

MangoVQA: A Multi-Answer Visual Question Answering System for Grading SNI-Aligned Mango Quality using Deep Learning and Text Processing

Nidaul Hasanati^{1,2}, Taufik Djatna^{3*}, Imas Sukaesih Sitanggang¹, Arif Imam Suroso⁴

Abstract—Visual Question Answering (VQA), a multimodal AI task combining computer vision and language understanding, answers text-based questions about visual content. Although VQA has developed rapidly, most existing approaches assume a single correct answer, whereas real-world images often require several valid answers simultaneously. This study proposes MangoVQA, to our knowledge the first multi-answer VQA framework for mango quality grading aligned with SNI 3164:2024. MangoVQA separates perception from reasoning, combining YOLOv11n-based neural perception with deterministic, rule-based question interpretation, unlike end-to-end learned-fusion VQA models. Each answer is grounded in detected mango instances, enabling traceable multi-answer responses rather than opaque predictions. The YOLOv11n detector localizes fruits and identifies four quality grades (Extra Class, Class I, Class II, and Reject), while the Perspective Engine extracts color and defect attributes through calibrated HSV analysis. These facts populate a structured answer representation, queried via predefined intents and slots. From 468 annotated images, the pipeline yielded 17,764 question-answer pairs across 22 question types, two reasoning levels, and two languages. 22.85% of questions required multi-answer responses, scored with set-based precision, recall, F1-score, and subset exact match. Evaluation reaches 0.919 accuracy on global open-ended questions, 0.937 on local open-ended questions, and 0.846 micro-F1 on global multi-answer questions, exceeding the 62.91% F1 reported by EMQA, the only prior multi-answer VQA baseline, on a different dataset. Practically, MangoVQA offers auditable decision support for smallholders, cooperatives, exporters

needing SNI-compliant evidence, and mango customers; theoretically, decoupling visual perception from text interpretation and reasoning offers a viable, interpretable alternative to end-to-end multimodal fusion.

Index Terms—Visual question answering, multi-answer, deep learning, rule-based reasoning, text processing, object detection, YOLOv11n, mango quality grading, SNI 3164:2024, interpretable machine learning.

I. INTRODUCTION

Computer vision has proven highly successful in post-harvest fruit handling, where convolutional neural network (CNN) models are now routinely applied to classify and grade harvested produce at scale [1]. The adoption of computer vision-based grading systems aims to address the limitations of manual inspection, which remains time-consuming and susceptible to inconsistency, fatigue, and subjective judgment[1]. Within this space, one-stage object detectors of the YOLO (You Only Look Once) family [2] are the dominant choice, because they localize and classify many fruits in a single forward pass at interactive frame rates, and the recent YOLOv11 generation further improves this accuracy–speed trade-off [3], [4]. Our own earlier work applied YOLOv11n to grade mangoes against the Indonesian National Standard SNI 3164:2024 [5], reaching $mAP@0.5 = 0.9558$ [6].

However, despite their strong recognition capability, conventional object detection systems remain limited in semantic understanding. A bounding box with a class label can identify and localize objects [7]–[11], but it does not inherently answer higher-level questions regarding the visual context. For example, an object detector can identify several mango instances with different quality grades, but it cannot answer questions such as "which grades are present in this tray?", "how many rejected mangoes are located on the left side?", or "is the entire batch suitable for sale?". Answering such questions requires not only object recognition but also interpretation of user queries, aggregation of visual information, and generation of context-dependent responses. Even with multi-class detection, assigning a grade label to each detected mango is not the same as answering a scene-level question about the tray.

Received: 4 August 2026; Revised: 12 August 2026; Accepted: 30 September 2026.

*Corresponding author

¹Nidaul Hasanati, Doctoral Program in Computer Science, School of Data Science, Mathematics, and Informatics, IPB University, Bogor, Indonesia (e-mail: nidaulhasanati@apps.ipb.ac.id).

²Nidaul Hasanati, Department of Information System, UIN Syarif Hidayatullah Jakarta, Indonesia (e-mail: nidaul.hasanati@uinjkt.ac.id).

³Taufik Djatna, Division of Agroindustrial Engineering and Technology, Faculty of Engineering and Technology, IPB University, Bogor, Indonesia (e-mail: taufikdjatna@apps.ipb.ac.id).

⁴Imas Sukaesih Sitanggang, School of Data Science, Mathematics, and Informatics, IPB University, Bogor, Indonesia (email: imas.sitanggang@apps.ipb.ac.id).

⁵Arif Imam Suroso, School of Business, IPB University, Bogor, Indonesia (email: arifmamsuroso@apps.ipb.ac.id).

Multi-class detection alone provides no built-in mechanism to consolidate these per-object labels into a query-driven answer. This gap between visual recognition and semantic interaction motivates the adoption of Visual Question Answering (VQA).

Visual Question Answering (VQA) is a multimodal artificial intelligence task that aims to answer natural-language questions about image content by integrating visual understanding and language interpretation [12]–[16]. Unlike computer vision tasks (such as classification, segmentation, and object detection), where the target output is predefined by the algorithm before inference, VQA introduces a question-driven setting in which the required visual information depends on the query provided by the user at runtime. Therefore, a VQA system must identify relevant visual evidence, interpret the question, and generate an answer according to the requested information. This capability extends computer vision beyond fixed prediction tasks toward interactive visual understanding [15], [16].

Despite rapid progress in VQA, most existing approaches, 83% ($n=48$) of 58 recent studies reviewed for this work, are still designed under the assumption that each question has a single correct answer, as seen in Fig. 1. This assumption is often unsuitable for real-world scenarios involving multiple objects and varying attributes within a single image [17]. For example, as shown in Fig. 2, an image containing several mangoes may simultaneously contain Extra Class, Class I, Class II, and Reject categories. A question such as "what grades are present in this image?" naturally requires multiple valid answers rather than a single output label. Therefore, multi-answer VQA is necessary for applications where visual scenes contain multiple entities and attributes that independently contribute to the final answer.



Fig. 1. VQA with single answer [14].

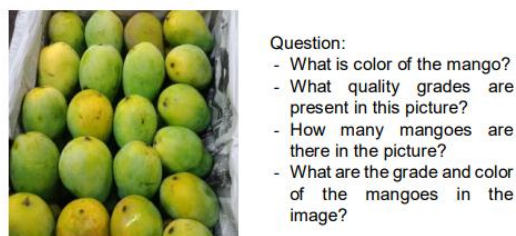


Fig. 2. VQA with multi-answer.

Based on these examples, the image may contain multiple answers in different regions. To accurately decipher such visual

complexity, VQA systems require not only the ability to perceive global image features, but also local-level interpretation through object-based representations. Object detection technology plays a crucial role as a visual foundation; the individual localization and classification of each object instance, along with its physical attributes, serve as a fundamental prerequisite before the system can formulate a comprehensive and reliable set of answers [18]. Without explicit object-level representations, a VQA system operates without explicit visual grounding. Object detection integration serves as a visual anchor that allows the system to granularly dissect each entity, help clarify ambiguous questions, and enable the creation of a complete set of answers that fully represent the complexity of visual content [19].

The multi-answer VQA problem was first formulated in 2021 by [17] through a glimpse-based CNN-LSTM framework (EMQA), which crops images into object-specific glimpses and aggregates predictions via a sliding-window mechanism. However, the approach remains limited to local object-level reasoning, lacks global scene understanding such as counting and spatial relationships, and reports modest performance on its own ICon Question Answering (ICQA) benchmark (F1-score of 62.91%), partly due to the general absence of standard multi-answer VQA datasets.

A systematic literature review of 58 recent studies on VQA, focusing on multi-answer concepts and object-level visual representations, reveals 2 critical gaps behind this single-answer bias. First, only 17% ($n=10$) of the studies explicitly design VQA models to generate multiple answers as a primary objective, an exception rather than the norm, with most work instead relying on multi-instance or attention mechanisms that ultimately collapse to one answer. This reflects a "technical lag" phenomenon: advanced visual perception mechanisms are increasingly capable of representing complex visual scenes, while answer-generation mechanisms remain largely designed around single-label prediction (e.g., softmax-based classification) rather than multi-label prediction (e.g., sigmoid-based outputs) [18]. Second, domain-specific research accounted for only 28 of the 58 studies, and agriculture and horticulture were represented by just two, a mere 7.1% of the domain-specific category, or 3.4% of the total. This scarcity is further amplified by the difficulty of constructing agricultural VQA datasets, which require expert annotation aligned with domain-specific quality standards [19].

Recent VQA approaches commonly rely on learned multimodal fusion mechanisms that achieve strong performance across general visual reasoning tasks. Although these approaches achieve impressive results, purely neural methods may face limitations in complex reasoning tasks, where the relationship between visual evidence and generated answers is difficult to interpret, and they introduce further challenges related to computational cost and explicit grounding of answers to visual evidence [19]. Neuro-symbolic AI has emerged as an alternative paradigm that combines neural perception with explicit symbolic representations and reasoning to improve interpretability and facilitate traceable

decision processes [20]–[22]. Unlike fusion-based approaches, Neural-Symbolic VQA (NS-VQA) explicitly separates vision and language understanding from reasoning, enabling answers to be derived from interpretable visual evidence rather than latent feature representations alone [23]. Inspired by this principle, MangoVQA adopts a modular perception–reasoning architecture in which neural object detection provides structured visual evidence, while deterministic text processing interprets user queries and generates answers from object-level representations. Unlike learned neuro-symbolic architecture designed for general reasoning benchmarks, MangoVQA is specifically optimized for quality inspection scenarios where answers must be traceable to detected objects and aligned with formal grading standards.

This study addresses these gaps by introducing MangoVQA, a multi-answer Visual Question Answering framework for SNI 3164:2024-aligned mango quality grading. MangoVQA combines a YOLOv11n-based deep learning visual perception module with deterministic text processing for question interpretation and answer generation. The detector identifies individual mango instances, their quality grades, and spatial locations, while the Perspective Engine extracts additional attributes, including color, defect-related color, position, and object size. These object-level visual facts are organized into a structured answer representation, allowing responses to be generated according to user queries and grounded in detected mango instances. Unlike single-answer VQA systems, MangoVQA supports answer-set responses when multiple grades, attributes, or conditions are present simultaneously. The system covers both global reasoning over the entire image and local reasoning for individual mango instances, with grading criteria, annotation rules, and answer vocabulary aligned with SNI 3164:2024.

The main contributions of this study are as follows: (1) developing a multi-answer VQA framework that integrates YOLOv11n-based visual perception with deterministic text processing for object-grounded answer generation in mango quality grading, aligned with SNI 3164:2024; (2) constructing a bilingual multi-answer VQA dataset containing 17,764 question-answer pairs from 468 annotated mango images, covering global and local reasoning scenarios; and (3) introducing a lightweight and traceable VQA approach suitable for agricultural quality assessment applications requiring interpretable answers and standard-aligned decision support.

II. RELATED WORK

A. Mango Quality Grading and SNI 3164:2024

Mango (*Mangifera indica* L.) represents an important agricultural commodity in Indonesia, where consistent quality assessment is essential for supporting domestic and export markets [24]. SNI 3164:2024 Mango is the current Indonesian National Standard for Mango [5]. This standard was developed through an independent development process and officially

established by the National Standardization Agency of Indonesia (BSN) in 2024 and revised version of SNI 3164:2009 [25]. This standard establishes the quality, packaging, and labeling requirements for mango fruit (*Mangifera indica* L.). This standard applies to commercial mango varieties of the family Anacardiaceae marketed for fresh consumption after handling and packaging. Mangoes intended for industrial or processed uses are not covered by this standard [5].

SNI 3164:2024 classifies mangoes into three quality categories based on surface defects, uniformity of shape, and physical condition: Super Class, Class I, and Class II [5], as shown in Fig. 3. This aligns with the regional (ASEAN) [26], and international (FAO/WHO Codex) [27] standards for mangoes, which also categorize quality into three grades: Extra Class, Class I, and Class II. In this study, author use the Extra Class label for Super Class to ensure widespread acceptance.



Fig. 3. Three different types of mangoes with super class (extra class), class I and class II quality [5].

Extra Class represents mangoes with the highest quality characteristics and no significant defects. Class I includes commercially acceptable mangoes with minor skin surface defects within a maximum tolerance of 5%, while Class II includes mangoes that do not qualify for higher grades but still satisfy general requirements with minor skin defects within a maximum tolerance of 10%.

Minor skin surface defects that are still tolerated in this case are defects due to scratches or sunburn, sap stains, and abrasions that do not exceed the specified limits and do not affect the flesh of the fruit. The four-class configuration extends this standard by introducing a Reject class to represent mangoes with severe damage, disease symptoms, deformities, or defects affecting the flesh, which are practically unsuitable for commercial distribution.

B. General-domain VQA

Emerging around 2014, VQA has become an interesting research area, because it enables human interaction with images, such as initially providing visually impaired individuals with the ability to perceive those images. This rapidly evolving field has witnessed substantial advancement through the integration of deep learning technologies, the growth of large-scale multimodal datasets, and the development of sophisticated vision-language approaches across diverse application domains, including healthcare, robotics, remote sensing, and artificial intelligence systems [28], [29].

The general workflow of Visual Question Answering (VQA), as described by [30] illustrates a single-answer VQA

architecture, in which a model learns from images paired with questions and their answers. The main stages are as follows:

1) *Image encoding.*

The image encoder extracts visual features that can be used in later stages. These may include color, shape, size, and other relevant characteristics of an image or video.

2) *Question encoding.*

The question encoder converts a natural-language question into a numerical representation. A long short-term memory (LSTM) network is one method that can be used at this stage.

3) *Joint feature representation.*

The semantic encoder combines the image, and question features so that the model can interpret the question in relation to the visual content.

4) *Answer predictions.*

The answer encoder uses the combined representation to predict an answer to the question about the input image.

VQA was popularized by large open-domain benchmarks, namely VQA [13], its successor VQA v2.0 [14], and by earlier real-image datasets such as DAQUAR [31]. However, previous VQA methods tend to memorize biases present in the training data, rather than learning proper behavior, such as grounding on images before predicting answers. Multi-answer VQA, in which a single question may map to a set of correct answers, was explored using a glimpse-based CNN+LSTM [17] that crops the image into local glimpses and recombines them; because training is per-glimpse, it cannot answer questions tied to global image context such as counting or location. MangoVQA instead uses whole-image object detection to construct a structured set of per-object records, allowing the same grounded representation to support both scene-level aggregation and object-specific question answering.

C. Recent Directions in VQA

Work since 2022 has pushed general VQA toward steadily heavier learned fusion. Attention has been refined with a similarity-modulated soft-attention gate [32], relational reasoning has been recast as graph attention over generated scene graphs [33], and region selection has been made multi-granular to balance information sufficiency against redundancy [34]. A parallel strand injects external knowledge, either by translating image–caption–question embeddings via a large language model [35] or by aligning semantically co-occurring attention with a knowledge base [36]. The task has also spread into specialized domains: fine-grained facial expression analysis [37], medical imaging [38], and disaster interpretation in remote sensing, where autonomous agents orchestrate several perception tools [39].

Several of these do acknowledge answer multiplicity, each in a different way. SCAG absorbs multiple ground-truth answers through a sigmoid restriction loss, yet inference still

returns the highest-scoring answer and is scored with single-answer VQA accuracy [36]. HiCA-VQA emits genuine multi-label predictions, but over a closed, hierarchically predefined clinical vocabulary [38]. RescueADI returns structured (plan, perception, recognition) tuples for disaster interpretation in remote sensing, where autonomous agents orchestrate several perception tools [39].

None of these systems tie the individual elements of an answer to the individual objects in the image. Their multiplicity comes from a loss function, from a predefined attribute list, or from a record of task decomposition, never from the objects themselves. MangoVQA takes the opposite position: each element of an answer set is produced by one detected fruit and can be traced back to it. This traceability carries weight in regulatory contexts such as SNI-based grading, where a black-box prediction cannot be independently audited, whereas MangoVQA’s per-object provenance lets every graded answer be verified against the specific detection that produced it.

Table 1 shows MangoVQA against the systems discussed above, verified from their original papers rather than abstracts, on the dimensions the field most commonly reports: whether the answer cardinality adapts to scene content, whether each answer element is grounded to a specific detected object, whether the reasoning path is inspectable without post-hoc tools, whether the system is aligned to a formal quality standard, its computational cost, and its dataset scale. None of the five prior systems ground every answer element to a detected object, none exposes an inspectable reasoning path, and none is aligned to a formal standard; MangoVQA is the only system in comparison that satisfies all three at once, while remaining computationally light.

D. VQA in Agriculture and Horticulture

Compared with general and medical VQA [40], [41], agricultural VQA is sparse, and horticulture-specific, standard-aligned VQA is rarer. As noted in Section I, this scarcity of multi-answer, agriculture-focused VQA work is consistent with our systematic review. The closest work to ours is HortiVQA-PP, a horticultural multitask framework that segments pests and their natural predators, models their ecological co-occurrence, and answers questions using a horticulture knowledge graph, reporting 83.5% multi-label accuracy on the co-occurrence task [42]. It shares our horticultural setting and interest in set-valued output, but it targets pest management rather than fruit quality, its answers are generated by a multimodal network rather than grounded in an auditable rule set, and it is not aligned to any grading standard. To our knowledge no public VQA benchmark targets mango quality grading against a formal standard. MangoVQA addresses this scarcity with an SNI-aligned [5], multi-answer, bilingual QA resource generated directly from detection labels.

Table 1.
Comparison with Prior Multi-Answer and Knowledge-Grounded VQA Systems

System	Domain	Cardinality	Grounding	Transparency	Std.-Aligned	Compute	Size
EMQA [17]	Icons/general	Adaptive	Partial	No	No	Light	~42,000 QA
SCAG [36]	Gen./med./RS	Fixed	No	No	No	Heavy	~204,000 QA

System	Domain	Cardinality	Grounding	Transparency	Std.-Aligned	Compute	Size
HiCA-VQA [38]	Medical	Adaptive	No	No	No	Moderate	3,720 QA
RescueADI [39]	Remote sensing	Fixed	Partial	Partial	No	Heavy	4,044 QA
HortiVQA-PP [42]	Horticulture	Adaptive	Partial	No	No	Moderate	14,460 QA
MangoVQA (current study)	Fruit grading	Adaptive	Yes	Yes	Yes—SNI 3164	Light	468 img / 17,764 QA

E. Rule-based versus learned question encoding.

Mainstream VQA encodes questions with learned sequence models such as LSTM [17], term frequency–inverse document frequency (TF–IDF) weighting, or a Transformer text encoder, and grounds that vector on region features produced by a detector [43]. MangoVQA does not encode the question; it processes it through a rule-based text processing pipeline. Text processing here refers to the general task of extracting and examining unstructured text by identifying its important words and the relationships among them. In MangoVQA, this task is carried out deterministically by mapping the words that appear in a question to their corresponding grade, color, side, and object slots. For an auditable grading system with a controlled vocabulary, we deliberately adopt a deterministic four-stage parser (normalization → tokenization → featurization → intent classification) that performs intent classification with slot filling, mapping free text to one of 22 question intents together with its slots (grade, color, side, object). The question therefore leaves the language stage as an explicit, structured representation rather than a distributed vector, consistent with the rule-based design of the reasoning side. This module occupies the position that a learned semantic parser holds in neural-symbolic VQA systems, but is implemented here as a fixed, deterministic parser rather than a learned one [23], and it trades open-domain linguistic coverage for full transparency and exact reproducibility while keeping system answers consistent with the gold generator, because both are produced by the same template rules. Unlike NS-VQA and NS-CL, whose question-interpretation stage is a learned semantic parser, MangoVQA’s question-interpretation stage is a static, rule-based syntax matcher—a fixed four-stage pipeline mapping free text to one of 22 predefined intents and their slots. This design severely limits open-domain linguistic flexibility, since only phrasings covered by the template rules are guaranteed to be parsed correctly. The parser’s-controlled vocabulary is designed specifically for SNI-graded terminology (grade, color, side, object); it does not support general open-domain queries outside this vocabulary. Replacing this parser with a learned one is left as future work.

III. RESEARCH METHOD

MangoVQA is organized as an explicit pipeline of modules as seen in Fig. 5: 1) InputAcquirer (image); 2) YOLOPreprocessor (letterbox resize and padding preserving aspect ratio); 3) MangoDetectorYOLO (YOLOv11n detection); 4) FruitPerspectiveEngine (per-object visual-attribute analysis); 5) QuestionAnsweringSystem (rule-based answering). The

engine executes four traceable stages: detection, answer-bank construction over the detections, question routing (template path or open free-text path), and presentation with a reasoning trace. Both paths resolve to the same answer bank, so system answers stay consistent with the gold data. The first three modules constitute the deep learning-based perception stage and the last two the rule-based reasoning stage, so the boundary between perceiving and reasoning is a boundary between modules rather than between layers of one network.

A. Formal Definition of the Rule-Based Parser (P1–P4)

The question-processing pipeline maps a free-text question q to an intent and its slots through four deterministic stages, P1 through P4, namely normalization, tokenization, featurization, and intent classification with slot filling, defined formally in (1) through (4).

In the first stage, normalization converts the raw question q into a canonical form q' by lowercasing, stripping punctuation, collapsing whitespace, and mapping known synonyms through a dictionary D_{syn} , as in (1).

$$q' = \text{normalize}(q) \quad (1)$$

The canonical question q' is then tokenized into a sequence of tokens $T = [t_1, \dots, t_n]$, as in (2).

$$T = \text{tokenize}(q') \quad (2)$$

Each token $t_i \in T$ is next assigned to a feature $f(t_i)$ drawn from the set $\{\text{grade, color, side, object, count, null}\}$, determined by matching t_i against the controlled vocabularies $V_{\text{grade}}, V_{\text{color}}, V_{\text{side}},$ and V_{object} , as in (3).

$$f(t_i) \in \{\text{grade, color, side, object, count, null}\} \quad (3)$$

Finally, the extracted features of T are matched against a rule set R to determine the question’s intent, with slots filled directly from the matched tokens; when no rule in R matches, the intent defaults to UNRECOGNIZED, as in (4).

$$\text{intent} = \text{match}(\text{features}(T), R) \quad (4)$$

Together, (1) through (4) converts an unstructured question into the structured intent-slot representation consumed by the answer bank.

B. Data Collection and Annotation

Authors used the four-class mango detection dataset from

our previous assessment work, annotated against SNI 3164:2024. Three classes (Extra Class, Class I, Class II) are the official quality levels. The fourth class, the Reject Class was added to represent unmarketable fruit. Table 2 summarizes the number of images collected.

Then, split the image dataset into 70% for training, and 30% for validation and testing. All reported evaluation scores are from the test split, and the QA dataset and responses are from the same images and labels, ensuring no image leakage between splits.

Table 2.

Image and Mango-Instance Split of The Four-Class Dataset	
Parameter	Images
Total Images	468
Total Annotations	745
Train (Image/ Annotations)	327 / 555
Test (Image/ Annotations)	141 / 190
Extra Class	187 (25.1%)
Class I	349 (46.8%)
Class II	159 (21.3%)
Reject	50 (6.7%)

The images come from two sources. The first is the publicly available mango varieties classification and grading dataset [44], captured under controlled studio conditions and containing single-mango images of the three standardized grades; these images retain the expert grading conventions of their original source, which the authors verified align closely with SNI 3164:2024 and other international standards, and were mapped to the four SNI classes. The second is a set of real-world images gathered from publicly available online sources, which introduces environmental variability and contributes both the multi-instance images and the reject cases that practical post-harvest sorting produces; these images were annotated by a single rater, after review of international, ASEAN, and SNI grading standards. Every image was annotated in YOLO bounding-box format, and each box was assigned to a grade using the percentage-based surface-defect thresholds of SNI 3164:2024 [5].

Annotation followed the structured decision flow in Fig. 4, applied so that labelling stops at the first matching condition. Fruit free of defects, where a very minor blemish is still acceptable, is labelled Extra Class; otherwise, a skin defect of at most 5% with unaffected flesh gives Class I, and a skin defect of at most 10% with unaffected flesh gives Class II; fruit that fails all three criteria is labelled Reject. Because the two sources were graded independently of each other rather than by multiple raters assessing the same images, standard inter-annotator agreement metrics such as Cohen's Kappa, which require overlapping items rated by more than one annotator, do not apply to this dataset; particular care was taken at the Class II–Reject boundary, where the decision turns on the severity and extent of the damage rather than on a clean numerical threshold.

C. System Architecture

Fig. 5 below shows the MangoVQA system architecture, organized as a five-stage pipeline in which each stage produces

the structured input the next stage requires.

- **InputAcquirer** accepts the raw visual input, a static image, and passes it downstream in a common format, so that the rest of the pipeline is agnostic to how the image was originally captured.
- **YOLOPreprocessor** prepares that raw image for the detector. It applies letterbox resizing, padding the image to preserve its original aspect ratio rather than stretching it, since distortion at this stage would shift object scale and degrade detection accuracy.

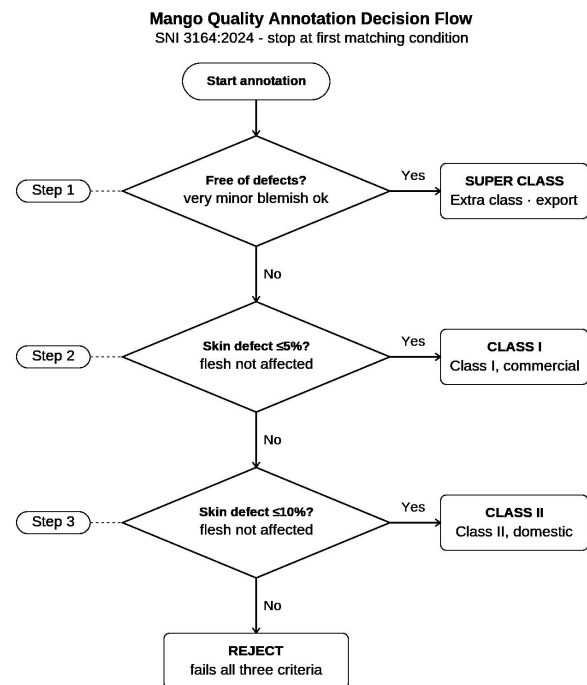


Fig. 4. Mango quality annotation decision flow adapted from SNI 3164:2024 [5].

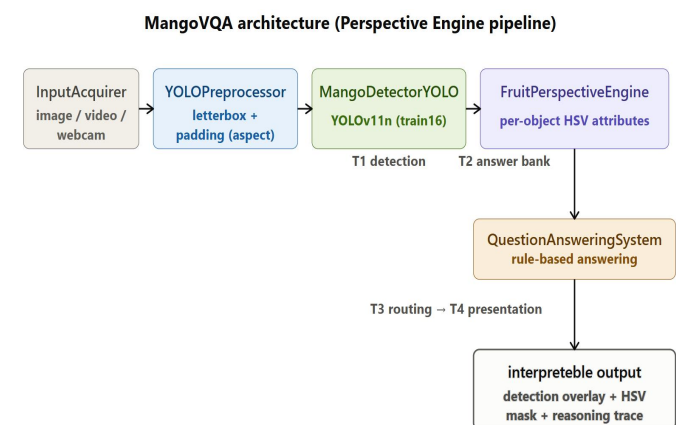


Fig. 5. MangoVQA system architecture.

- **MangoDetectorYOLO** is the YOLOv11n detector itself (model train16). It localizes every mango in the image and assigns each one a bounding box and a quality-grade label (Extra Class, Class I, Class II, or Reject) completing what we label T1: detection in Fig. 5. At this point, the pipeline has

produced a set of grounded, per-object detections, but nothing yet about color.

- **FruitPerspectiveEngine** takes each detected bounding box individually and derives additional visual attributes not reported by the detector itself, adding multi-answer nuance to MangoVQA: the color of the mango fruit, and the color of defects for fruit marked as rejected. Both are calculated through a calibrated hue–saturation–value (HSV) analysis of the pixels within each box. This step, labeled T2: answer bank in Fig. 5, transforms a simple detection (box and color level) into a more complete per-object record, augmented with color, position, and size levels, that can support a much wider range of questions than detection alone.
- **QuestionAnsweringSystem** is a rule-based reasoning layer. The system accepts a natural language question, parses it through four stages, identifies its intent, and then answers it by querying the per-object record generated by the Perspective Engine, either at a global level (reasoning over the entire image, e.g., "what levels are present?") or a local level (reasoning about a specific fruit, e.g., "what color is this fruit?"). This stage includes T3: routing (matching the deciphered intent with the correct answer generation rules) and T4: presentation (formatting the generated answers, including set-valued answers if more than one answer is valid).

Finally, the system produces interpretable output, that each answer is returned along with an annotated detection overlay (an image with bounding boxes and levels drawn on it) and a step-by-step reasoning trace showing which rules were activated and which detected objects were drawn, so that each answer can be audited back to the specific visual evidence and logical rules that generated it, rather than being returned as a vague prediction.

Figure 5 shows how the perception and reasoning stages are connected in the MangoVQA pipeline, allowing the source of each answer to be traced at two levels: the visual evidence produced by the YOLOv11n detector, and the explicit rule applied by the rule-based reasoning component. To make this correspondence concrete rather than merely conceptual, Table 3 maps each design concept introduced in Section II to its implementation in the system.

Table 3.

Design Concept and Concrete Implementation in MangoVQA	
Design Concept	Concrete Implementation in MangoVQA
Deep learning perception component	MangoDetectorYOLO, the trained YOLOv11n detector, which localizes each fruit and assigns its quality grade
Structured object representation	FruitPerspectiveEngine, which builds a per-object record combining detected grade and bounding box with color, position, and size attributes
Rule-based text processing and reasoning component	QuestionAnsweringSystem, which parses the question through the four-stage rule-based parser (P1–P4) and applies deterministic rules to the answer bank

D. Object Detection Module

The detector is YOLOv11n trained for 100 epochs at image size 480 with batch size 4 and AMP disabled (model trained). Letterbox pre-processing preserves fruit geometry before the network input, improving bounding-box regression. At inference the settings are fixed to match the official configuration: image size 480 (identical to training), confidence threshold 0.30, non-maximum suppression (NMS) IoU 0.55, class-agnostic NMS enabled to suppress duplicate cross-class boxes on the same fruit, and a maximum of 300 detections per image. We pin the evaluation image size to match the training size, rather than letting it vary. This matters because feeding the detector a larger image shifts how object scale is represented relative to what it learned during training, which in practice causes two problems: leaves get mistaken for Reject-class fruit, and bounding boxes drift slightly away from the true object position. Keeping the two sizes consistent avoids both issues. Table 4 below shows the training is performed in a fixed number of epochs with pre-trained weights to speed up convergence and improve generalization.

Table 4.
Training Hyperparameters for YOLOv11n Fine-Tuning

Parameter	Value
Model	yolo11n.pt (YOLO11 Nano)
Task	Object Detection
Epochs	100
Batch size	4
Image size	480 x 480
Optimizer	Auto (lr0 >= 0.01: SGD, lr0 < 0.01: Adam/ AdamW)
Initial learning rate (lr0)	0.01
Patience (early stopping)	100
Pretrained weights	Yes
AMP (mixed precision)	Disabled
Workers	8
IoU threshold	0.7
Device	Auto (NVIDIA MX450)

E. Perspective Engine: Color Attribute Analysis

Fruit-pixel separation does not use a trained segmentation model but two threshold-based filtering stages: cropping the central bounding-box region (12% edge shrink) to remove the background at the box corners, then HSV masking that removes low-saturation background, deep shadows, and over-bright highlights. Hue-based color assessment for mango has an established basis in the color-feature literature [45], further reviewed and confirmed by the domain researcher. Table 5 gives the calibrated hue bins, cross-checked against this literature rather than validated through blind independent human color labeling, that were used to build the color-determination rules below.

Hue (H) is color tone, ranging 0–179 in OpenCV, and is the main determinant of color; Saturation (S) is color saturation or depth, 0–255 (low values indicate pale or grey); Value (V) is brightness, 0–255 (low values indicate dark or shadowed

regions). The four base fruit colors are determined from the median hue of the cropped region against the threshold bins in Table 5, calibrated from the hue distribution of the annotated four-class dataset ($n = 745$ fruit boxes), with an additional "Unknown" category assigned when the color cannot be determined. The defect color for the Reject class is determined by matching per-pixel HSV values against the same threshold bins in Table 5, triggered specifically when the object grade indicates a decay, bruise, or mold condition.

Table 5.
Calibrated HSV Color Scheme (OpenCV Hue Scale)

No	Color	Condition HSV Hue (0–179) & Degree (0–360)	Actual Color	Hexa Code
1	Green	H 41–89 (82°–178°)		#3AD153
2	Yellow-green	H 33–40 (66°–80°)		#B0D13A
3	Yellow	H 20–32 (40°–64°)		#D1BD3A
4	Orange-red	H 0–19, 160–179 (0°–38°, 320°–358°)		#D16A3A
5	Black/dark (defect)	$V < 70$		#231B1B
6	Grey (defect)	$S < 45$ AND $V \geq 70$		#968A8A
7	Brown (defect)	$H \leq 25$ AND $S \geq 45$ AND $70 \leq V < 170$		#785840
8	Unknown	None of the above hue ranges match	-	-

F. Question-Template Design

The question set comprises 22 types at two reasoning levels. Seventeen global types reason over the whole image (counts, existence, which-grades, dominant, best/worst quality, largest, overall marketability, per-side grades, the grade and color breakdowns, and a combined grade–color summary); five local types ask about a single detected object (grade, color, horizontal position, marketability, and a combined color–grade answer). After slot filling (grade, color, side), the 22 types instantiate 35 concrete question patterns for a single-object image per language (30 global patterns plus 5 local patterns per detected object) generated in both Indonesian and English from the same template rules.

Table 6 lists every question type with its reasoning level (global over the whole image versus local about one detected fruit), its answer cardinality (single open-ended answer versus a set-valued multi-answer), and its evaluation track. Questions are grouped into three evaluation tracks based on the evidentiary source of their gold answers: Track A questions are grade-based, with gold answers from human-verified ground truth; Track B questions are color-based, with gold answers derived from HSV pseudo-labels; and Track C questions combine both.

Fifteen types are grade-based (Track A, the primary claim: twelve global and three local), five are color-based (Track B, HSV pseudo-labels), and two combine both (Track C). Seven

types are set-valued by construction, which is where the multi-answer capability is exercised: a single image may legitimately contain several grades or several colors at once, and a single-answer VQA formulation cannot express that. In addition, object_color is set-valued for Reject fruit, where a defect color and a healthy base color are both reported. The rightmost column gives how many concrete questions each type instantiates per image after slot filling. For example, *count_class* and *exist_class* expand over the four grades, and *count_color* over the seven color bins (four color bins plus three defect bins), which is why the seventeen global types yield 30 global patterns per image.

G. Gold Answer Generation and QA Dataset Construction

Answers follow a uniform schema: the answers field is always a list of strings, and the answer type is either open-ended (single) or multi-label (set). Global answers aggregate over all boxes (totals, per-grade counts, presence, best/worst by the SNI quality ordering Extra Class > Class I > Class II > Reject, per-side grades, and breakdowns); local answers are emitted per detected object with its object identifier and bounding box. Gold answers are produced directly from human ground-truth detection labels using the exact template rules the system uses at inference, so any score difference reflects detection or attribute error rather than answer-wording differences. Running the generator over the 468 images yields 17,764 QA (Question Answer) pairs, of which about $\approx 22.85\%$ are multi-answer. Table 7 breaks this down so the count is reproducible.

Table 7.
Generated QA Dataset (Indonesian; English Equivalent)

Split	Images	Objects	Global QA Pair	Local QA pair	Total QA pairs
Train	327	555	9,810	2,775	12,585
Validation	70	105	2,100	525	2,625
Test	71	85	2,129	425	2,554
Total	468	745	14,039	3,725	17,764

Each image contributes 30 global questions: the four repeating types expand to *count_class* $\times 4$ and *exist_class* $\times 4$ over the grades, *count_color* $\times 7$ over the color bins, and *grades_on_side* $\times 2$ over the two sides, with the remaining thirteen global types contributing one each. Local questions are emitted per detected object rather than per image, 5 per object, so a split with more fruit per image yields proportionally more local pairs. These per-image counts are maximum: a type is skipped whenever its answer is undefined, which across the whole dataset happens exactly once, dominant on a single test image where no grade is most frequent and is why the test split yields 2,129 rather than 2,130 global questions.

Table 6.
The 22 Question Types by Reasoning Level, Answer Cardinality, and Evaluation Track

No	Question type	Level	Answer	Track	Per image	Question
1	count_total	global	single	A	1	How many mangoes are in the image?
2	count_class	global	single	A	4 (class)	How many mangoes are graded as {Class}?
3	exist_class	global	single	A	4 (class)	Is there a mango graded as {Class}?
4	count_grades	global	single	A	1	How many different quality grades are in the image?

No	Question type	Level	Answer	Track	Per image	Question
5	dominant	global	single	A	1	Which quality grade is the most frequent in the image?
6	best_quality	global	single	A	1	What is the highest quality grade present in the image?
7	worst_quality	global	single	A	1	What is the lowest quality grade present in the image?
8	largest_grade	global	single	A	1	What is the grade of the largest mango in the image?
9	overall	global	single	A	1	Are all mangoes in the image marketable?
10	which_grades	global	set	A	1	Which quality grades are present in the image?
11	grades_on_side	global	set	A	2 (left/ right)	Which quality grades are on the {left/right} side of the image?
12	grade_breakdown	global	set	A	1	Which quality grades are present and how many of each?
13	count_color	global	single	B	7 (color)	How many mangoes are {color} in color?
14	count_colors	global	single	B	1	How many different mango colors are in the image?
15	which_colors	global	set	B	1	Which colors of mango are there in the picture?
16	color_breakdown	global	set	B	1	Which colors are present and how many of each?
17	grade_color_global	global	set	C	1	What are the quality and color of the mangoes in this image?
18	object_grade	local	single	A	1	What is the quality grade of this mango?
19	object Marketable	local	single	A	1	What is the colour of this mango?
20	object_position	local	single	A	1	Where is this mango located horizontally?
21	object_color	local	single	B	1	Is this mango marketable?
22	object_color_grade	local	set	C	1	What is the colour and quality of this mango?
Total		global			30	
Total		local			5	

Because these types are generated for every image, the multi-answer share is nearly identical across splits (train 22.80%, validation 23.01%, test 22.94%). The number of elements per answer set, however, depends on how many mangoes and grades an image contains. Decoupled Evaluation and Metrics

Evaluation compares system answers, obtained by running the detector on the test images and then the same templates, against gold answers, and is split into three tracks to keep the claims honest. Track A (grade, primary) covers every grade-based question type, with gold answers from human ground-truth labels. Track B (color, secondary) covers the color-related types, whose gold answers are HSV pseudo-labels; because Track B would otherwise score the HSV heuristic against itself, it is reported separately and never merged into the main grading metric. Track C (composite) covers question types that combine grade and color in a single answer set; because it mixes human-verified grade labels with HSV pseudo-labels, it is likewise reported apart from the primary metric. Global answers are matched by question type and question text; local answers match each gold object to a predicted box by $\text{IoU} \geq 0.5$, and an unmatched gold object counts as a missed detection, so detection recall is charged to the VQA score rather than hidden. Gold answers are never produced by the model: they are generated from the human ground-truth labels through the same template rules the system uses at inference, so any gap between gold and system answers isolates detection and attribute error from answer-wording differences.

Metrics follow the answer type. Open-ended (single) answers use per-type accuracy (exact match). Multi-label (set) answers use set-based precision, recall, and F1-score, plus subset exact match (subset-EM), which counts an answer correct only when the predicted set equals the gold set exactly, with no missing and no spurious element. Precision therefore

penalizes over-answering and recall penalizes under-answering, so F1 measures completeness of the answer set rather than merely hitting one acceptable answer. Scores are pooled micro-wise within a question type and averaged macro-wise across types, so that rare question types are not drowned out by frequent ones. Let A_{sys} and A_{gold} denote the system and gold answer-sets for a question, precision, recall, and F1-score are defined in (5) through (7):

$$P = |A_{\text{sys}} \cap A_{\text{gold}}| / |A_{\text{sys}}| \quad (5)$$

$$R = |A_{\text{sys}} \cap A_{\text{gold}}| / |A_{\text{gold}}| \quad (6)$$

$$F1 = 2 \cdot P \cdot R / (P + R) \quad (7)$$

Letterbox pre-processing is $O(N)$ in the number of pixels N ; YOLOv11n inference is approximately constant per image at fixed image size; and the rule-based VQA stage is $O(M \times K)$, where M is the number of detected objects and K is the crop size analyzed by the Perspective Engine. Because the answering layer is rule-based, T_{VQA} is dominated by per-object HSV analysis and remains small; overall cost is governed by detection.

IV. RESULT

A. Detection Performance Recap

The detector attains $\text{mAP}@0.5 = 0.9558$, $\text{mAP}@0.5:0.95 = 0.8838$, $\text{precision} = 0.8889$, and $\text{recall} = 0.8744$, on the four-class task at the single-split checkpoint (epoch 88), as detailed in Table 8. Per-class breakdown (Table 9) shows the Reject precision of 1.0000 indicates that every Reject prediction at this checkpoint was correct, with no false positives; but here, the Reject class has the weakest recall of the four (0.8102) meaning some Reject fruit may have been missed.

To overcome this condition, the deployment interface incorporates dynamic parameters with default settings of Confidence ($\text{conf} = 0.30$) and Intersection over Union ($\text{IoU} =$

0.55). This configuration directly addresses the limited representation of the Reject class. By lowering the confidence threshold to 0.30, the model can reliably capture faint, subtle, or ambiguous defect features on the mango surface without sacrificing the overall precision of the well-represented commercial classes. This dynamic thresholding allows operators to fine-tune the grading sensitivity on-the-fly, ensuring stable object localization and precise class scoring without requiring any modifications to the core codebase.

Table 8.

Detection Performance: Single-Split (70:30) vs. 5-Fold Cross-Validation

Metric	Single Split (70:30)	5-Fold Cross Validation (Mean)
mAP@0.5	0.9558 (ep.88)	0.8896 $\pm$ 0.0581
mAP@0.5:0.95	0.8838 (ep.88)	0.8248 $\pm$ 0.0670
Precision	0.8889 (ep.88)	0.8623 $\pm$ 0.0588
Recall	0.8744 (ep.88)	0.8074 $\pm$ 0.0797

Table 9.

Per-class Performance at Best Evaluation Checkpoint

Metric	Extra Class	Class I	Class II	Reject	Mean
4-Class (ep. 88)					
mAP@0.5	0.9624	0.9517	0.9333	0.9757	0.9558
mAP@0.5:0.95	0.9538	0.8672	0.9228	0.7916	0.8838
Precision	0.8444	0.9024	0.8088	1.0000	0.8889
Recall	0.9500	0.8721	0.8651	0.8102	0.8744

Figure 6 shows the normalized confusion matrix for the 4-class model. The 4-class model maintains the same strong commercial grade diagonal (Class I = 0.85, Class II = 0.89, Extra Class = 0.90), while the reject class reaches 0.75 despite adding only a small amount of training data for the reject class.

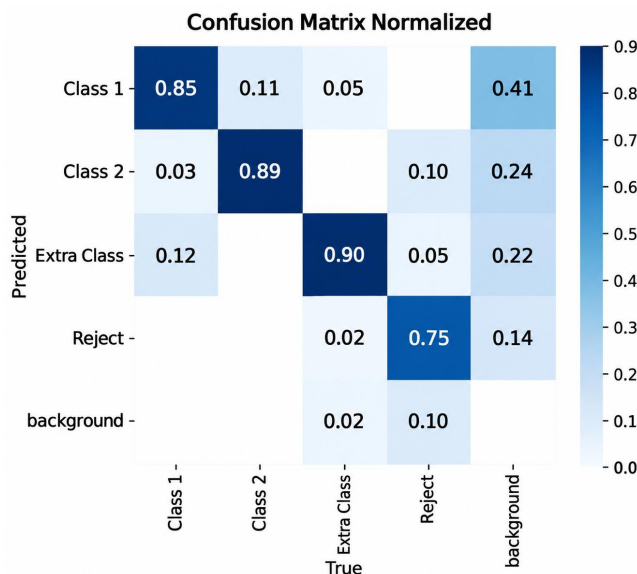


Fig. 6. Normalized confusion matrix of the four-class YOLOv11n detector.

To illustrate the model's behavior beyond the controlled test set, the deployed four-class YOLOv11n model was applied to unconstrained, real-world mango images spanning studio, on-tree, and packing-crate conditions. Figure 7 presents four

representative cases.

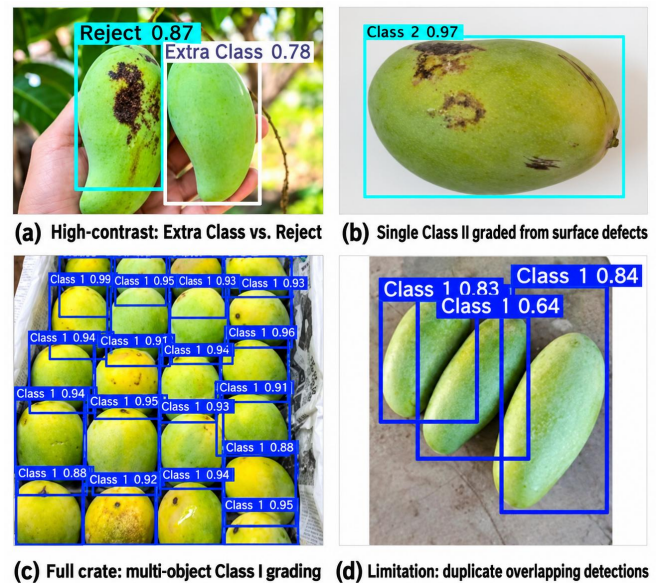


Fig. 7. Qualitative inference of the four-class YOLOv11n model on unconstrained real-world images: (a) high-contrast Extra Class vs. Reject; (b) a single Class II fruit graded from its localized surface defects; (c) multi-object Class I grading across a full packing crate; and (d) a limitation case showing duplicate, overlapping detections on adjacent fruit. Bounding-box labels are shown exactly as produced by the deployed model (which prints "Class 1"/"Class 2" for Class I/Class II).

B. Dataset Coverage and Composition

Before turning to how well MangoVQA answers questions, it is worth pausing on what it is asked to answer, since the dataset itself is one of this study's contributions. Applying the template generator (Section III.E) to the 468 annotated images produces 17,764 gold QA pairs, spanning 22 question types, 17 that reason over the whole tray and 5 that ask about one fruit at a time, split 14,039 to 3,725 between the two views. Every one of these pairs exists in two languages at once.

Table 10 confirms that Indonesian and English versions match exactly in count, order, and answer-set size across every split, a direct consequence of both editions being rendered from the same underlying symbolic answer through language-specific lexicons at the final step, rather than generated independently. In practice, this means a result reported in Indonesian is not an approximation of the English one: the two are interchangeable, and every score in this paper could equally well have been computed on either edition.

Table 10.
Bilingual parity audit of the MangoVQA dataset.

Split	Released file	Pairs (ID)	Pairs (EN)
Train	vqa_train_{id,en}.json	12,585	12,585
Validation	vqa_val_{id,en}.json	2,625	2,625
Test	vqa_test_{id,en}.json	2,554	2,554
Total	6 files, 16.5 MB	17,764	17,764

C. System Walkthrough: From Query to Answer

To illustrate how the components described in Section III operate together at inference time, Figure 8 traces a

representative query end to end. An image is uploaded together with a question, either selected from a template menu or typed as free text, for instance, "how many class 2 mangoes?" On the image side.

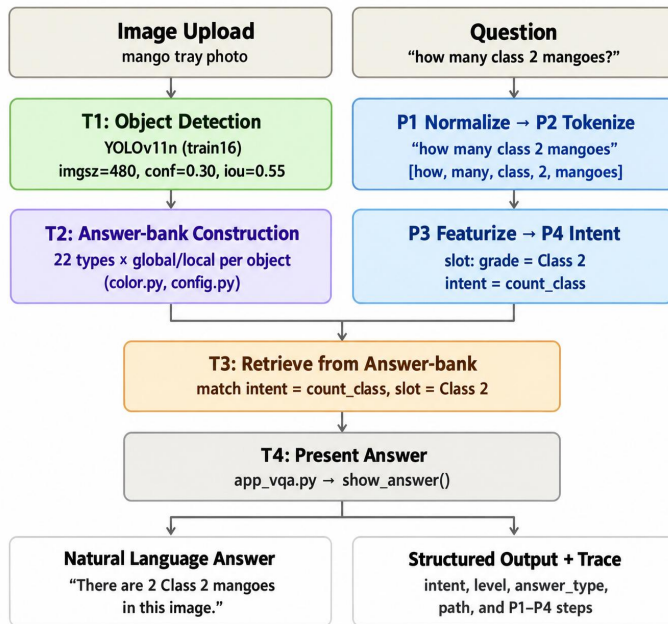


Fig. 8. MangoVQA query pipeline, from image and question to structured answer.

MangoDetectorYOLO first localizes and grades every fruit in the frame (T1), after which FruitPerspectiveEngine derives each fruit's color and assembles the results into a per-object answer bank spanning all 22 question types, at both the global and local level (T2). This construction happens once per image, regardless of how many questions are later asked of it. On the question side, if the query was typed rather than selected from the template menu, the rule-based parser reduces it through four stages (normalization, tokenization, featurization, and intent classification (P1–P4)) arriving at the intent count_class with the slot grade = Class 2. T3 then retrieves the matching entry from the answer bank by intent and slot, rather than recomputing anything, and T4 presents the result: a natural-language answer alongside a structured output containing the intent, reasoning level, answer type, the path taken, and the full P1–P4 trace.

Figure 9 shows this pipeline running in the deployed Streamlit interface. After an image is uploaded, the detection overlay is rendered alongside the original photo, together with a summary of the grades found. Selecting a template question, "how many mangoes are in the image?", returns the answer immediately from the same pre-built answer bank, with no separate inference pass. A second query on the same image is answered just as inexpensively: no new answer-bank construction, only a new look up against results that already

exist.

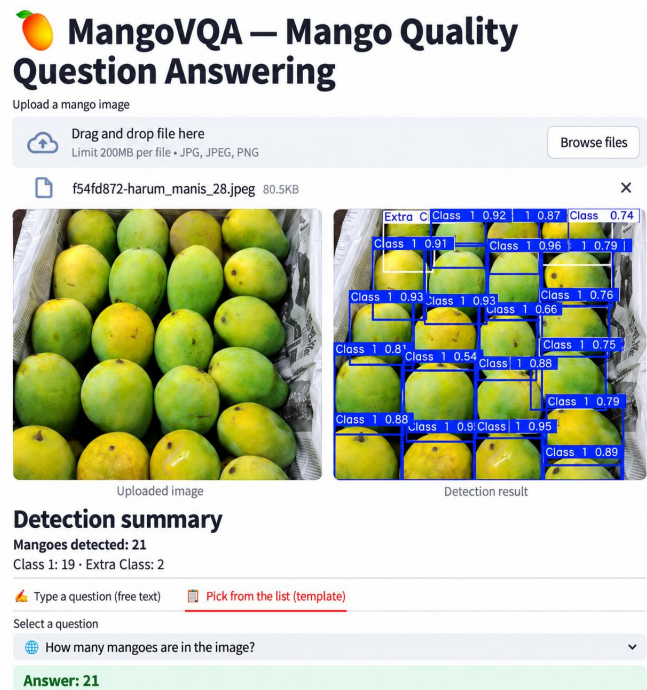


Fig. 9. MangoVQA Streamlit interface: image upload, detection overlay, and template-question answering.

D. Overall VQA Performance

Table 11 aggregates the test-split results by track and level. On the primary Track A, global open-ended questions reach 0.919 accuracy, local open-ended questions 0.937, and global multi-answer questions 0.846 micro-F1. Track B numbers are high but semi-circular (HSV gold vs. HSV system) and are reported separately; Track C, which mixes grade with HSV color, reaches 0.829 F1 globally and 0.919 F1 locally and is likewise kept apart from the primary claim.

The global multi-answer subset-EM (0.838) is slightly lower than micro-F1 (0.846) because subset-EM only credits an exact match between the predicted and gold answer sets, penalizing partially correct answers (one missing or one spurious grade) that F1 still rewards proportionally. This indicates the system usually identifies most correct grades present, but occasionally over- or under-reports by one distinction that matters more for applications requiring an exact answer set than for applications tolerant of partial answers. Because the pipeline is deterministic once the detector is trained, re-running it with different seeds would reproduce identical scores; the relevant variability is sampling variability over test images.

Table 11.
Aggregate VQA Results on The Test Split (conf 0.30, imgsz 480, NMS IoU 0.55, matching IoU 0.5)

Track / level	<i>n</i>	Metric	95% CI
A – grade, global, multi	284	F1 = 0.846 (P 0.850,	0.776–0.907

Track / level	<i>n</i>	Metric	95% CI
		R 0.841, subsetEM (0.838)	
A – grade, global, open	106	Acc = 0.919	0.872–0.957
A – grade, local, open	255	Acc = 0.937	0.897–0.969
B – color, global, multi	142	F1 = 0.966 (P 0.960, R 0.973, subsetEM 0.965)	0.926–1.000
B – color, global, open	568	Acc = 0.989	0.975–1.000
B – color, local, multi	4	F1 = 1.000 (P 1.000, R 1.000, subsetEM 1.000)	n/a (n = 4)
B – color, local, open	81	Acc = 0.975	0.937–1.000
C – composite, global, multi	71	F1 = 0.829 (P 0.840, R 0.818, subsetEM 0.817)	0.744–0.903
C – composite, local, multi	85	F1 = 0.919 (P 0.924, R 0.914, subsetEM 0.835)	0.880–0.952

The 95% confidence intervals in Tables 11 and 12 therefore come from a cluster bootstrap that resamples the 71 test images with replacement (2,000 resamples), keeping all questions of an image together because they share one set of detections. On Track A, global open-ended accuracy is 0.919 (95% CI 0.872–0.957; SD 0.023), local open-ended accuracy 0.937 (0.897–0.969; SD 0.019), and global multi-answer micro-F1 0.846 (0.776–0.907; SD 0.033). Intervals are widest for the once-per-image grade-ranking questions (e.g., *worst_quality* 0.845, 0.761–0.916), whose outcome hinges on the grade of the detected fruit, and narrowest for counting questions (*count_total* 0.986, 0.958–1.000), which depend only on the number of boxes.

E. Per-Question-Type Performance

Table 12 reports Track A performance by question type, sorted from best to worst. Fig. 10 visualizes this same per-type breakdown.

Table 12.

Track A Per-Type Results (All Types, Sorted Best to Worst)			
Type	<i>n</i>	Metric	95% CI
overall	71	Acc = 1.000	1.000–1.000
<i>count_total</i>	71	Acc = 0.986	0.958–1.000
<i>count_grades</i>	71	Acc = 0.986	0.958–1.000
<i>object Marketable</i>	85	Acc = 0.988	0.961–1.000
<i>object position</i>	85	Acc = 0.977	0.921–1.000
<i>exist class</i>	284	Acc = 0.926	0.884–0.961
<i>count class</i>	284	Acc = 0.923	0.880–0.958
<i>best quality</i>	71	Acc = 0.859	0.775–0.930
<i>largest grade</i>	71	Acc = 0.859	0.775–0.930
<i>dominant</i>	70	Acc = 0.857	0.771–0.930
<i>which grades</i>	71	F1 = 0.855	0.772–0.925
<i>object grade</i>	85	Acc = 0.847	0.768–0.915
<i>worst quality</i>	71	Acc = 0.845	0.761–0.915
<i>grades on side</i>	142	F1 = 0.843	0.770–0.907
<i>grade breakdown</i>	71	F1 = 0.841	0.750–0.918

Robust types include overall marketability (1.000), *count_total* (0.986), *object Marketable* (0.988), *count_grades* (0.986), *object position* (0.977), *exist class* (0.926), and *count class* (0.923). The hardest types are those requiring fine quality ranking, namely *worst_quality* (0.845), *dominant* (0.857), *best_quality* and *largest_grade* (0.859), together with *object grade* (0.847), all sensitive to confusions between

adjacent grades (chiefly Class I and Class II). Multi-answer types land in the 0.84–0.86 F1 range.

Authors keep the *largest_grade* question so that superlative reasoning stays complete; on this dataset, however, almost every image (94.4%) holds a single mango, so its answer nearly always coincides with *best_quality*, and its accuracy should be read as a lower bound on genuine size reasoning.

Track A — Per-question-type VQA performance (test split)

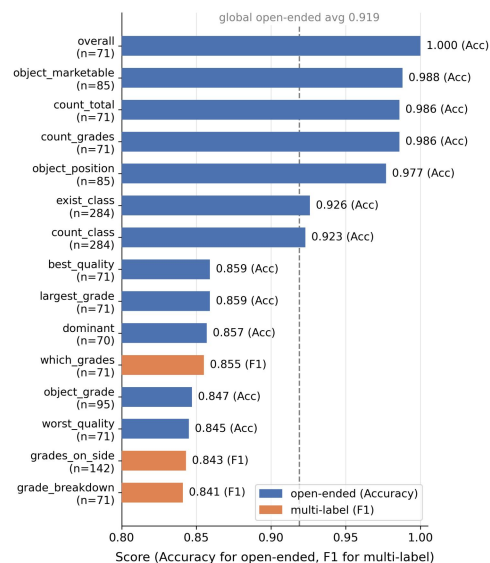


Fig. 10. Track A per-question-type performance on the test split (accuracy for open-ended, F1 for multi-label).

F. Comparison with a Learned Baseline

To test whether a learned answering layer could replace the templates, we trained a classical neural VQA baseline on the same inputs: the same YOLOv11n detections and HSV colors, summarized as a 69-dimensional symbolic vector, and the same test split, gold answers, and metrics; only the answering layer differs. An LSTM encodes the question, the two vectors are fused as $[q; v; q \odot v]$, and a softmax head predicts single answers while a sigmoid head predicts answer sets (340,975 parameters). Hyper-parameters were selected on the validation split only, and results are averaged over five seeds.

Table 13 shows that the YOLO+LSTM baseline comes close to the templates on the primary Track A, where the macro difference is not significant (95% image-level bootstrap CI $[-0.001, 0.021]$). A paired, question-level comparison (Table 14) explains why. The two systems agree on 2,499 of the 2,554 test questions (97.8%): 2,319 are answered correctly by both and 180 incorrectly by both, mostly because of the same detector grade errors. Only 55 questions separate them. The templates alone are correct on 47 of these (e.g., “Which colours appear on the mangoes in the image?”, gold Grey (defect), baseline Green), and the baseline alone on 8 (e.g., “What is the lowest quality grade present in the image?”, gold Class 2, templates Class 1). This 47:8 split is far from the roughly even split expected if both answerers were equally accurate (McNemar exact test, $p < 0.001$), so the templates’ advantage across all 22 types is systematic even though the overall accuracy gap is small (about 1.5 percentage points).

The gap concentrates where exact symbolic operations are needed: object counting (*count_total* 0.986 vs. 0.952), per-label count breakdowns (0.841 vs. 0.808), per-object color (0.975 vs. 0.906), and grade–color composites. Because the baseline treats counts as classes, it cannot output a count or label–count pair it never saw during training; it also needs training and tuning, becomes over-confident (its validation loss rises while accuracy improves), and gives no reasoning trace. The templates therefore match or exceed a trained neural answerer while remaining training-free and fully interpretable.

To test whether a learned visual encoder that bypasses the detector can do better, we also trained a CNN+LSTM baseline using a frozen ImageNet-pretrained ResNet-50 image encoder instead of YOLO. For object-level questions it also sees a crop of the queried fruit taken from its ground-truth box, which means it never misses a fruit and is slightly favored on local questions; everything else, the question encoder, fusion, answer heads, training protocol, and five seeds, matches the YOLO+LSTM baseline (856,943 trained parameters). Averaged over five seeds, this baseline scores 0.891 overall and 0.897 on Track A, below both MangoVQA (0.919, 0.906) and YOLO+LSTM (0.905, 0.900). At the question level, MangoVQA gets right what the CNN misses far more often than the reverse (209 vs. 106, McNemar $p < 0.001$).

The pattern is uneven, though: the CNN edges out MangoVQA on grade recognition by 0.04 to 0.05, for example *object_grade* (0.894 vs. 0.847) and *worst_quality* (0.899 vs. 0.845), and even on Track A's global open-ended questions (0.926 vs. 0.919), but it falls apart wherever an answer depends on position or count, dropping to 0.593 on *grades_on_side* (vs. 0.843) and losing over 0.12 F1 on global multi-answer questions. Because these gains and losses offset each other across question types, the bootstrap interval on the average per-type difference still crosses zero. This lines up with the error analysis in Section IV-G: most of MangoVQA's mistakes trace back to the detector's grade calls rather than the reasoning layer, and a pixel-based encoder happens to read grade slightly better on this largely single-fruit test set, while the templates remain unmatched on anything spatial, countable, or composite, need no training, and always show their reasoning. A second-stage grade classifier on each detected crop looks like the natural next step to close that one remaining gap.

Table 13.

MangoVQA Templates vs. Learned YOLO+LSTM and CNN+LSTM Baselines on The Test Split (Baselines: Mean ± SD over Five Seeds)			
Metric	MangoVQA	YOLO+LSTM	CNN+LSTM
Track A, global open (Acc)	0.919	0.912 ± 0.002	0.926 ± 0.003
Track A, local open (Acc)	0.937	0.944 ± 0.002	0.936 ± 0.004
Track A, global multi (F1)	0.846	0.834 ± 0.005	0.724 ± 0.015
Macro, Track A (15 types)	0.906	0.900 ± 0.002	0.897 ± 0.007
Macro, all 22 types	0.919	0.905 ± 0.001	0.891 ± 0.005

Table 14.

Question-level agreement between MangoVQA and the YOLO+LSTM baseline on the test split ($n = 2,554$)

	Baseline correct	Baseline wrong
MangoVQA correct	2,319	47
MangoVQA wrong	8	180

G. Error Propagation and Inference-Time Effects

To quantify how perception errors propagate to answers, the template engine was re-run on four progressively corrected box sets for every test image: S0, the YOLOv11n predictions; S1, S0 with the class of every matched box ($\text{IoU} \geq 0.5$) replaced by its ground-truth class; S2, S1 with false positives removed and missed objects restored from ground truth; and S3, the ground-truth boxes. Each wrong answer at S0 is attributed to the first correction that makes it correct. On ground-truth boxes (S3), the engine answers all 2,554 test questions correctly, so the reasoning stage contributes no error, and every error originates in perception.

Table 15 breaks down the 188 wrong answers. These 188 errors represent 7.4% of the 2,554 test questions, consistent with the accuracy figures reported in Table 11. Correcting the grade class alone (YOLO misclassification) removes 152 of them (80.9%), correcting spurious or missed boxes removes 18 (9.6%), and correcting box geometry removes the remaining 18: 11 (5.9%) for position- or size-dependent questions and 7 (3.7%) for color questions, i.e., HSV color computed on a displaced box. For Track A, the corresponding shares are 87.2%, 5.4%, 7.4%, and 0%. Because these shares rest on 12 grade confusions in 10 test images, they identify the dominant error source rather than precise proportions. The 15 detection errors on the 71 test images (12 grade confusions, one missed and two spurious boxes) account for 171 wrong answers, about 11 per detection error, because a single mis-graded fruit feeds every grade-dependent question about its image: images without detection errors are answered 99.2% correctly, against 60.0% for images whose only error is a grade confusion. Because Track B gold labels are HSV pseudo-labels, the HSV category measures only the sensitivity of colour answers to box displacement, not agreement with human colour perception.

Table 15.

Sources of Wrong Answers on The Test Split, Attributed by Stepwise Correction of The Detections

Source (first correction that fixes the answer)	All tracks	Track A	Track B	Track C
Grade misclassification (S1)	152 (80.9%)	129 (87.2%)	0 (0.0%)	23 (85.2%)
Spurious or missed boxes (S2)	18 (9.6%)	8 (5.4%)	8 (61.5%)	2 (7.4%)
Box geometry: position/size (S3)	11 (5.9%)	11 (7.4%)	0 (0.0%)	0 (0.0%)
Box geometry: HSV colour (S3)	7 (3.7%)	0 (0.0%)	5 (38.5%)	2 (7.4%)
Template reasoning (never fixed)	0	0	0	0
Wrong/total answers	188/2,554	148/1,603	13/795	27/156
Actual error rate	7.4%	9.2%	1.6%	17.3%
Correct rate	92.6%	90.8%	98.4%	82.7%

Because system answers are computed from detections, detector errors propagate into VQA. Although Reject has the lowest detection recall (0.810; Table 9), it is not the main source of VQA error on the test split (Table 16): all five detected Reject fruits are graded correctly, and the single missed Reject fruit causes the only object_marketability error.

Table 16.

Detection Confusion on the 71 VQA Test Images (Boxes Matched at IoU ≥ 0.5)

Ground truth \ Predicted	Extra Class	Class I	Class II	Reject	Missed
Extra Class (17)	17	0	0	0	0
Class I (38)	3	30	5	0	0
Class II (24)	0	4	20	0	0
Reject (6)	0	0	0	5	1
Spurious boxes	0	1	0	1	–

Grade-dependent errors instead come from confusions between adjacent grades, Class I $\rightarrow$ Class II (9 of 12 confusions) and Class I $\rightarrow$ Extra Class (3 of 12), which propagate to every grade-ranking and grade-counting question about the affected image. By contrast, overall marketability stays at 1.000 because it only needs the presence or absence of any Reject, which is more robust than exact per-object grading; this is the concrete bridge between detection quality and VQA quality on grade-dependent questions. Pinning image size to the training size is the single most important inference setting: at the default size, background leaves are detected as Reject and boxes drift, inflating counts, whereas the pinned configuration removes these spurious detections and lifts count-oriented answers markedly.

H. Interpretability and Deployment Feasibility

Each answer is delivered with three interpretable artifacts: an annotated detection overlay, an HSV attribute view, for example the dark/brown mask that justifies a Reject defect answer, and a reasoning trace. For the open free-text path, the trace records the four parsing stages, so a reader can see exactly why a sentence was routed to a given question type; Track B's separation is itself an interpretable machine learning, preventing an HSV heuristic from appearing to validate itself. Fig. 11 shows a detection overlay alongside the per-pixel HSV defect mask for a Reject fruit.

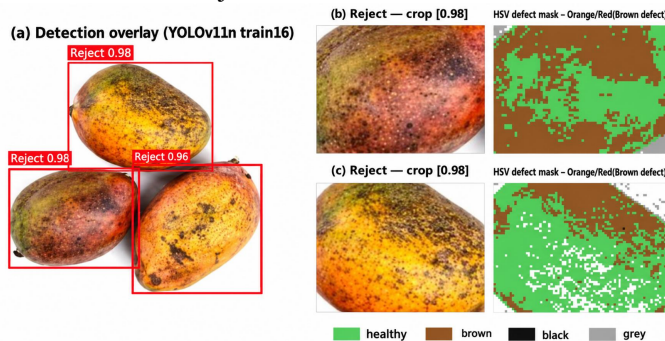


Fig. 11. Explainability example: detection overlay and per-pixel HSV defect mask for a reject fruit.

All training and inference were performed on a mid-range laptop (Intel Core i7-1195G7, NVIDIA GeForce MX450 with 2 GB VRAM), which is why batch size 4 and AMP-off were used. The nano-scale detector, the letterbox and pinned-image-size setting, and the rule-based, LLM-free answering layer make the full pipeline light enough for offline sorting stations and edge inspection without cloud dependence, suiting settings such as trade, customs, and cooperative grading where auditability is required.

To substantiate the deployment claim, the full pipeline was timed on the laptop used throughout this study (Intel Core i7-1195G7, 16 GB RAM, NVIDIA GeForce MX450 with 2 GB VRAM) with Windows set to its best-performance power mode, in three independent runs on the GPU and two on the CPU, each covering the 71 test images three times after ten warm-up inferences. On the GPU, turning an image into its complete answer bank takes 16.5 ms on average (run means 15.0–19.1 ms; about 61 images/s), of which YOLOv11n detection accounts for 13.3 ms and the template and HSV colour stage for 0.9 ms; answering a free-text question (parsing P1–P4 plus answer-bank lookup) takes a further 0.06 ms. On the CPU alone, the same steps take 60.4 ms per image (59.6–61.2 ms; about 16.5 images/s) and 0.11 ms per question. The 2.59 M-parameter detector (5.2 MB) peaks at 183 MB of allocated GPU memory, and the CPU-only process peaks at 525 MB of RAM. GPU timings depend on the laptop's power mode and thermal state, whereas the reasoning stage stays near 1 ms per image in every run, so nearly all latency comes from detection, and the pipeline runs on a mid-range laptop even without a GPU.

V. CONCLUSION

MangoVQA makes three main contributions. First, it is the first multi-answer Visual Question Answering system aligned with a formal national quality standard (SNI 3164:2024). Second, it releases 17,764 bilingual (Indonesian English) question-answer pairs across 468 annotated mango images as a public resource for future agricultural VQA research. Third, it introduces an auditable, lightweight deployment architecture that runs on commodity hardware, in which every answer is traceable to the specific detection and rule that produced it. A trained YOLO+LSTM answerer given the same detections did not surpass the templates, and a CNN+LSTM baseline that reads the image directly scored lower overall while recognizing grades slightly better, confirming that the rule-based layer is competitive without any training.

Because every answer can be traced back to a specific detected fruit and the rule that generated it, MangoVQA supports traceable quality assessment in settings where accountability matters most: smallholder farmers seeking consistent self-grading feedback, cooperatives standardizing sorting across multiple workers, and exporters needing auditable evidence of SNI compliance for regulatory or buyer verification.

Future works, first, the framework will be extended to other fruit varieties and their corresponding national or international grading standards, testing the generality of the

perception–reasoning separation beyond mango. Second, the current color attribute extraction relies on HSV color-space thresholding as a deterministic baseline and proof-of-concept for domain rule alignment; color answers therefore function as HSV pseudo-labels rather than human-verified ground truth, leaving the module sensitive to extreme lighting variations, specular reflections, and complex background shadows in unconstrained post-harvest environments. Future deployments will integrate a lightweight instance segmentation model (e.g., YOLOv11-seg) alongside expanded imaging conditions to achieve illumination-invariant fruit isolation and validate color labels against independent human perception. Third, system answers inherit YOLO detection error: adjacent-grade confusion (chiefly Class I and Class II) is the main source of grade-dependent VQA errors, a pattern consistent with these grades' close visual similarity under SNI's percentage-based thresholds, while the Reject class's lower detection recall (0.810) had limited practical impact, contributing only one missed fruit on the test split; we therefore aim to sharpen the boundary between adjacent grades to lift the grade-dependent VQA performance ceiling. Finally, the rule-based parser guarantees gold–system consistency but limits linguistic variety to the provided templates: because it performs deterministic pattern matching against a controlled vocabulary, free-text queries containing typographical errors, unlisted synonyms, or ungrammatical phrasing outside this rule set are not guaranteed to be correctly classified and instead return a fallback “intent not recognized” response. Planned extensions include replacing this parser with a learned or Transformer question encoder to broaden linguistic coverage and support open-domain inquiries beyond controlled grading vocabularies, while retaining traceability.

DATA AVAILABILITY

For reproducibility, the full dataset containing all 17,764 bilingual QA pairs (English and Indonesian), and full JSON annotations is made publicly available at <https://github.com/nhasanati/MangoVQA>. The annotated mango images are available at <https://github.com/nhasanati/Mango-YOLO>.

ACKNOWLEDGMENT

The authors acknowledge IPB University, Bogor, Indonesia, and the State Islamic University (UIN) Syarif Hidayatullah Jakarta, Indonesia, for their continuous support for this research, particularly through the Directorate of Research and Community Engagement.

REFERENCES

- [1] H. M. R. Iqbal and A. Hakim, “Classification and grading of harvested mangoes using convolutional neural network,” *Int. J. Fruit Sci.*, vol. 22, no. 1, pp. 95–109, 2022, doi: 10.1080/15538362.2021.2023069.
- [2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *Proc. IEEE Comput. Soc. Conf.*

- Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.
- [3] R. Khanam and M. Hussain, “YOLOv11: An overview of the key architectural enhancements,” *arXiv (Cornell University)*, Oct. 2024, doi: 10.48550/arxiv.2410.17725.
- [4] N. Jegham, C. Y. Koh, M. Abdelatti, and A. Hendawi, “YOLO evolution: A comprehensive benchmark and architectural review of YOLOv12, YOLOv11, and their previous versions,” *arXiv.org*, Oct. 31, 2024, <https://arxiv.org/abs/2411.00201>
- [5] National Standardization Agency, “SNI 3164-2024: Mango (in Indonesian),” Jakarta, Indonesia, 2024. Accessed: May 18, 2026. [Online]. Available: <https://akses-sni.bsn.go.id>
- [6] N. Hasanati, T. Djatna, I. S. Sitanggang, and A. I. Suroso, “Evaluating Rare-Class Modeling in YOLOv11n for Imbalanced Agricultural Object Detection: A Mango Case Study,” 2026, <https://github.com/nhasanati/Mango-YOLO>
- [7] J. S. Ignacio, K. N. A. Eisma, and M. V. C. Caya, “A YOLOv5-based deep learning model for in-situ detection and maturity grading of mango,” in *2022 6th International Conference on Communication and Information Systems (ICCIS)*, 2022, pp. 141–147, doi: 10.1109/ICCIS56375.2022.9998163.
- [8] S. Arwathchananukul, S. Wongsawat, S. Chaiwong, M. Chen, and R. Saengrayap, “Automated physical feature extraction of namdokmai sithong mangoes using YOLOv8 and image processing techniques,” *AgriEngineering*, vol. 7, no. 9, Art. no. 312, Sep. 2025, doi: 10.3390/agriengineering7090312.
- [9] A. Sopian, “System detection ripeness of fresh fruits bunch palm oil with YOLOv4,” *Proc. Int. Conf. Ind. Eng. Oper. Manag. Istanbul, Turkey*, no. Usda 2021, pp. 5307–5318, 2022, [Online]. Available: <https://index.icomsociety.org/index.cfm/item/41183>
- [10] J. W. Lai, H. R. Ramli, L. I. Ismail, W. Zuha, and W. Hasan, “Real-time detection of ripe oil palm fresh fruit bunch based on YOLOv4,” *IEEE Access*, vol. 10, no. September, pp. 95763–95770, 2022, doi: 10.1109/ACCESS.2022.3204762.
- [11] H. Wibowo, I. S. Sitanggang, M. Mushthofa, and H. A. Adrianto, “Large-scale oil palm trees detection from high-resolution remote sensing images using deep learning,” *Big Data Cogn. Comput.*, vol. 6, no. 3, Art. no. 89, 2022, doi: 10.3390/bdcc6030089.
- [12] Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. van den Hengel, “Visual question answering: A survey of methods and datasets,” *Comput. Vis. Image Underst.*, vol. 163, pp. 21–40, 2017, doi: 10.1016/j.cviu.2017.05.001.
- [13] S. Antol *et al.*, “VQA: Visual question answering,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter. pp. 2425–2433, 2015, doi: 10.1109/ICCV.2015.279.
- [14] Y. Goyal, T. Khot, A. Agrawal, D. Summers-Stay, D. Batra, and D. Parikh, “Making the V in VQA matter: Elevating the role of image understanding in visual question answering,” *Int. J. Comput. Vis.*, vol. 127, no. 4, pp. 398–414, 2019, doi: 10.1007/s11263-018-1116-0.
- [15] Q. Wu, P. Wang, and X. Wang, *Visual Question Answering From Theory to Application*. Springer, 2022.
- [16] S. Barra, C. Bisogni, M. De Marsico, and S. Ricciardi, “Visual question answering: Which investigated applications?,” *Pattern Recognit. Lett.*, vol. 151, pp. 325–331, 2021, doi: 10.1016/j.patrec.2021.09.008.
- [17] S. H. Hosseinabad, M. Safayani, and A. Mirzaei, “Multiple answers to a question: a new approach for visual question answering,” *Vis. Comput.*, vol. 37, no. 1, pp. 119–131, 2021, doi: 10.1007/s00371-019-01786-4.
- [18] M. Luo *et al.*, “‘Just because you are right, doesn’t mean I am wrong’: Overcoming a bottleneck in the development and evaluation of open-ended visual question answering (VQA) tasks,” *EACL 2021 - 16th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference*. pp. 2766–2771, 2021, doi: 10.18653/v1/2021.eacl-main.240.
- [19] N. Hasanati, T. Djatna, I. S. Sitanggang, and A. I. Suroso, “Exploring multi-answer visual question answering with object detection: a systematic review,” *Int. J. Informatics Commun. Technol.*, vol. 15, no. 3, pp. 1097–1114, 2026, doi: 10.11591/ijict.v15i3.pp1097-1114.
- [20] S. Harnad, “The Symbol Grounding Problem,” *Phys. D Nonlinear Phenom.*, vol. 42, pp. 335–346, 1990, doi: 10.1016/0167-2789(90)90087-6.

- [21] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu, "The Neuro-Symbolic Concept Learner: Interpreting Scenes, Words, and Sentences From Natural Supervision," *7th Int. Conf. Learn. Represent. ICLR 2019*, pp. 1–28, 2019, [Online]. Available: https://api.elsevier.com/content/abstract/scopus_id/85083954234
- [22] A. d'Avila Garcez and L. C. Lamb, "Neurosymbolic AI: The 3rd wave," *Artif. Intell. Rev.*, vol. 56, no. 11, pp. 12387–12406, 2023, doi: 10.1007/s10462-023-10448-w.
- [23] K. Yi, A. Torralba, J. Wu, P. Kohli, C. Gan, and J. B. Tenenbaum, "Neural-symbolic VQA: Disentangling reasoning from vision and language understanding," *Advances in Neural Information Processing Systems*, vol. 2018-Decem. pp. 1031–1042, 2018. [Online]. Available: https://api.elsevier.com/content/abstract/scopus_id/85064824191
- [24] A. M. Kiloos and D. Joyce, "Exploring the challenges and opportunities of mango export from Indonesia: Insights from stakeholder interviews," *Qual. Report-NSUWorks*, vol. 29, no. 3, pp. 811–830, 2024, doi: 10.46743/2160-3715/2024.6343 This.
- [25] National Standardization Agency, "SNI 3164:2009: Mango (in Indonesian)," Jakarta, Indonesia, 2009. [Online]. Available: <https://akses-sni.bsn.go.id>
- [26] ASEAN, "ASEAN STANDARD FOR MANGO (ASEAN Stan 2:2006, Rev.1-2012)," 2012. [Online]. Available: <https://www.scribd.com/document/421547432/MANGO-Asean-Standard>
- [27] FAO/WHO Codex Alimentarius Commission, "Codex Standard for Mangoes (CODEX STAN 184-1993, Amended 2005)," FAO/WHO, Rome, Italy, 1993. [Online]. Available: https://www.fao.org/input/download/standards/315/315/CXS_184e.pdf
- [28] K. Kafle and C. Kanan, "Visual question answering: Datasets, algorithms, and future challenges," *Comput. Vis. Image Underst.*, vol. 163, pp. 3–20, 2017, doi: 10.1016/j.cviu.2017.06.005.
- [29] K. Bayoudh, R. Knani, F. Hamdaoui, and A. Mtibaa, "A survey on deep multimodal learning for computer vision: advances, trends, applications, and datasets," *Vis. Comput.*, vol. 38, no. 8, pp. 2939–2970, 2022, doi: 10.1007/s00371-021-02166-7.
- [30] M. T. Desta, L. Chen, and T. Kornuta, "Object-based reasoning in VQA," *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018*, vol. 2018-Janua. pp. 1814–1823, 2018. doi: 10.1109/WACV.2018.00201.
- [31] M. Malinowski and M. Fritz, "A multi-world approach to question answering about real-world scenes based on uncertain input," *Adv. Neural Inf. Process. Syst.*, vol. 2, no. January, pp. 1682–1690, 2014.
- [32] S. Upadhyay and S. S. Tripathy, "Enhancing VQA with emphasis-based soft attention," *Signal, Image Video Process.*, vol. 19.0, no. 13, 2025, doi: 10.1007/s11760-025-04710-7.
- [33] Y. Miao, W. Cheng, S. He, and H. Jiang, "Research on visual question answering based on GAT relational reasoning," *Neural Process. Lett.*, vol. 54, no. 2, pp. 1435–1448, 2022, doi: 10.1007/s11063-021-10689-2.
- [34] H. Liu *et al.*, "Multi-granularity feature interaction and multi-region selection based triplet visual question answering," *IEEE Trans. Big Data*, 2024, doi: 10.1109/TBDATA.2024.3453750.
- [35] H. Liu, B. Wang, X. Li, Y. Sun, Y. Hu, and B. Yin, "ICQ-TransE: LLM-enhanced image-caption-question translating embeddings for knowledge-based visual question answering," *IEEE Trans. Artif. Intell.*, 2025, doi: 10.1109/TAI.2025.3575553.
- [36] Z. Liu, K. Yang, Y. Weng, Z. He, X. Liu, and H. Gao, "SCAG: Semantic co-occurring attention guided alignment for knowledge-based visual question answering," *ACM Trans. Multimed. Comput. Commun. Appl.*, vol. 21.0, no. 7, 2025, doi: 10.1145/3734220.
- [37] Y. Yuan, J. Zeng, and S. Shiguang, "Exp-VQA: Fine-grained facial expression analysis via visual question answering," *Pattern Recognit.*, vol. 168.0, 2025, doi: 10.1016/j.patcog.2025.111783.
- [38] J. Zhang, B. Li, and S. Zhou, "Hierarchical modeling for medical visual question answering with cross-attention fusion," *Appl. Sci.*, vol. 15, no. 9, 2025, doi: 10.3390/app15094712.
- [39] Z. Liu, D. Zhao, B. Yuan, and Z. Jiang, "RescueADI: Adaptive disaster interpretation in remote sensing images with autonomous agents," *IEEE Trans. Geosci. Remote Sens.*, vol. 63, 2025, doi: 10.1109/TGRS.2025.3532594.
- [40] B. Liu, L. M. Zhan, L. Xu, L. Ma, Y. Yang, and X. M. Wu, "Slake: A semantically-labeled knowledge-enhanced dataset for medical visual question answering," *Proceedings - International Symposium on Biomedical Imaging*, vol. 2021-April. pp. 1650–1654, 2021. doi: 10.1109/ISBI48211.2021.9434010.
- [41] H. Demirhan and W. Zadrozny, "Survey of multimodal medical question answering," *BioMedInformatics*, vol. 4, no. 1, pp. 50–74, 2024, doi: 10.3390/biomedinformatics4010004.
- [42] Z. Li *et al.*, "HortiVQA-PP: Multitask framework for pest segmentation and visual question answering in horticulture," *Horticulturae*, vol. 11, no. 9, Art. no. 1009, Aug. 2025, doi: 10.3390/horticulturae11091009.
- [43] P. Anderson *et al.*, "Bottom-up and top-down attention for image captioning and visual question answering," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. pp. 6077–6086, 2018. doi: 10.1109/CVPR.2018.00636.
- [44] S. Shahane, "Mango Varieties Classification and Grading," 2021. <https://www.kaggle.com/datasets/saurabhshahane/mango-varieties-classification>
- [45] K. Ratprakhon, W. Neubauer, K. Riehn, J. Fritsche, and S. Rohn, "Developing an automatic color determination (mangifera indica) using a CCD camera and color standards," *Foods*, vol. 9, no. 11, Art. no. 1709, 2020, doi: 10.3390/foods9111709.