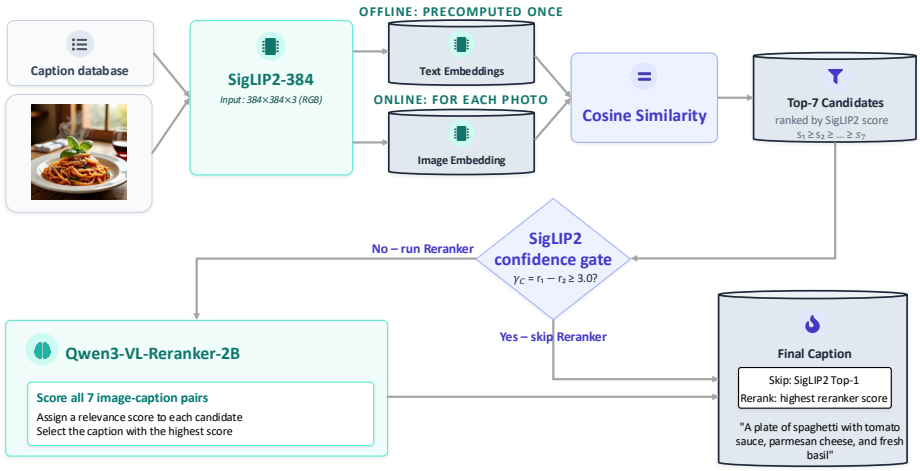



**Figure 2.** Stage 4-E caption retrieval and reranking pipeline for Task 2 (arrows indicate the data flow; green blocks denote neural models, the blue diamond denotes the confidence gate, and grey cylinders denote data/embedding stores). Similarity values are fixed ×100 scaled cosine scores; the 3.0 guard therefore corresponds to a cosine-similarity margin of 0.03. SigLIP2 operates at 384 × 384; platform-specific Qwen visual preprocessing is described in Section 7.3.

### *5.6. Stage 5: Leaderboard-Oriented Upper Bound*

The final pipeline introduces an additional CLIP-based embedding branch [15] that was ensembled with SigLIP2 in the recall stage. The motivation was to improve candidate recall by combining complementary visual-textual representations.

This final pipeline introduced an ingredient correlation matrix. This matrix was built using external recipe knowledge from Recipe1M [27] and CulinaryDB [28], and was used to boost ingredients that frequently co-occur with high-confidence predictions. For example, if the model assigned a high confidence score to "pasta", correlated ingredients such as tomato sauce, basil, or cheese could receive an additional score boost. This approach injected useful culinary priors into the retrieval process, but also introduced the risk of reinforcing dataset biases and predicting plausible ingredients that were not actually visible in the image.

The final addition was the reintroduction of an open-choice VLM call during the recall stage. The motivation was to recover fine-grained ingredients that the embedding-based retriever could miss, especially in difficult cases where subtle visual details were not sufficiently captured by the contrastive model.

#### Role of Stage 5 in the Rest of the Paper

Stage 5 was introduced as a challenge-oriented variant of the pipeline. Its purpose was to maximize the public Kaggle leaderboard score by combining all the available accuracy-improving components, including the additional CLIP-based embedding branch, the ingredient correlation matrix, and the re-introduced open-choice VLM call during the recall stage. As reported in the ablation study, this configuration achieved the highest challenge score among the tested variants.

However, Stage 5 should not be considered the final edge-deployable pipeline. Each of the additional components increases either the inference time, the number of query-time model calls, or the memory footprint of the system. Therefore, although Stage 5 is useful to show the maximum score reachable by the proposed approach on the Kaggle challenge, it is less aligned with the edge-oriented constraints defined in Section 4. This is also reflected in the complexity analysis in Section 7, where Stage 5 has a larger parameter footprint than the corresponding Stage 4 and Stage 4-E configurations.

For this reason, throughout the remaining sections, Stage 4 and Stage 4-E are treated as final proposed pipelines. Stage 4 represents the accuracy-oriented operating point, while Stage 4-E represents the efficiency-oriented operating point. Stage 5 is reported separately as a leaderboard-oriented upper bound, because it adds additional components that increase memory footprint, query-time computation, and system complexity.

### *5.7. Final Experimental Configuration and Hyperparameter Selection*

The final Stage 4-E experiments use a fixed configuration for the two retrieval branches introduced in Section 5.5. To avoid adaptive tuning on the hidden Dishcovery test set, all pipeline hyperparameters including candidate pool sizes, confidence-gating thresholds, score-fusion weights, final selection cutoffs, and prompt structure were selected offline using empirical analyses on the 350 instance auxiliary development subsets of MM-Food-100K and Food500-Cap (disjoint from the final evaluation sets) and kept unchanged across all reported experiments.

#### 5.7.1. Task 1 Configuration

The final Task 1 configuration used a fixed candidate pool of $K = 20$ ingredients. Retrieval scores were computed according to Equation (2), and Qwen3-VL-4B-Instruct was applied only when the relative gap $\gamma_I = (s_{(1)} - s_{(2)})/|s_{(1)}|$ was below 0.25. When the

VLM branch was executed, only labels returned in the `present` set were used to derive the semantic score vector over the full vocabulary. The normalized SigLIP2 and VLM-derived scores were then combined with weights 0.25 and 0.75, respectively. Final predictions were selected using a normalized fused-score threshold of 3.5, together with a relative threshold of 0.85 with respect to the highest fused score. At most seven ingredients were retained. Table 1 summarizes the main Task 1 parameters.

These values were determined empirically on the auxiliary development data: the 0.25/0.75 fusion weights prioritize explicit VLM visual verification while retaining the SigLIP2 ranking signal to break ties, whereas the fused-score cutoff (3.5) and relative threshold (0.85) filter out low-confidence tail candidates. The cap of seven output ingredients reflects the natural cardinality distribution of the dataset, where over 99% of dishes contain seven or fewer annotated ingredients. Similarly, the 0.25 relative-gap skip threshold identifies queries where the top SigLIP2 candidate is strongly separated from the runner-up.

**Table 1.** Main parameters used for Stage 4-E Task 1.

| Parameter | Value |
|---|---|
| Candidate pool size | 20 |
| VLM skip threshold | 0.25 relative gap |
| SigLIP2/VLM fusion weights | 0.25/0.75 |
| Final VLM selection | Score $\geq$ 3.5 and $\geq$85% of best |
| Maximum output ingredients | 7 |

#### 5.7.2. Task 2 Configuration

The final Task 2 configuration used a fixed candidate pool of $K = 7$ captions and Qwen3-VL-Reranker-2B as the second-stage model. Retrieval scores followed Equation (2). Reranking was skipped when $r_{(1)} - r_{(2)} \geq 3.0$, which is equivalent to a cosine-similarity margin of 0.03. Otherwise, all seven candidates were scored by the reranker and the caption with the highest reranker score was returned. The 3.0 scaled-score skip threshold (0.03 cosine margin) was selected on the Food500-Cap development subset as the operating point that bypasses approximately one quarter of unambiguous queries without degrading Top-1 accuracy. Table 2 summarizes the main Task 2 parameters.

**Table 2.** Main parameters used for Stage 4-E Task 2.

| Parameter | Value |
|---|---|
| Candidate pool size | 7 |
| Reranker | Qwen3-VL-Reranker-2B |
| Reranker skip threshold | 3.0 scaled-score gap (0.03 cosine gap) |
| Final prediction | Highest reranker score |

#### 5.7.3. Candidate Pool Size Selection

The candidate pool size controls an important tradeoff in both tasks. Increasing the number of candidates improves the probability that the correct ingredient or caption reaches the Qwen stage, but it also increases the amount of multimodal computation required for each query.

The final candidate pool sizes were selected empirically using two separate subsets of 350 instances. For Task 1, the subset was constructed using images from MM-Food-100K,

while for Task 2 it was constructed using images from Food500-Cap. In both cases, the images used for parameter selection were excluded from the subsets used for the final evaluations reported in Sections 6.2 and 6.3. Candidate pool selection and final evaluation were therefore performed on disjoint sets of images.

For Task 1, fixed candidate pools of 10, 20, and 30 ingredients were compared with a dynamic candidate strategy. For Task 2, fixed pools of 3, 5, and 7 captions were compared with a corresponding dynamic strategy. Table 3 reports the results.

The dynamic strategy adapts the number of candidates to the confidence of the SigLIP2 ranking. When the highest scores are clearly separated, fewer candidates are retained; when several candidates receive similar scores, a larger candidate set is passed to the second-stage model.

**Table 3.** Candidate size ablation study using quantized models on separate 350 instance subsets reserved for parameter selection.

| Task | Run | Avg. Candidates | Score | Avg. Time |
|---|---|---|---|---|
| Task 1 | Top 10 | 10.0 | 0.743 F1 | 0.868 s |
| Task 1 | Top 20 | 20.0 | 0.758 F1 | 0.978 s |
| Task 1 | Top 30 | 30.0 | 0.746 F1 | 1.244 s |
| Task 1 | Dynamic | 13.2 | 0.741 F1 | 0.998 s |
| Task 2 | Top 3 | 3.0 | 0.666 Acc. | 0.691 s |
| Task 2 | Top 5 | 5.0 | 0.671 Acc. | 1.052 s |
| Task 2 | Top 7 | 7.0 | 0.689 Acc. | 1.388 s |
| Task 2 | Dynamic | 6.8 | 0.691 Acc. | 1.565 s |

For Task 1, $K = 20$ provided the highest F1 score among the tested configurations. Increasing the pool to 30 candidates increased inference time without improving retrieval quality, while the smaller and dynamic configurations both produced lower F1 scores. A fixed pool of 20 ingredients was therefore retained for the final pipeline.

For Task 2, the dynamic strategy achieved the highest accuracy, but only by a small margin over the fixed Top 7 configuration. The Top 7 setting reached 0.689 accuracy compared with 0.691 for the dynamic strategy, while requiring less average inference time and providing a simpler deterministic candidate budget. It also improved accuracy over both Top 3 and Top 5. For these reasons, $K = 7$ was selected for the final Stage 4-E caption retrieval pipeline.

#### 5.7.4. VLM Prompts

The prompts provided to Qwen3-VL-4B-Instruct (Task 1) and Qwen3-VL-Reranker (Task 2) were kept fixed during all final experiments; their exact text is reported in Appendix A.

For Task 1, the prompt instructs the model to classify each candidate ingredient from the SigLIP2 list into `present` (clearly visible), `possible` (partially hidden or ambiguous), or not visible, returning a structured JSON output. Only the ingredients classified as `present` are used in the final prediction. Including the `possible` category serves as an intermediate uncertainty buffer: in preliminary tests on the auxiliary development subset, asking the model for a binary selection still produced substantial overprediction, whereas separating clearly visible from uncertain candidates made the `present` assignments more selective. The `possible` predictions are discarded because even a weak weight tended to introduce false positives. For Task 2, the reranker jointly evaluates the query image and each

candidate text and assigns a relevance score based on dish identity, ingredients, cooking method, sauce, and plating details.

## 6. Results

Performance was evaluated on the CVPR Dishcovery VLM MetaFood Challenge benchmark [1]. Task 1 was measured using F1-score for multi-label ingredient retrieval, while Task 2 was evaluated using Top-1 accuracy for dense-caption retrieval. The final challenge score was computed as the harmonic mean of the two task metrics, as defined in Equation (1).

Since Stage 5 (Section 5.6) was designed mainly as a leaderboard-oriented upper bound, the edge-oriented analysis uses Stage 4 (Section 5.4) and Stage 4-E (Section 5.5) as the reference deployment-feasible pipelines.

In addition to the official challenge benchmark, the two tasks were evaluated independently on the auxiliary datasets introduced in Section 3.1. MM-Food-100K was used to evaluate the ingredient-retrieval component, while Food500-Cap was used to evaluate the dense-caption retrieval component. These auxiliary evaluations were reported separately, without computing a harmonic mean, because they were used to analyze the two subtasks independently.

For MM-Food-100K, performance was measured using multi-label F1-score between the predicted ingredient set and the ground-truth annotations. For Food500-Cap, performance was measured using Top-1 accuracy, considering a prediction correct when the ground-truth caption was retrieved at rank one. The fixed Stage 4-E hyperparameters, candidate-pool selection procedure, and VLM prompts are specified in Section 5.7.

### *6.1. Ablation Study Dishcovery Challenge*

The contribution of each architectural stage was evaluated through a progressive ablation study, as reported in Table 4.

**Table 4.** Progressive ablation study of the proposed retrieval pipeline.

| Pipeline | Task 1 Configuration | Task 2 Configuration | Score |
|---|---|---|---|
| Stage 1 | SigLIP2 only | SigLIP2 only | 0.519 |
| Stage 2 | SigLIP2 + open-choice VLM | SigLIP2 + open-choice VLM | 0.579 |
| Stage 3 | SigLIP2 + closed-choice VLM | SigLIP2 + open-choice VLM | 0.583 |
| Stage 4-E | SigLIP2 + closed-choice VLM | SigLIP2 + Qwen3-VL-Reranker-2B | 0.621 |
| Stage 4 | SigLIP2 + closed-choice VLM | SigLIP2 + Qwen3-VL-Reranker-8B | 0.635 |
| Stage 5 | SigLIP2 + ViT + open-choice VLM + closed-choice VLM | SigLIP2 + ViT + open-choice VLM + Qwen3-VL-Reranker-8B | 0.681 |

Table 4 reports a monotonic improvement across the proposed pipeline stages. The score increases from 0.519 with the SigLIP2-only baseline to 0.681 with the leaderboard-oriented Stage 5 configuration. The deployment-oriented Stage 4-E configuration reaches a score of 0.621.

### *6.2. MM-Food-100K Task 1 Evaluation*

Task 1 was evaluated on the Dishcovery-aligned subset of MM-Food-100K comprising 6756 images. Two analyses were performed: a comparison of three vocabulary configurations (Table 5) and an evaluation of the confidence-gated policy used in Stage 4-E (Table 6). In Table 5, *Original* evaluates the 399-label pipeline on the uncleaned annotations, *Harmonized* maps the original 399-label predictions post-hoc to the 295-label space for eval-

uation against the cleaned ground truth, and *Cleaned retrieval* performs both retrieval and evaluation directly with the 295-label vocabulary.

**Table 5.** Task 1 performance under different vocabulary and annotation configurations.

| Configuration | Labels | Precision | Recall | F1 |
|---|---|---|---|---|
| Original | 399 | 0.613 | 0.765 | 0.681 |
| Harmonized | 295 | 0.757 | 0.807 | 0.781 |
| Cleaned retrieval | 295 | 0.769 | 0.798 | 0.783 |

**Table 6.** Effect of confidence-gated closed-choice selection on Task 1 performance.

| Pipeline | F1 | Skipped Images | Closed-Choice Calls |
|---|---|---|---|
| Stage 4 | 0.783 | 0 | 6756 |
| Stage 4-E | 0.780 | 3359 | 3397 |

### *6.3. Food500-Cap Task 2 Evaluation*

To evaluate the dense-caption retrieval component independently from the hidden challenge leaderboard, the Task 2 pipelines Stage 4 and Stage 4-E were tested on the Food500-Cap dataset. The experiment used SigLIP2 as the first-stage image-to-caption retriever and Qwen3-VL-Reranker as a late-stage multimodal reranker. The caption bank contains 4940 candidate captions, and the evaluation was performed on the full available set of 2820 query images. Table 7 reports the main Task 2 retrieval results.

**Table 7.** Task 2 results on Food500-Cap using SigLIP2 retrieval and Qwen3-VL reranking.

| Pipeline | Top-1 Acc. | Reranked Pairs | Reranked Images |
|---|---|---|---|
| SigLIP2 | 0.5961 | 0 | 0 |
| Stage 4-E | 0.7248 | 14,798 | 2114 |
| Stage 4 | 0.7638 | 19,740 | 2820 |

To isolate the effect of reducing the reranker size (8B to 2B) from that of the confidence gate ($\gamma_C \geq 3.0$), an additional $2 \times 2$ ablation study was conducted on the same 2820 queries and 4940 candidate captions ($K = 7$) using a uniform FP32 SigLIP2 and FP16 Qwen setup. Latency was measured sequentially at batch size 1 with resident models after two untimed ungated warm-up queries (14 image-caption pairs) followed by CUDA synchronization, recording a single pass over the evaluation set without averaging repeated full-dataset runs (excluding model loading, caption-bank preparation, and file I/O; gated latencies reflect skipping the measured reranker time on gated queries, validated against direct gated execution). Table 8 reports the resulting Top-1 accuracy, average end-to-end latency per image, and number of reranked images.

**Table 8.** Ablation of reranker size (8B vs. 2B) and confidence gating ($\gamma_C \geq 3.0$) on Food500-Cap (2820 queries, $K = 7$).

| Reranker | Gate | Top-1 Acc. [%] | Avg. Time [s/Img] | Reranked Images |
|---|---|---|---|---|
| Qwen3-VL-8B | Off | 76.38 | 9.036 | 2820 |
| Qwen3-VL-8B | On | 76.24 | 6.807 | 2114 |
| Qwen3-VL-2B | Off | 71.95 | 4.286 | 2820 |
| Qwen3-VL-2B | On | 72.48 | 3.366 | 2114 |

### *6.4. Backbone Comparison and Cross-Benchmark Evaluation*

To validate the choice of the first-stage retriever, SigLIP2 (ViT-gopt-16-SigLIP2-384) [16] was compared against CLIP (ViT-H-14-378-quickgelu) [15]. On the Dishcovery 2026 benchmark [1], standalone visual-only retrieval (Stage 1) achieved a combined score of 0.519 with SigLIP2 and 0.520 with CLIP ViT-H/14. Table 9 reports the candidate retrieval recall at different shortlist sizes (R@$K$) and standalone visual scores on the MM-Food-100K (14,119-image pool) and Food500-Cap (2820 images) datasets.

**Table 9.** First-stage retrieval comparison between SigLIP2 and CLIP ViT-H/14 on MM-Food-100K (Task 1) and Food500-Cap (Task 2).

| | **Task 1 (MM-Food-100K)** | | | **Task 2 (Food500-Cap)** | | |
|---|---|---|---|---|---|---|
| **Encoder** | **R@10 [%]** | **R@20 [%]** | **Visual F1** | **R@1 [%]** | **R@5 [%]** | **R@7 [%]** |
| CLIP ViT-H/14 | 82.53 | 89.89 | 0.624 | 59.82 | 86.60 | 90.92 |
| SigLIP2-384 | 85.37 | 91.81 | 0.648 | 58.30 | 87.52 | 91.70 |

Furthermore, the proposed Stage 4 and Stage 4-E pipelines were evaluated on the CVPR 2025 Dishcovery test set [25] without modifying any hyperparameters, prompts, candidate pool sizes, or confidence thresholds from the 2026 configuration. Table 10 compares the resulting geometric-mean scores and architectural characteristics against the leading methods from the 2025 edition, F4-ITS [19] and LXlearning [20].

**Table 10.** Comparison of the proposed pipelines with reference methods on the CVPR 2025 Dishcovery benchmark [25]. Individual Task 1 and Task 2 scores are not reported by the official competition leaderboard.

| **Method** | **Training-Free** | **Backbone/VLM strategy** | **Edge Deployed** | **2025 Score** |
|---|---|---|---|---|
| F4-ITS [19] | Yes | Contrastive retrieval + open-ended Gemini 2.5 Flash feature fusion | No | 0.5752 |
| LXlearning [20] | No (LoRA) | SigLIP + LoRA-adapted Qwen2.5-VL-7B reranking | No | 0.5704 |
| Stage 4 (Ours) | Yes | SigLIP2 + closed-choice Qwen3-VL-4B/8B reranker | Orin | 0.5731 |
| Stage 4-E (Ours) | Yes | SigLIP2 + confidence-gated Qwen3-VL-4B/2B reranker | Orin & IQ-9075 | 0.5601 |

## 7. Computational Analysis and Edge Deployment

This section complements the retrieval evaluation with an analysis of parameter footprint, approximate weight memory, latency, power consumption, and hardware utilization. The objective is to determine which pipeline configurations represent practical operating points for edge deployment and to evaluate how the selected Stage 4-E architecture (Section 5.5) behaves when optimized for different hardware platforms.

The analysis first considers the parameter and memory requirements of the different pipeline stages. The original Stage 4 (Section 5.4) and Stage 4-E implementations are then evaluated on the NVIDIA Jetson AGX Orin (NVIDIA Corporation, Santa Clara, CA, USA) under different power configurations. Finally, optimized implementations of Stage 4-E are evaluated on the NVIDIA Jetson AGX Orin and Qualcomm Dragonwing IQ-9075 EVK (Qualcomm Technologies, Inc., San Diego, CA, USA) and compared in terms of retrieval performance, inference latency and RAM usage.

This first part of this section reports the parameter footprint of the proposed pipelines. The analysis considers the number of unique model parameters that must be loaded to execute each pipeline. Table 11 summarizes the main models used in the system and their approximate weight memory.

The memory estimates depend on the numerical representation of each model. SigLIP2 and CLIP ViT-H/14 are considered in FP32, corresponding to approximately 4 bytes per parameter, while the Qwen3-VL models are considered in FP16, corresponding to approximately 2 bytes per parameter. These estimates include model weights only and do not include activations, KV cache, framework overhead, or temporary inference buffers.

**Table 11.** Approximate parameter and model weight footprint of the models used in the pipeline. SigLIP2 and CLIP ViT-H/14 are considered in FP32, while the Qwen3-VL models are considered in FP16.

| Model | Parameters [B] | Approx. Memory [GB] |
|---|---|---|
| SigLIP2 ViT-gopt-16 (384 × 384) | 1.87 | 7.49 |
| CLIP ViT-H/14 | 0.99 | 3.94 |
| Qwen3-VL-4B-Instruct | 4.44 | 8.88 |
| Qwen3-VL-Reranker-2B | 2.13 | 4.26 |
| Qwen3-VL-Reranker-8B | 8.77 | 17.53 |

Using these values, Table 12 reports the model footprint of each pipeline configuration. From Stage 4 onward, Task 1 and Task 2 are reported separately because the two tasks require different Qwen components. The values represent the models required during execution of the corresponding pipeline and should not be interpreted as measurements of total system RAM. Actual memory consumption also depends on activations, KV cache, runtime libraries, intermediate tensors, and other implementation overhead.

**Table 12.** Task specific parameter footprint of the different pipeline stages. Stages are reported separately by task whenever the two branches require different models during online inference. The reported memory represents the approximate model weight footprint and does not include activations, KV cache, framework overhead, or temporary inference buffers.

| Pipeline | Task | Models Loaded During Inference | Parameters [B] | Approx. Weight Memory [GB] |
|---|---|---|---|---|
| Stage 1 | Both | SigLIP2 visual tower | 1.16 | 4.65 |
| Stage 2 | Both | SigLIP2 + Qwen3-VL-4B-Instruct | 6.31 | 16.37 |
| Stage 3 | Task 1 | SigLIP2 visual tower + Qwen3-VL-4B-Instruct | 5.60 | 13.53 |
| Stage 3 | Task 2 | SigLIP2 + Qwen3-VL-4B-Instruct | 6.31 | 16.37 |
| Stage 4-E | Task 1 | SigLIP2 visual tower + Qwen3-VL-4B-Instruct | 5.60 | 13.53 |
| Stage 4-E | Task 2 | SigLIP2 visual tower + Qwen3-VL-Reranker-2B | 3.29 | 8.91 |
| Stage 4 | Task 1 | SigLIP2 visual tower + Qwen3-VL-4B-Instruct | 5.60 | 13.53 |
| Stage 4 | Task 2 | SigLIP2 visual tower + Qwen3-VL-Reranker-8B | 9.93 | 22.18 |
| Stage 5 | Task 1 | SigLIP2 + CLIP ViT-H/14 + Qwen3-VL-4B-Instruct | 7.30 | 20.31 |
| Stage 5 | Task 2 | SigLIP2 + CLIP ViT-H/14 + Qwen3-VL-Reranker-8B | 11.63 | 28.96 |

On the NVIDIA Jetson AGX Orin, hardware telemetry for the unoptimized Stage 4 and Stage 4-E pipelines (Table 13) was collected during a single-pass online inference loop over the evaluation set using `tegrastats` at a nominal 200 ms sampling interval, after initial model, engine, and embedding initialization (which are excluded from per-query latency, without a separate discarded warm-up loop or multi-run averaging; the same single-pass post-initialization protocol applies to the optimized deployments in Sections 7.1 and 7.2). Reported power corresponds to the mean sum of the `VDD_GPU_SOC` and `VDD_CPU_CV` rails (rather than total board power), while Energy per query [J] is derived as the product of average rail power and per-query latency. CPU and GPU utilizations represent system-wide

95th-percentile (P95) values across online cores and the GR3D_FREQ engine, respectively. Continuous thermal monitoring (every 200 ms) across the 350-image benchmarks showed that CPU and GPU temperatures remained stable between 51 °C and 61.4 °C for both the unoptimized Stage 4 (peak 56.2 °C for Task 1 and 61.4 °C for Task 2) and the optimized Stage 4-E pipelines (peak 57.2 °C for Task 1 and 59.5 °C for Task 2). Since these values are well below the platform's thermal throttling threshold (~95 °C), no thermal throttling occurred during the experiments.

**Table 13.** Jetson AGX Orin power, latency, energy per query, and P95 CPU/GPU utilization for the original retrieval pipelines across the 15 W, 30 W, and 50 W power modes, evaluated on 350 images.

| Mode | Task | Pipeline | Avg. Power [W] | Latency [s] | Energy [J] | P95 CPU [%] | P95 GPU [%] |
|---|---|---|---|---|---|---|---|
| 15 W | Task 1 | Stage 4 | 7.581 | 11.849 | 89.83 | 15.67 | 99.0 |
| 15 W | Task 1 | Stage 4-E | 7.555 | 9.764 | 73.77 | 15.00 | 99.0 |
| 15 W | Task 2 | Stage 4 | 7.688 | 34.950 | 268.70 | 26.15 | 99.0 |
| 15 W | Task 2 | Stage 4-E | 7.125 | 12.454 | 88.73 | 12.00 | 99.0 |
| 30 W | Task 1 | Stage 4 | 11.258 | 6.929 | 78.01 | 18.67 | 99.0 |
| 30 W | Task 1 | Stage 4-E | 11.195 | 5.731 | 64.16 | 21.32 | 99.0 |
| 30 W | Task 2 | Stage 4 | 11.441 | 20.638 | 236.12 | 21.69 | 99.0 |
| 30 W | Task 2 | Stage 4-E | 10.768 | 7.024 | 75.63 | 16.44 | 99.0 |
| 50 W | Task 1 | Stage 4 | 11.716 | 6.362 | 74.54 | 32.50 | 99.0 |
| 50 W | Task 1 | Stage 4-E | 11.413 | 5.364 | 61.22 | 29.50 | 99.0 |
| 50 W | Task 2 | Stage 4 | 21.204 | 8.985 | 190.52 | 34.80 | 99.0 |
| 50 W | Task 2 | Stage 4-E | 19.023 | 3.528 | 67.11 | 15.80 | 99.0 |

### *7.1. Optimized Mixed Precision Deployment on Jetson AGX Orin*

After selecting Stage 4-E as the main deployment oriented configuration, the complete pipeline was optimized for execution on the NVIDIA Jetson AGX Orin 64GB. Rather than applying the same numerical precision to every component, each model was converted according to its architecture, sensitivity to reduced precision, and the support provided by the available runtime. Table 14 summarizes the final configuration.

**Table 14.** Precision and runtime configuration of the models used in the optimized Stage 4-E deployment on the NVIDIA Jetson AGX Orin 64 GB.

| Model | Runtime | Precision |
|---|---|---|
| SigLIP2 ViT-gopt-16 (384 × 384) | TensorRT | FP16 |
| Qwen3-VL-4B-Instruct | TensorRT EdgeLLM | Hybrid INT4/FP16 |
| Qwen3-VL-Reranker-2B | TensorRT EdgeLLM | W4A16 AWQ |

For SigLIP2, only the visual encoder is executed during online inference through a TensorRT FP16 engine, while the ingredient and caption text embeddings are computed in advance and cached. This avoids repeatedly executing the text encoder for the fixed retrieval collections.

The Qwen3-VL-4B-Instruct deployment uses INT4 weights for the language model feed forward components, while the visual tower and KV cache remain in FP16. The Qwen3-VL-Reranker-2B uses W4A16 AWQ quantization [29] with 4 bit weights, 16 bit activations, and a group size of 128. Its visual tower, token embeddings, language model head, and KV cache remain at higher precision.

The resulting configuration should therefore be considered an optimized mixed precision deployment rather than a uniform conversion of the complete pipeline to a single low precision format. Different components retain higher precision when reducing it provides limited computational benefit or produces a larger degradation in model behaviour.

Table 15 reports the results obtained with the optimized Stage 4-E deployment. The score changes are computed with respect to the corresponding Stage 4-E retrieval results reported in Section 6. The latency and RAM changes compare the optimized implementation with the original Stage 4-E deployment on the Jetson AGX Orin at 50 W.

These measurements include the complete online inference pipeline and therefore reflect the combined effect of model precision, model conversion, and runtime optimization. They should not be interpreted as an isolated measurement of the effect of quantization.

System RAM usage on the Jetson AGX Orin was recorded from the tegrastats system-wide RAM used field during inference.

**Table 15.** Results of the optimized Stage 4-E deployment on the NVIDIA Jetson AGX Orin 64 GB. Score changes are reported relative to the corresponding Stage 4-E retrieval results, while latency and RAM changes are reported relative to the original Stage 4-E deployment at 50 W.

| Task | Score (Δ) | Avg. Time (Δ) | RAM Usage (Δ) |
|---|---|---|---|
| Task 1 | 0.773 F1 (−0.007) | 0.923 s (5.81× faster) | 9.22 GiB (−6.51 GiB) |
| Task 2 | 0.7004 Acc. (−0.0244) | 1.311 s (2.69× faster) | 6.21 GiB (−3.78 GiB) |

For Task 1, the optimized deployment reduces the average processing time from 5.364 s to 0.923 s while decreasing the F1 score from 0.780 to 0.773. For Task 2, the average processing time decreases from 3.528 s to 1.311 s, while Top 1 accuracy decreases from 0.7248 to 0.7004. The results therefore show that the optimized deployment substantially reduces inference time and memory consumption while preserving most of the retrieval performance of the original Stage 4-E configuration.

### *7.2. Optimized Deployment on Qualcomm Dragonwing IQ-9075 EVK*

The Stage 4-E pipeline was also adapted for execution on the Qualcomm Dragonwing IQ-9075 EVK. The retrieval procedure, candidate sizes, confidence gates, and score fusion parameters were kept unchanged. The deployment work therefore focused on adapting the three neural models to the hardware and software stack available on the Qualcomm platform.

As for the Jetson deployment, a single numerical precision was not applied to all components. Each model was configured independently according to its architecture and the execution backends supported by the IQ-9075 EVK. Table 16 summarizes the final configuration.

**Table 16.** Precision and runtime configuration of the models used in the optimized Stage 4-E deployment on the Qualcomm Dragonwing IQ-9075 EVK.

| Model | Precision | Runtime Configuration |
|---|---|---|
| SigLIP2 ViT-gopt-16 (384 × 384) | FP16 | ONNX Runtime with the QNN execution provider on the HTP. Only the visual encoder is executed during inference, while text embeddings are precomputed and cached. |
| Qwen3-VL-4B-Instruct | INT8/INT16 | QAIRT and QNN execution on the HTP through GenieX. Language weights use INT8 with selected INT16 exceptions, activations use INT16, the KV cache uses INT8, and the visual encoder uses W8A16. |
| Qwen3-VL-Reranker-2B | Q8_0/FP16 | HTP execution for the transformer and KV cache, Adreno OpenCL for the multimodal projector, and CPU execution for the unsupported vision upscale operation. |

For SigLIP2, only the visual encoder is executed during online inference, while the ingredient and caption text embeddings are computed in advance and cached. The visual encoder is kept in FP16 and executed through ONNX Runtime using the QNN execution provider on the HTP.

The Qwen3-VL-Reranker-2B used for Task 2 adopts a `Q8_0` representation for the transformer, while the token embeddings, classifier head, and multimodal projector remain in FP16. The transformer and KV cache are executed on the HTP. The multimodal projector is executed on the Adreno GPU through OpenCL, while CPU execution is required only for the unsupported vision upscale operation.

For Task 1, the official W4A16 Qwen3-VL-4B-Instruct model available through Qualcomm AI Hub was replaced by a promoted INT8/INT16 configuration. This configuration improves the F1 score by approximately 3.7 percentage points compared with W4A16. Most of the improvement comes from recall, which increases from 0.593 to 0.657, while precision remains almost unchanged.

The promoted configuration also reduces the execution time of the Qwen3-VL-4B-Instruct branch. When the VLM is invoked, its average processing time decreases from approximately 3.38 to 3.11 s per image. The W4A16 model occasionally generates excessively long or malformed responses, increasing the number of generated tokens and the corresponding inference time. The promoted configuration instead produces shorter and more stable structured outputs.

These values refer specifically to queries for which the Qwen branch is executed. The average latency of the complete Stage 4-E pipeline is lower because the confidence gate avoids the VLM call when the SigLIP2 prediction is sufficiently confident. Using the final INT8/INT16 configuration, the complete Task 1 pipeline reaches an average latency of 2.795 s per image.

Table 17 reports the final retrieval results obtained on the Qualcomm Dragonwing IQ-9075 EVK. On the IQ-9075 EVK, reported RAM usage represents the board-level additional memory footprint above the pre-benchmark baseline (sampled from /proc/meminfo as MemTotal-MemAvailable every 50 ms, since process-level PSS undercounts shared FastRPC and DMA-BUF accelerator allocations). Continuous thermal monitoring across the 350-image benchmarks showed stable SoC temperatures (mean/peak of 53.6/54.4 °C for Task 1 and 53.3/54.3 °C for Task 2, with NSP sensor peaks of 75.5 °C and 69.3 °C, respectively) and zero cooling-state increases, confirming the absence of thermal throttling. Power consumption and CPU/GPU utilization time series are not reported because readable hardware power rails (`hwmon`) and unified utilization counters were not exposed in the target EVK runtime environment.

**Table 17.** Results of the optimized Stage 4-E deployment on the Qualcomm Dragonwing IQ-9075 EVK.

| Task | Score | Avg. Time | RAM Usage |
|---|---|---|---|
| Task 1 | 0.732 F1 | 2.795 s | 11.06 GiB |
| Task 2 | 0.694 Acc. | 9.108 s | 7.60 GiB |

### *7.3. Hardware Specifications and Cross-Platform Deployment Results*

The optimized Stage 4-E pipeline was evaluated on two different edge AI platforms: the NVIDIA Jetson AGX Orin 64GB and the Qualcomm Dragonwing IQ-9075 EVK. The two devices differ substantially in accelerator architecture, memory capacity, supported numerical formats, and software stack. Table 18 summarizes the hardware and software characteristics that are most relevant to the deployments considered in this work.

**Table 18.** Main hardware and software characteristics of the NVIDIA Jetson AGX Orin 64 GB and Qualcomm Dragonwing IQ-9075 EVK relevant to AI inference.

| Characteristic | Jetson AGX Orin 64 GB | Dragonwing IQ-9075 EVK |
|---|---|---|
| AI compute | Up to 275 INT8 TOPS | Up to 100 dense INT8 TOPS |
| GPU | NVIDIA Ampere, 2048 CUDA cores and 64 Tensor Cores | Qualcomm Adreno 663, up to 800 MHz |
| GPU FP32 compute | Approximately 5.3 TFLOPS | Not specified by the manufacturer |
| Dedicated AI accelerator | 2 NVDLA v2 accelerators | Qualcomm Hexagon NPU |
| System memory | 64 GB LPDDR5 | 36 GB LPDDR5 with inline ECC |
| Memory interface | 256 bit LPDDR5, 204.8 GB/s | 96 bit LPDDR5 |
| CPU | 12 core Arm Cortex A78AE, up to 2.2 GHz | 8 core Qualcomm Kryo Gen 6, up to 2.36 GHz |
| Main AI software stack | CUDA and TensorRT | QAIRT, QNN and GenieX |
| Main accelerator used in this work | NVIDIA GPU | Qualcomm Hexagon NPU, with additional GPU and CPU execution for selected operations |

The Jetson AGX Orin provides a larger memory capacity and a GPU architecture designed for highly parallel general purpose and tensor computation. The IQ-9075 instead provides a dedicated Hexagon NPU for neural network execution, together with an Adreno GPU and CPU resources used for operations that cannot be executed directly on the HTP.

The peak compute values reported for the two devices are not directly comparable because they refer to different accelerator architectures, numerical formats, and software execution paths. The measured pipeline latency therefore depends not only on the nominal hardware capability but also on model conversion, operator support, precision, memory movement, and the efficiency of the corresponding inference runtime.

Another relevant difference concerns the input shapes supported by the visual components. The SigLIP2 retrieval encoder is a separate branch and operates at its fixed 384 × 384 input resolution on both platforms. The 512 × 512 constraint discussed here applies instead to the Qwen visual graph used on the Qualcomm NPU path. On the Orin, the corresponding TensorRT deployment can use runtime dimensions through optimization profiles, so the Qwen visual processing is not tied to a single compiled spatial shape.

The QNN graph used for the Qwen visual component on the IQ-9075 is compiled for a fixed 512 × 512 input. Query images passed to this branch must therefore be adapted to that resolution before NPU execution. Consequently, an image whose original resolution is 256 × 256 is still processed through the 512 × 512 Qwen execution graph. This fixed-shape requirement does not affect the SigLIP2 retrieval branch, which remains 384 × 384, but it can add visual computation to Qwen-based selection or reranking for smaller source images compared with the more flexible TensorRT execution used on the Orin.

Table 19 reports the final results obtained with the optimized Stage 4-E configurations described in Sections 7.1 and 7.2.

**Table 19.** Comparison of the optimized Stage 4-E pipeline on the NVIDIA Jetson AGX Orin 64 GB and Qualcomm Dragonwing IQ-9075 EVK.

| Task | Jetson AGX Orin | | | Dragonwing IQ-9075 EVK | | |
|---|---|---|---|---|---|---|
| | Score | Avg. Time [s] | RAM [GiB] | Score | Avg. Time [s] | RAM [GiB] |
| Task 1 | 0.773 F1 | 0.923 | 9.22 | 0.732 F1 | 2.795 | 11.06 |
| Task 2 | 0.7004 Acc. | 1.311 | 6.21 | 0.694 Acc. | 9.108 | 7.60 |

The comparison shows that the same Stage 4-E retrieval logic can be deployed on both platforms without changing the candidate sizes, confidence gates, or final selection rules. The implementation required different numerical representations and execution backends, but the retrieval procedure itself remained unchanged. This provides a direct example of the separation between the retrieval architecture and the hardware specific deployment choices introduced in Section 5.

## 8. Discussion

This section interprets the experimental results reported in Section 6 and relates them to the computational analysis presented in Section 7. The discussion focuses on the accuracy and computational cost tradeoff between the proposed pipeline variants, the effect of the optimized deployments, the comparison between the two edge platforms, the role of vocabulary cleaning in ingredient retrieval, and the main sources of error in the final system, as well as a comparison with reference methods.

### *8.1. Accuracy and Computational Cost Tradeoff*

The ablation study reported in Table 4 shows that the highest overall score is obtained by Stage 5 (Section 5.6), which reaches 0.681 on the Dishcovery benchmark. However, this configuration should be interpreted as a leaderboard oriented upper bound rather than as the main solution for edge deployment. Its additional CLIP branch, ingredient correlation prior, and extra open choice VLM call improve accuracy, but also increase query time computation, memory requirements, and system complexity. For this reason, the most relevant deployment comparison remains the one between Stage 4 (Section 5.4) and Stage 4-E (Section 5.5).

The main deployment result is that Stage 4-E substantially reduces the computational requirements of the pipeline while retaining most of the retrieval performance of Stage 4. Stage 4 reaches a score of 0.635 on the Dishcovery benchmark and a Top-1 accuracy of 0.7638 (95% Wilson CI: $[0.7478, 0.7791]$) on Food500-Cap for Task 2, while Stage 4-E reaches 0.621 and 0.7248 (95% CI: $[0.7080, 0.7410]$; paired bootstrap difference of +3.90 percentage points, 95% CI: $[+2.55, +5.28]$), respectively (Tables 4 and 7). For Task 1, the difference is smaller: on the MM-Food-100K evaluation subset, Stage 4 reaches 0.783 F1 (95% bootstrap CI: $[0.776, 0.790]$), while Stage 4-E reaches 0.780 F1 (95% CI: $[0.772, 0.787]$) after skipping the closed-choice selector for 49.72% of the images (Table 6). A paired bootstrap comparison yields a difference of $-0.0032$ F1 (95% CI: $[-0.0074, +0.0011]$ across the two archived runs, or $-0.0057$ F1 with 95% CI $[-0.0076, -0.0039]$ when isolating the gate on the exact same Stage 4 predictions), confirming that bypassing nearly half of the VLM calls introduces at most a sub-percent F1 reduction. Therefore, the efficiency oriented variant loses part of the full reranking accuracy in Task 2, but preserves almost the same ingredient retrieval performance in Task 1.

The computational advantage of Stage 4-E differs between the two tasks. For Task 1, Stage 4 and Stage 4-E require the same model set, corresponding to approximately 5.60 billion parameters and 13.53 GB of model weights (Table 12). The reduction in computational cost therefore comes primarily from the confidence gate, which avoids executing Qwen3-VL for sufficiently confident queries.

For Task 2, Stage 4-E combines two modifications: replacing the 8B reranker with the 2B variant and introducing the confidence gate. Since the reranker must remain loaded in memory regardless of the confidence gate, memory savings depend entirely on the model choice, which lowers the required weight footprint from 9.93B parameters (22.18 GB) to 3.29B parameters (8.91 GB), as reported in Table 12. Furthermore, the $2 \times 2$ ablation in Table 8 shows that the accuracy gap between Stage 4 and Stage 4-E is attributable to

this model size reduction rather than to confidence gating: switching from the 8B to the 2B reranker reduces Top-1 accuracy by 4.43 percentage points without gating (76.38% [74.78, 77.91] to 71.95% [70.26, 73.58]) and by 3.76 points with gating (76.24% [74.64, 77.78] to 72.48% [70.80, 74.10]), while cutting per-query latency by more than half (9.036 s to 4.286 s). Conversely, enabling the confidence gate skips reranking on 25.04% of the queries (706 of 2820 images) and reduces average latency by a further 24.7% for the 8B model and 21.5% for the 2B model, without degrading accuracy (−0.14 percentage points with 95% paired bootstrap CI [−0.60, +0.32] for 8B, and +0.53 percentage points with 95% CI [+0.04, +1.06] for 2B, where skipping high-confidence SigLIP2 predictions prevents occasional reranker errors).

This difference is also visible in the latency measurements obtained with the original Jetson implementation. At the 50 W power mode, Task 1 latency decreases from 6.362 to 5.364 s per query (a 15.7% reduction, lowering per-query energy from 74.54 J to 61.22 J, or −17.9%). For Task 2, latency decreases from 8.985 to 3.528 s per query (a 60.7% reduction, lowering per-query energy from 190.52 J to 67.11 J, or −64.8%, Table 13). The larger reduction observed for Task 2 results from the combined effect of guarded reranking and the replacement of the 8B reranker with the smaller 2B model. In contrast, the Task 1 improvement is obtained primarily by skipping Qwen3-VL closed choice selection for confident queries.

A further reduction in computational cost is obtained through the optimized mixed precision deployment described in Section 7.1. On the Jetson AGX Orin, the optimized configuration reduces Task 1 latency to 0.923 s per query and RAM usage to 9.22 GiB, while the F1 score decreases by only 0.007. For Task 2, latency is reduced to 1.311 s per query and RAM usage to 6.21 GiB, with an accuracy decrease of 0.0244 (Table 15).

These improvements result from the combined effect of reduced precision, model conversion, and runtime optimization rather than from quantization alone. The deployment keeps selected components at higher precision while using reduced precision for the computationally expensive parts of the Qwen models. This mixed precision strategy provides substantial latency and memory savings while limiting the degradation in retrieval performance.

Overall, the results suggest that efficient edge deployment requires reducing both the number and the computational cost of expensive VLM calls. For confident examples, SigLIP2 retrieval is often sufficient, while Qwen based selection or reranking is most useful for ambiguous cases. When a VLM call is required, the optimized mixed precision execution further reduces its computational and memory requirements. Stage 4 therefore remains the accuracy oriented configuration, whereas Stage 4-E provides a more practical operating point when latency, memory, or power constraints are more important.

### *8.2. Cross-Platform Tradeoffs and Runtime Analysis*

The comparison between the two optimized deployments shows that hardware specifications alone are not sufficient to characterize the execution of a multimodal retrieval pipeline. Although both devices provide dedicated resources for neural network inference, the implementations use different accelerators, numerical representations, runtime libraries, and execution strategies. The measured differences therefore reflect the complete hardware and software configuration rather than only the peak computational capability of the processors.

For Task 1, the IQ-9075 requires 2.795 s per image compared with 0.923 s on the Jetson AGX Orin, corresponding to approximately 3.03× the latency (Table 19). The F1 score decreases from 0.773 on the Orin to 0.732 on the IQ-9075. The analysis performed during the Qualcomm optimization showed that the SigLIP2 candidate retrieval stage produced comparable results across the two platforms, while a larger difference appeared after the

Qwen3-VL-4B-Instruct selection stage. This suggests that differences in the numerical configuration and execution of the VLM may contribute to the remaining Task 1 accuracy gap.

For Task 2, the retrieval performance of the two deployments is similar, with a Top 1 accuracy of 0.7004 on the Orin and 0.694 on the IQ-9075. The latency difference is considerably larger, increasing from 1.311 to 9.108 s per image, corresponding to approximately 6.95× the latency on the IQ-9075 EVK. Task 2 depends more heavily on Qwen3-VL-Reranker-2B, which evaluates multiple image and caption pairs for queries that pass the confidence gate. Its execution on the IQ-9075 is distributed across the HTP, Adreno GPU, and CPU, whereas the Jetson implementation uses a more unified TensorRT execution path.

The fixed 512 × 512 QNN visual graph input shape on the IQ-9075 (Section 7.3), compared with dynamic TensorRT optimization profiles on the Orin, also contributes to the higher Task 2 latency.

The measured latency therefore reflects several factors, including numerical precision, operator placement, input resolution, runtime support, and the degree of optimization available for each model. In particular, the Qualcomm deployment combines HTP, GPU, and CPU execution for selected components, while the Jetson implementation uses optimized TensorRT execution for the models considered in this work.

The optimized deployments also differ in measured memory usage. For Task 1, RAM usage is 9.22 GiB on the Jetson AGX Orin and 11.06 GiB on the IQ-9075 EVK, while for Task 2 the corresponding values are 6.21 GiB and 7.60 GiB. The less aggressive weight quantization used on the Qualcomm platform is likely to contribute to its larger measured memory footprint. For Task 1, the Qwen3-VL-4B-Instruct model primarily uses 8-bit weights with 16-bit activations, whereas the Orin deployment uses a hybrid INT4/FP16 configuration. Similarly, for Task 2, the Qualcomm reranker uses a Q8_0 representation, while the Orin implementation adopts W4A16 AWQ.

However, the measured RAM values cannot be explained from model weight precision alone. They represent the complete deployed pipeline and also include activations, KV cache, intermediate tensors, runtime libraries, and hardware-specific execution buffers. Moreover, the two platforms use substantially different inference stacks and operator mappings. The memory measurements should therefore be interpreted as system-level properties of the respective optimized deployments rather than as a direct comparison of model weight size alone.

Despite these differences, both platforms are able to execute the complete Stage 4-E pipeline locally. The results demonstrate that the same high-level retrieval procedure can be preserved across different edge accelerators even when the numerical representations and execution backends must be adapted to the target hardware. At the same time, preserving similar retrieval quality does not necessarily result in comparable inference latency or memory usage.

### *8.3. The Role of Vocabulary Normalization in Ingredient Retrieval*

Vocabulary normalization plays an important role in Task 1, where retrieval operates over a predefined set of ingredient labels. Redundant, ambiguous, or inconsistent categories can complicate both candidate selection and evaluation. By consolidating semantically overlapping labels and removing ambiguous categories, preprocessing provides a more coherent representation of the ingredient space and helps align model predictions with ground-truth annotations.

The MM-Food-100K experiments highlight the importance of this alignment. While the original evaluation achieved an F1 of 0.681, harmonizing its predictions with the cleaned label space increased the score to 0.781, close to the 0.783 obtained by performing retrieval directly with the cleaned vocabulary. The small difference between the latter two config-

urations (+0.002 F1, with a 95% paired bootstrap confidence interval of [−0.003, +0.007] over 10,000 resamples) falls within statistical uncertainty, providing no clear evidence of a retrieval advantage from the cleaned configuration. Moreover, differences in historical configuration settings prevent isolating the effect of vocabulary normalization alone.

These findings emphasize that vocabulary preprocessing should be considered an integral part of reliable ingredient retrieval, primarily by ensuring semantic consistency and meaningful evaluation rather than necessarily improving visual recognition performance. This distinction is particularly relevant in benchmarks such as Dishcovery, where the fixed vocabulary and hidden ground truth make it difficult to identify and resolve potential mismatches between predicted and expected labels.

### *8.4. Error Analysis*

To better understand the limitations of the final deployment oriented pipeline, the optimized Stage 4-E configurations for Task 1 and Task 2 were analyzed using their complete evaluation traces. The analysis separates errors introduced during initial candidate retrieval from those occurring during the subsequent VLM selection or reranking stages.

#### 8.4.1. Task 1: Ingredient Retrieval

For Task 1, the optimized Stage 4-E deployment reaches an F1 score of 0.773 on the 6756 image evaluation set. Since SigLIP2 is used as an intermediate retriever rather than as the final multi label classifier, its main limitation can be characterized through ground truth candidate recall. The retrieved candidate lists contain 10,441 of the 11,362 annotated ingredient occurrences, corresponding to a candidate recall of 91.89%. Consequently, 921 true ingredient occurrences are absent from the candidate set and cannot be recovered by the subsequent closed choice VLM.

Attributing the false negatives to the stage where they originate shows that errors are distributed across several parts of the pipeline. SigLIP2 retrieval misses account for 33.80% of the final false negatives, while 28.95% correspond to correctly retrieved ingredients that are subsequently rejected by Qwen3-VL. A further 18.64% are ingredients marked as present by the VLM but removed by the final fusion and thresholding stage, and 18.61% originate from the visual only path used when the confidence guard skips the VLM. Therefore, no single component completely dominates false negatives, and improving candidate recall alone would not eliminate most of the remaining errors.

A clearer pattern emerges for false positives. Of the 2349 final false positives, 2217 (94.38%) correspond to candidates explicitly classified as present by the closed choice VLM. This indicates that distinguishing difficult negative candidates is the principal source of overprediction. The final fusion stage partially corrects this behaviour by increasing precision, but does so at the cost of removing a non negligible number of valid ingredients confirmed by the VLM. Future improvements should therefore consider both stronger discrimination among visually similar candidates in the closed choice stage and better calibrated final selection, rather than simply increasing the number of retrieved candidates.

The pipeline also exhibits a mild tendency to predict too few ingredients, with performance decreasing for images containing larger numbers of annotated ingredients. A more adaptive estimate of the number of output ingredients and an evaluation that accounts for hierarchical relationships between labels could therefore provide useful directions for future work.

#### 8.4.2. Task 2: Dense Caption Retrieval

For Task 2, the optimized Stage 4-E deployment achieves a Top 1 accuracy of 70.04% on the 2820 image Food500-Cap evaluation set. However, the correct caption appears among

the seven SigLIP2 candidates for 92.30% of the images. Of the 845 final errors, only 217 (25.68%) are caused by the correct caption being absent from the candidate set, whereas 628 (74.32%) occur despite the correct caption being available to the reranker. This indicates that selecting the final caption from a set of similar candidates is currently a larger bottleneck than the initial candidate retrieval stage.

The difference between exact caption accuracy and dish class accuracy further clarifies the nature of these errors. While exact caption accuracy is 70.04%, dish class accuracy reaches 88.55%. In 522 of the 845 exact caption errors, the selected caption belongs to the correct dish class but corresponds to a different description. Many remaining failures therefore arise from the difficulty of distinguishing among semantically and visually similar captions, rather than from a complete failure to recognize the dish.

The confidence based reranking strategy nevertheless provides a substantial benefit. The guard bypasses the reranker on 780 images with high SigLIP2 confidence, while ambiguous examples are passed to Qwen3-VL-Reranker. Among the queries processed by the reranker, it corrects 519 cases in which the first SigLIP2 prediction was wrong, but it also replaces a correct first prediction with an incorrect caption in 225 cases. Overall, the complete pipeline combining the confidence guard and reranker improves exact caption accuracy from 59.61% for SigLIP2 alone to 70.04%.

### *8.5. Comparison with Reference Methods*

The results reported in Section 6.4 contextualize the design choices of the proposed pipeline with respect to both alternative contrastive backbones and top solutions from the CVPR 2025 Dishcovery challenge. Regarding the first-stage retriever, Table 9 shows why SigLIP2 was selected over CLIP ViT-H/14 despite similar standalone baseline scores (0.519 vs. 0.520 on Dishcovery 2026 and 58.30% vs. 59.82% Top-1 accuracy on Food500-Cap). In a two-stage architecture, the primary role of the dual encoder is to maximize candidate recall within the shortlist passed to the VLM. SigLIP2 consistently achieves higher recall at the target shortlist sizes on both MM-Food-100K (91.81% vs. 89.89% at $K = 20$) and Food500-Cap (91.70% vs. 90.92% at $K = 7$), reducing the number of true candidates lost before VLM verification.

This comparison highlights three practical findings. First, both Stage 4 (0.5731) and Stage 4-E (0.5601) were evaluated on the 2025 test set using the exact hyperparameters, prompts, and thresholds selected for the 2026 pipeline, even though the 2025 benchmark uses a 24.1% larger ingredient vocabulary (495 vs. 399 labels) and a different aggregation metric. Achieving performance on par with methods specifically tuned for the 2025 competition suggests that the proposed modular architecture generalizes well across label spaces without dataset-specific overfitting. Second, unlike LXlearning (0.5704) [20], which relies on pseudo-labeling, synthetic data generation, and supervised LoRA fine-tuning of a 7B VLM, our approach operates with strictly frozen weights and still reaches a higher score with Stage 4 (+0.27 percentage points). Third, compared with F4-ITS (0.5752) [19], which relies on cloud VLM text generation (Gemini 2.5 Flash) and feature-level fusion for every query, Stage 4-E trades only 1.30 percentage points relative to Stage 4 (and 1.51 points relative to F4-ITS) to substantially reduce inference cost through confidence gating and a compact 2B reranker, enabling local execution on constrained edge platforms.

## 9. Conclusions and Future Work

This paper presented a training free retrieval pipeline for food image and text understanding under edge computing constraints. The proposed approach combines efficient SigLIP2 candidate retrieval with constrained vision language reasoning and multimodal reranking. For ingredient retrieval, the use of closed choice VLM selection over a restricted

candidate set reduces the ambiguity and hallucination problems associated with open choice generation. For dense caption retrieval, SigLIP2 is used as an efficient first stage retriever, while Qwen3-VL-Reranker is applied only to a small set of candidate captions.

The experimental analysis showed that the different pipeline stages provide different operating points in terms of retrieval performance and computational cost. Stage 5 achieved the highest public Dishcovery challenge score of 0.681, but its additional model branches and query time components make it less suitable for constrained deployment. Stage 4 and Stage 4-E therefore represent the main deployment relevant configurations. Stage 4 remains the accuracy oriented solution, while Stage 4-E reduces computation through confidence based execution and, for Task 2, through the use of a smaller 2B reranker.

The computational advantage of Stage 4-E depends on the task. For Task 1, Stage 4 and Stage 4-E use the same model set, corresponding to approximately 5.60 billion parameters and 13.53 GB of model weights, while Stage 4-E reduces the average computation by skipping the Qwen3-VL selector for sufficiently confident queries. For Task 2, Stage 4-E reduces the model footprint from approximately 9.93 billion parameters and 22.18 GB of weights to 3.29 billion parameters and 8.91 GB by replacing the 8B reranker with the 2B variant and applying confidence based reranking.

Further improvements were obtained through hardware specific deployment optimization. On the NVIDIA Jetson AGX Orin 64GB, the optimized mixed precision Stage 4-E implementation reaches an F1 score of 0.773 with an average latency of 0.923 s for Task 1, and a Top 1 accuracy of 0.7004 with an average latency of 1.311 s for Task 2. The measured RAM usage is 9.22 GiB and 6.21 GiB, respectively. These results show that reduced precision, model conversion, and runtime optimization can substantially reduce latency and memory requirements while preserving most of the retrieval performance.

The same Stage 4-E retrieval procedure was also deployed on the Qualcomm Dragonwing IQ-9075 EVK. The candidate sizes, confidence gates, and final selection rules were kept unchanged, while the numerical representations and execution backends were adapted to the Qualcomm hardware and software stack. The final deployment reaches an F1 score of 0.732 with an average latency of 2.795 s and a measured RAM usage of 11.06 GiB for Task 1, and a Top 1 accuracy of 0.694 with an average latency of 9.108 s and a measured RAM usage of 7.60 GiB for Task 2. This demonstrates that the high-level retrieval architecture can be preserved across different edge accelerators, although the resulting latency and memory footprint depend strongly on numerical precision, runtime support, operator placement, and hardware-specific optimization.

Future work could extend this analysis to a wider range of edge accelerators and memory budgets in order to characterize the accuracy, latency, and memory tradeoff across different hardware classes. In particular, the Task 2 deployment on the IQ-9075 leaves room for further optimization of the multimodal reranker, including improved operator support, more efficient accelerator placement, and alternative numerical representations. Additional work could also investigate more adaptive confidence policies and model compression strategies that further reduce the number and cost of VLM calls.

Finally, because the proposed method relies on frozen general purpose vision language models rather than food specific training, an important direction for future work is to evaluate whether the same retrieval strategy transfers to other domains requiring detailed image and text matching.

**Author Contributions:** Conceptualization, A.S., S.B., R.S. and D.P.; methodology, A.S.; software, A.S.; validation, A.S.; formal analysis, A.S.; investigation, A.S.; data curation, A.S.; writing—original draft preparation, A.S.; writing—review and editing, A.S., S.B., R.S. and D.P.; visualization, A.S.; supervision, S.B., R.S. and D.P. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The source code and deployment scripts used to reproduce the edge inference experiments presented in this study are publicly available on GitHub. The NVIDIA Jetson AGX Orin implementation is available at https://github.com/andreaspagnolo/demo_dishcovery_orin (accessed on 27 October 2026), while the Qualcomm Dragonwing IQ-9075 EVK implementation is available at https://github.com/andreaspagnolo/demo_dishcovery_qualcomm (accessed on 27 October 2026). The complete 399-to-295 ingredient vocabulary mapping table (`vocabulary_mapping_complete.csv`) and the auxiliary dataset construction files are available in the public repositories.

**Acknowledgments:** The authors would like to thank Akshaya Nagendra from Qualcomm Technologies, Inc. for technical support, discussions, and guidance during the deployment and optimization of the proposed pipeline on the Qualcomm Dragonwing IQ-9075 EVK.

**Conflicts of Interest:** Author Danilo Pau was employed by the company STMicroelectronics. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Appendix A. VLM Prompts

### Task 1 (Qwen3-VL-4B-Instruct closed-choice selector).

```
Classify each candidate ingredient by direct visual evidence in this meal
image.
Use this confidence scale:
2 = clearly visible as a physical food item or ingredient
1 = possibly visible, partially hidden, ambiguous, or only weakly supported
0 = not visible
Rules:
Choose only from the candidate IDs below. Do not invent labels or use
synonyms.
Do not infer recipe ingredients, fillings, sauces, seasonings, or cuisine
context unless they are directly visible.
For dish names, preparations, or specific variants, use 2 only when that
exact visual form is clear.
If two candidates are similar and the image does not clearly distinguish
them, put the uncertain candidate in possible, not present.
Return IDs using the 1-based numbers in the candidate list.
Omit all 0 items.
Return only valid JSON with exactly these two keys and no markdown:
{"present": [1,4], "possible": [2]}
If nothing is visible, return:
{"present": [], "possible": []}

Candidate ingredients:
[SigLIP2 candidate list]
```

### Task 2 (Qwen3-VL-Reranker).

```
Carefully compare the food image with the candidate caption.
Score high only when the visible dish, ingredients, cooking method, sauce,
plating, and other visual details match the caption.
```

## References


1. Kaggle. Dishcovery Mission II—CVPR. 2026. Available online: https://www.kaggle.com/competitions/dishcovery-mission-ii-cvpr-2026/overview (accessed on 20 October 2026).
2. Yin, Y.; Qi, H.; Zhu, B.; Chen, J.; Jiang, Y.G.; Ngo, C.W. FoodLMM: A Versatile Food Assistant using Large Multi-modal Model. *arXiv* **2023**, arXiv:2312.14991. [CrossRef]
3. Min, W.; Jiang, S.; Liu, L.; Rui, Y.; Jain, R. A Survey on Food Computing. *ACM Comput. Surv.* **2019**, *52*, 92. [CrossRef]
4. Bossard, L.; Guillaumin, M.; Van Gool, L. Food-101: Mining Discriminative Components with Random Forests. In *Proceedings of the Computer Vision—ECCV 2014*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2014; Volume 8694, pp. 446–461. [CrossRef]
5. Ciocca, G.; Napoletano, P.; Schettini, R. CNN-Based Features for Retrieval and Classification of Food Images. *Comput. Vis. Image Underst.* **2018**, *176–177*, 70–77. [CrossRef]
6. Min, W.; Wang, Z.; Liu, Y.; Luo, M.; Kang, L.; Wei, X.; Wei, X.; Jiang, S. Large Scale Visual Food Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 9932–9949. [CrossRef] [PubMed]
7. Min, W.; Jiang, S.; Sang, J.; Wang, H.; Liu, X.; Herranz, L. Being a Supercook: Joint Food Attributes and Multimodal Content Modeling for Recipe Retrieval and Exploration. *IEEE Trans. Multimed.* **2017**, *19*, 1100–1113. [CrossRef]
8. Salvador, A.; Drozdzal, M.; Giro-i Nieto, X.; Romero, A. Inverse Cooking: Recipe Generation from Food Images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 10453–10462. [CrossRef]
9. Marin, J.; Biswas, A.; Ofli, F.; Hynes, N.; Salvador, A.; Aytar, Y.; Weber, I.; Torralba, A. Recipe1M+: A Dataset for Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 187–203. [CrossRef] [PubMed]
10. Wang, H.; Sahoo, D.; Liu, C.; Shu, K.; Achananuparp, P.; Lim, E.P.; Hoi, S.C.H. Cross-Modal Food Retrieval: Learning a Joint Embedding of Food Images and Recipes With Semantic Consistency and Attention Mechanism. *IEEE Trans. Multimed.* **2022**, *24*, 2515–2525. [CrossRef]
11. Lee, K.H.; Chen, X.; Hua, G.; Hu, H.; He, X. Stacked Cross Attention for Image-Text Matching. In *Proceedings of the Computer Vision—ECCV 2018*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11208, pp. 212–228. [CrossRef]
12. Li, J.; Selvaraju, R.R.; Gotmare, A.D.; Joty, S.; Xiong, C.; Hoi, S.C.H. Align before Fuse: Vision and Language Representation Learning with Momentum Distillation. *arXiv* **2021**, arXiv:2107.07651. [CrossRef]
13. Li, J.; Li, D.; Savarese, S.; Hoi, S. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *Proceedings of the International Conference on Machine Learning*; Proceedings of Machine Learning Research; PMLR: Cambridge, MA, USA, 2023; Volume 202, pp. 19730–19742. Available online: https://proceedings.mlr.press/v202/li23q.html (accessed on 20 October 2026).
14. Li, M.; Zhang, Y.; Long, D.; Chen, K.; Song, S.; Bai, S.; Yang, Z.; Xie, P.; Yang, A.; Liu, D.; et al. Qwen3-VL-Embedding and Qwen3-VL-Reranker: A Unified Framework for State-of-the-Art Multimodal Retrieval and Ranking. *arXiv* **2026**, arXiv:2601.04720. [CrossRef]
15. Radford, A.; Kim, J.W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. Learning Transferable Visual Models from Natural Language Supervision. *arXiv* **2021**, arXiv:2103.00020. [CrossRef]
16. Tschannen, M.; Gritsenko, A.; Wang, X.; Naeem, M.F.; Alabdulmohsin, I.; Parthasarathy, N.; Evans, T.; Beyer, L.; Xia, Y.; Mustafa, B.; et al. SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features. *arXiv* **2025**, arXiv:2502.14786. [CrossRef]
17. Karthik, S.; Roth, K.; Mancini, M.; Akata, Z. Vision-by-Language for Training-Free Compositional Image Retrieval. In Proceedings of the International Conference on Learning Representations (ICLR), Vienna, Austria, 7–11 May 2024. Available online: https://github.com/ExplainableML/Vision_by_Language (accessed on 20 October 2026).
18. Sun, S.; Ye, F.; Gong, S. Training-free Zero-shot Composed Image Retrieval with Local Concept Reranking. *arXiv* **2023**, arXiv:2312.08924. [CrossRef]
19. Asokan, R. F4-ITS: Fine-Grained Feature Fusion for Food Image-Text Search. *arXiv* **2025**, arXiv:2508.17037. [CrossRef]
20. LXlearning. CVPR2025 Dishcovery VLM MetaFood Challenge 1st Solutions. 2025. Available online: https://github.com/LXlearning/CVPR2025-Dishcovery-VLM-MetaFood-Challenge-1st-Solutions (accessed on 20 October 2026).
21. Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; et al. Qwen2.5-VL Technical Report. *arXiv* **2025**, arXiv:2502.13923. [CrossRef]
22. Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. *arXiv* **2022**, arXiv:2106.09685. [CrossRef]
23. Dong, Y.; Muraoka, Y.; Shi, S.; Zhang, Y. MM-Food-100K: A 100,000-Sample Multimodal Food Intelligence Dataset with Verifiable Provenance. *arXiv* **2025**, arXiv:2508.10429. [CrossRef]

24. Ma, Z.; Pan, M.; Wu, W.; Cheng, K.; Zhang, J.; Huang, S.; Chen, J. Food-500 Cap: A Fine-Grained Food Caption Benchmark for Evaluating Vision-Language Models. In Proceedings of the ACM International Conference on Multimedia, Ottawa, ON, Canada, 29 October–3 November 2023; pp. 5674–5685. [CrossRef]
25. Kaggle. Dishcovery VLM MetaFood Challenge—CVPR. 2025. Available online: https://www.kaggle.com/competitions/dishcovery-vlm-mtf-cvpr-2025/overview (accessed on 20 October 2026).
26. Bai, S.; Cai, Y.; Chen, R.; Chen, K.; Chen, X.; Cheng, Z.; Deng, L.; Ding, W.; Gao, C.; Ge, C.; et al. Qwen3-VL Technical Report. *arXiv* **2025**, arXiv:2511.21631. [CrossRef]
27. Salvador, A.; Hynes, N.; Aytar, Y.; Marin, J.; Ofli, F.; Weber, I.; Torralba, A. Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3020–3028. [CrossRef]
28. Bagler, G.; Singh, N. Data-Driven Investigations of Culinary Patterns in Traditional Recipes across the World. In Proceedings of the IEEE International Conference on Data Engineering Workshops, Paris, France, 16–20 April 2018; pp. 157–162. [CrossRef]
29. Lin, J.; Tang, J.; Tang, H.; Yang, S.; Chen, W.M.; Wang, W.C.; Xiao, G.; Dang, X.; Gan, C.; Han, S. AWQ: Activation-aware Weight Quantization for On-Device LLM Compression and Acceleration. In Proceedings of the Machine Learning and Systems, Santa Clara, CA, USA, 13–16 May 2024; Volume 6, pp. 87–100. Available online: https://proceedings.mlsys.org/paper_files/paper/2024/hash/42a452cbafa9dd64e9ba4aa95cc1ef21-Abstract-Conference.html (accessed on 20 October 2026).


**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.