Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PAPER

View Article Online
View Journal | View Issue

Cite this: *Digital Discovery*, 2026, **5**, 1172

Food additive lens: an on-device AI application for real-time science-based consumer education on food additives using retrieval-augmented generation

Yihang Feng, [ab] Yi Wang, [a] Xinhao Wang,[a] Bo Zhao,[b] Jinbo Bi,[b] Song Han*[b] and Yangchao Luo *[a]

Consumer concerns about food additives have intensified amid widespread misinformation, with the 2024 IFIC survey revealing that 35% of consumers actively avoid artificial ingredients despite authoritative safety data existing in FDA and USDA databases. This work investigates whether on-device artificial intelligence can effectively translate complex regulatory information into accessible consumer education while maintaining scientific accuracy and privacy. This paper presents Food Additive Lens (FAL), an iOS application implementing a three-agent architecture: (1) a food category classifier achieving 87.2% top-3 accuracy across 257 categories, (2) a hybrid additive identifier combining database lookup with AI extraction (F1-score: 0.757), and (3) an explanation generator producing contextualized, consumer-friendly descriptions. The system deploys Meta's Llama 3.2 3B model quantized to 1.8 GB through 4-bit compression, achieving a generation speed of 13–30 tokens/second while operating entirely offline. Integration of FDA's Substances Added to Food Inventory (3971 substances) and USDA's Global Branded Food Products Database enables comprehensive coverage with direct links to the Code of Federal Regulations for professional users. The Retrieval-Augmented Generation workflow grounds AI responses in authoritative sources, reducing hallucination while maintaining accessibility. Performance evaluation on iPhone 14 and MacBook Air M1 demonstrated stable memory usage (peak: 2.36 GB) with complete offline functionality, ensuring user privacy. The application transforms complex ingredient lists into accessible information through camera-based OCR scanning, progressive disclosure interfaces, and context-aware explanations tailored to specific food products. This work demonstrates the feasibility of deploying sophisticated AI for science communication on consumer devices, offering a scalable model for combating food-related misinformation while preserving privacy and accessibility.

Received 3rd October 2025
Accepted 6th February 2026

DOI: 10.1039/d5dd00444f

rsc.li/digitaldiscovery

1 Introduction

Food additives have been integral to human food systems for millennia, from ancient salt preservation techniques to modern synthetic antioxidants that prevent rancidity and extend shelf life. The modern food industry utilizes over 10 000 different additives globally, with regulatory frameworks varying significantly across jurisdictions.[1,2] In the United States alone, the FDA's Substances Added to Food Inventory (SAFI) catalogs nearly 4000 approved substances, each serving specific technological functions ranging from antimicrobial preservation to texture modification, color stabilization, and nutritional fortification.[3] Despite rigorous safety assessments and regulatory oversight, public perception of food additives has become increasingly negative over the past decade, creating a significant disconnect between scientific evidence and consumer beliefs that has profound implications for public health policy and food industry practices.[4]

Recent consumer research reveals the depth of this perception gap. The 2024 International Food Information Council Food and Health Survey[5] found that 35% of consumers actively try to limit or avoid artificial ingredients/colors and 25% of consumers' top choice of definition of healthy food is "limited or no artificial ingredients or preservatives", with younger demographics showing even higher avoidance rates despite having less nutritional knowledge overall. This widespread apprehension is not merely a preference but often manifests as food anxiety, with studies documenting that concerns about additives contribute to disordered eating patterns and unnecessary dietary restrictions.[6] The phenomenon has been further amplified by social media, where misinformation about food

[a]*Department of Nutritional Sciences, University of Connecticut, Storrs, CT 06269-4017, USA. E-mail: yangchao.luo@uconn.edu; Web: https://yangchao-luo.uconn.edu/; Fax: +860-486-3674; Tel: +860-486-2180*

[b]*School of Computing, University of Connecticut, Storrs, CT 06269-4017, USA. E-mail: song.han@uconn.edu; Web: https://cps.cse.uconn.edu/; Tel: +860-486-8771*

ingredients spreads rapidly through viral posts and influencer content. A recent study[7] found that nutritional misinformation, particularly on platforms like TikTok, was not only prevalent but also significantly more engaging than accurate content, with inaccurate posts receiving higher likes and comments. This dynamic fosters a digital environment where fear-based narratives about food additives dominate public discourse, often outpacing factual, science-based communication. Social media platforms tend to amplify emotionally charged and sensational content, which includes misinformation about food ingredients. For instance, negative claims about additives, especially those framed around health risks, tend to receive significantly more engagement than posts presenting scientific evidence or regulatory context.[8] This amplification contributes to a broader phenomenon known as chemophobia, a disproportionate fear of chemicals in food, which can distort public understanding and hinder informed decision-making.[9] As a result, legitimate safety discussions are frequently overshadowed by pseudoscientific claims, making it increasingly difficult for consumers to distinguish between evidence-based concerns and unfounded fears.

The challenge of food additive communication is compounded by the complexity of chemical nomenclature and regulatory language used in ingredient lists. Consumer research demonstrates significant barriers to understanding food ingredient information, with studies revealing that consumers frequently rely on simplified heuristics when interpreting chemical names on food labels. Aschemann-Witzel *et al.* found that consumers perceived additives as more harmful when the additives had names that were difficult to pronounce, indicating that unfamiliarity creates greater risk perception and leads to avoidance behaviors.[10] For instance, the same consumer who might be comfortable with "vitamin C" may experience concern when encountering "ascorbic acid" on a label, despite these being identical substances. This nomenclature barrier represents a fundamental obstacle to informed food choices, as consumers demonstrate a strong preference for ingredients with "familiar or recognizable" names rather than "chemical-sounding" names.[11] Eye-tracking studies conducted on food label reading behavior reveal that consumers demonstrate systematic but brief visual attention patterns when examining ingredient information. Research using mobile eye-tracking technology found that participants' visual attention to health labels was significantly reduced under time constraints, with consumers spending limited time processing complex ingredient information.[12] The resulting cognitive overload leads many consumers to rely on simplified decision-making strategies, such as avoiding products with longer ingredient lists or unfamiliar chemical names, heuristics that often lead to suboptimal nutritional choices.[13]

The current landscape of digital tools addressing food additive information reveals significant gaps in meeting consumer needs for accurate, accessible, and actionable information. Analysis of mobile health applications in the nutrition domain shows that most food scanning apps lack comprehensive data validation and evidence-based information sources. Research examining food tracking mobile applications found that only a small percentage incorporate data from authoritative regulatory sources, with many relying on simplified scoring algorithms that may perpetuate misconceptions rather than provide educational value.[14] Among applications that do reference nutritional databases, studies indicate significant limitations in providing contextual, product-specific explanations that account for individual dietary needs or knowledge levels.[15] Furthermore, existing solutions predominantly operate on client-server architectures, requiring constant internet connectivity and raising substantial privacy concerns. Comprehensive privacy assessments of mobile health applications reveal widespread data collection practices, with research showing that 88.0% of analyzed mHealth apps included code that could potentially collect user data, and 28.1% provided no privacy policy at all.[16] Analysis of mHealth app privacy policies demonstrates that a significant proportion collect and share user dietary data with third parties for marketing purposes, with many building behavioral profiles that could potentially be used for discriminatory practices.[17] This privacy-functionality trade-off forces consumers to choose between accessing information about their food and protecting their personal health data, a choice that becomes particularly problematic for individuals with specific dietary requirements or stigmatized health conditions.[18]

The technical challenges of developing effective food additive education tools extend beyond simple database queries. Food additives often serve multiple functions depending on the food matrix, processing conditions, and interactions with other ingredients. Research on food matrices demonstrates that additives exhibit context-dependent functionality, with studies showing that the same additive can perform different roles based on environmental factors such as pH, temperature, and the presence of other compounds.[19] For example, citric acid may function as an acidulant in beverages, a chelating agent in canned vegetables, or a flavor enhancer in confectionery products, requiring context-aware explanation systems that can account for these nuances.[20] Additionally, the same additive may be derived from different sources or produced through various methods (synthetic, fermentation, and extraction), each with different implications for consumers with specific dietary restrictions or preferences.[21] Current database structures and query systems struggle to capture these multidimensional relationships, resulting in oversimplified or potentially misleading information when translated for consumer audiences.[22]

Recent advances in artificial intelligence, particularly in natural language processing and on-device deployment, offer unprecedented opportunities to address these challenges. The development of transformer-based language models has revolutionized the ability of machines to understand and generate human-like text, with models demonstrating remarkable capability in translating technical information into accessible explanations.[23] Studies on transformer models for text simplification have shown significant improvements in converting complex scientific language into plain language formats, particularly for domain-specific applications such as biomedical text.[24] However, deploying these models on mobile devices has historically been infeasible due to their massive computational requirements, with

popular models requiring gigabytes of memory and server-grade processing power.[25] Research on mobile AI deployment challenges reveals that large language models (LLMs) typically require hundreds of megabytes of memory footprints, making it challenging to deploy on resource-constrained platforms such as mobile devices and IoT systems.[26]

This limitation has been dramatically altered by recent breakthroughs in model compression techniques. Quantization methods, which reduce the precision of model weights from 32-bit floating-point to as low as 4-bit integers, have demonstrated the ability to compress LLMs by factors of 8–10x while maintaining over 95% of their original performance on domain-specific tasks.[27,28] Studies on deep neural network quantization for mobile deployment show that post-training quantization can achieve up to a 95% reduction in parameters while maintaining model accuracy, making deployment on edge devices feasible.[29] These advances, combined with hardware acceleration frameworks specifically designed for mobile devices such as Apple's MLX and Google's MediaPipe, have made it possible to run sophisticated AI models entirely on consumer smartphones.[30,31] MLX provides optimized machine learning (ML) inference for Apple silicon through unified memory architecture and efficient computation graphs, while MediaPipe enables cross-platform deployment of ML pipelines with GPU acceleration and multi-threading capabilities.

The emergence of Retrieval-Augmented Generation (RAG) architectures represents another crucial development for domain-specific AI applications. Unlike traditional language models that rely solely on patterns learned during training, RAG systems dynamically retrieve relevant information from external knowledge bases to ground their responses in authoritative sources.[32] In scientific and medical domains, RAG implementations have demonstrated significant reductions in hallucination rates—instances where AI generates plausible but factually incorrect information. Béchard[33] demonstrated that RAG systems can dramatically improve the quality of structured outputs while reducing hallucinations in enterprise applications, with their implementation showing substantial improvements in generalization to out-of-domain settings. Similarly, Shuster *et al.*[34] found that retrieval augmentation significantly reduces hallucination in conversational AI systems, particularly when querying based on complex multi-turn dialogue contexts. For food science applications, where accuracy is paramount and misinformation could have health implications, the ability to anchor AI responses in regulatory databases and peer-reviewed literature is essential. Recent systematic reviews of RAG applications in educational contexts have shown that users consistently rate RAG-enhanced responses as more trustworthy and actionable compared to traditional language model outputs, with trust improvements observed when sources are explicitly referenced and retrieved information is validated against authoritative knowledge bases.[32]

The intersection of privacy concerns and AI deployment has become increasingly critical as consumers become more aware of data collection practices. Traditional cloud-based AI services require transmitting user queries to remote servers, creating permanent records of personal interests and concerns that can be aggregated into detailed behavioral profiles.[35] For health-related queries, including those about food and nutrition, this raises significant ethical and legal concerns under frameworks such as The General Data Protection Regulation (GDPR) and The Health Insurance Portability and Accountability Act of 1996 (HIPAA).[36] Studies examining privacy in AI healthcare applications have documented widespread data collection practices, with comprehensive assessments revealing that traditional cloud-based systems create substantial privacy risks through data transmission and storage in external servers.[37] On-device AI processing eliminates these privacy risks by ensuring that all computation occurs locally on the user's device, with no data transmission required. This approach aligns with the principle of data minimization and provides users with complete control over their information, addressing one of the primary barriers to adoption of digital health tools. Recent research on user acceptance of privacy-preserving AI applications has found empirical evidence supporting increased willingness to use health-related AI tools when guaranteed on-device processing is implemented. Wang *et al.*[38] demonstrated that local data processing significantly enhances user privacy protection in health monitoring applications, while survey studies examining technology acceptance in healthcare contexts have shown that privacy concerns are among the most significant factors influencing user adoption of AI-powered health applications.[39]

This paper presents Food Additive Lens (FAL), a novel iOS application that synthesizes recent advances in on-device AI, RAG, and food science communication to address the critical gap between scientific knowledge about food additives and consumer understanding. The application implements a three-agent AI architecture, comprising a food category classifier, additive identifier, and explanation generator, that works in concert to provide contextual, accurate, and accessible information about food additives. By deploying a quantized version of Meta's Llama 3.2 3B model (compressed to 1.8 GB through 4-bit quantization) directly on iOS devices, the system achieves processing speeds of 13–30 tokens per second while maintaining complete offline functionality. The integration of SAFI and USDA's Global Branded Food Products Database (GBFPD) through an embedding-based search system enables the application to provide authoritative information for nearly 4000 additives across one million food products, with direct links to relevant Code of Federal Regulations (CFR) sections for professional users. This work demonstrates the feasibility of deploying sophisticated AI systems for food science education on consumer devices while maintaining privacy, accuracy, and accessibility. The implications extend beyond food additives to suggest new paradigms for science communication in an era of information overload and digital misinformation, where the challenge is not the absence of authoritative information but rather its translation and delivery at the point of need.

2 Methods

2.1 System architecture and design

FAL implements a three-agent artificial intelligence architecture deployed entirely on iOS devices, eliminating the need for server

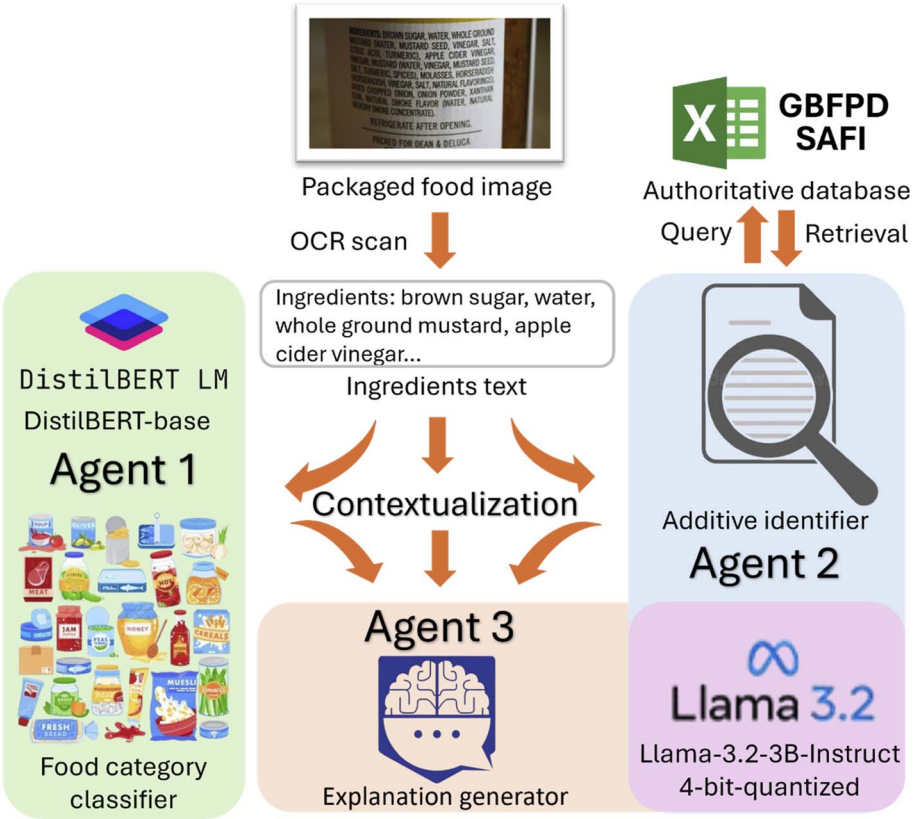

Fig. 1 The three-agent system of FAL.

communication while maintaining sophisticated analytical capabilities. The system comprises three specialized agents working in concert as shown in Fig. 1: (1) a food category classifier that identifies product types to provide contextual understanding, (2) an additive identifier that extracts food additives using a hybrid approach combining direct database lookup and AI-powered analysis, and (3) an explanation generator that produces accessible, contextualized explanations tailored to specific food products. This multi-agent design ensures both accuracy through systematic analysis and relevance through context-aware processing.

The application architecture follows a modular design pattern implemented in Swift using SwiftUI for the user interface layer. The core system integrates Apple's MLX framework for on-device ML acceleration, enabling the deployment of a quantized Llama 3.2 3B language model compressed to 1.8 GB through 4-bit quantization. The architecture maintains complete offline functionality by embedding all necessary databases and models within the application bundle, including SAFI, GBFPD samples, and pre-computed embeddings for semantic search capabilities.

The iOS application implements the Model-View-ViewModel (MVVM) architectural pattern[40] using SwiftUI's reactive framework. The architecture separates concerns through three distinct layers: (1) models representing data structures (FoodRecord, ClassificationResult, and AdditiveKnowledge), (2) ViewModels managing business logic and state (MLXAdditiveIdentifier, AdditiveKnowledgeManager, CFRManager, and FoodCategoryClassifier) marked with '@Observable' for reactive updates, and (3) views implemented in SwiftUI that bind reactively to ViewModel state changes. This MVVM implementation ensures clear separation of concerns, facilitates unit testing, and maintains efficient UI updates through SwiftUI's declarative binding system.

The RAG workflow forms the backbone of the explanation system as shown in Fig. 2. When processing user queries, the system first generates search embeddings using a custom embedding function that captures both lexical and semantic features of food additive names. These embeddings enable similarity-based retrieval from the knowledge base containing 3971 FDA-approved substances with their technical effects and regulatory information. Retrieved information is then augmented with contextual data from the food category classification before being passed to the language model for explanation generation, ensuring that responses are grounded in authoritative sources while remaining accessible to consumers.

2.2 Data sources and preparation

2.2.1 FDA substances added to food database. The primary knowledge base derives from SAFI (last updated on February 13, 2025), containing comprehensive regulatory information for food additives approved for use in the United States. The raw dataset included 3971 unique substances with associated metadata across 37 columns, including CAS Registry Numbers,

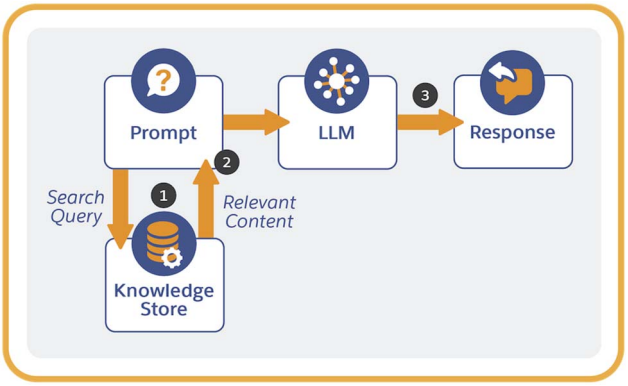

Fig. 2 RAG workflow for food additive explanation generation. The system employs a three-step process: (1) user queries generate search embeddings to retrieve relevant content from SAFI, (2) retrieved authoritative information is combined with the user prompt, and (3) the LLM generates contextualized, consumer-friendly explanations grounded in regulatory sources.

substance names, alternative nomenclature, technical effects, and CFR references. Data preprocessing involved several critical steps to ensure accuracy and usability within the mobile application context.

The cleaning pipeline addressed multiple data quality issues inherent in the government database export. HTML entities and formatting artifacts (*e.g.*, special characters and embedded HTML markup tags) were systematically removed using regular expression patterns and HTML unescaping utilities. The "Other Names" field, containing pipe-delimited alternative nomenclature for each substance, required special handling to preserve synonym relationships while removing redundant entries. Technical effect descriptions were normalized from their original all-caps format to sentence case, with multi-effect entries separated by pipe delimiters for structured retrieval. Missing data analysis revealed that 183 substances (4.6%) lacked technical effect descriptions and 26 substances (0.7%) had no alternative names listed, necessitating fallback strategies in the retrieval system.

For efficient on-device search capabilities, we generated semantic embeddings for all substances using a custom embedding algorithm optimized for chemical nomenclature. The embedding generation process created 384-dimensional vectors capturing both character-level patterns common in chemical names (*e.g.*, suffixes like "-ate", "-ine", and "-ide") and word-level semantic features. These embeddings enable fuzzy matching for additive identification, crucial for handling variations in naming conventions and potential OCR errors from ingredient label scanning.

2.2.2 USDA Global Branded Food Products Database. The USDA FoodData Central's GBFPD, last updated on April 24, 2025, provided training data for the food category classifier and real-world ingredient examples for the sampling feature. The database export contained 1 048 575 branded food products with detailed compositional data, though memory constraints required selective sampling for mobile deployment. We extracted 10% (104K) representative products across diverse food categories, prioritizing entries with complete ingredient lists exceeding 50 characters and containing multiple ingredients separated by commas. The original dataset contained diverse branded food category labels that were used directly without consolidation, resulting in 257 unique categories for classification. Each product's ingredient list underwent lowercase conversion and special character normalization to create consistent training examples.

The parsing pipeline for CSV data required specialized handling due to encoding issues and complex field structures. The original files used latin-1 encoding with nested quotation marks and comma-delimited fields containing internal commas. We developed a custom CSV parser that correctly handled these edge cases, extracting seven essential fields: FDC ID, brand owner, brand name, sub-brand name, ingredients, branded food category, and product description. These structured data enable the random sampling feature ($n = 49$ k) on-device, allowing users to explore real product examples while learning about food additives in context.

2.3 Three-agent AI implementation

2.3.1 Agent 1: food category classifier. The food category classifier employs a DistilBERT-base transformer model[41] fine-tuned on the GBFPD to identify product types from ingredient lists. The model architecture consists of a pre-trained DistilBERT-base-uncased model with a custom classification head mapping to 257 food categories. The model training employed Hugging Face's Transformers framework with optimized hyperparameters for large-scale classification. Training utilized the consolidated GBFPD with ingredients as input sequences and branded food categories as labels. The training configuration included 3 epochs with a batch size of 500 samples per device, 500 warmup steps, and 0.01 weight decay. Mixed precision training (fp16) was enabled for memory efficiency, with evaluation performed every 500 steps. The Trainer class from Transformers managed the training loop with DataCollatorWithPadding for dynamic batching. This configuration achieved 90.8% top-1 accuracy on a held-out test set of 10% (104K) products.

For mobile deployment, the trained PyTorch model underwent conversion to Core ML format using coremltools with post-training quantization. The conversion process included input tokenization compatibility layers to handle the DistilBERT vocabulary of 30 522 tokens within the Core ML framework. The final model package occupies 134 MB in the application bundle, with inference times within 0.5 s on iPhone 14 or newer devices. The tokenization pipeline implements special token handling for [CLS] and [SEP] markers while maintaining a maximum sequence length of 256 tokens to accommodate lengthy ingredient lists.

2.3.2 Agent 2: hybrid additive identifier. The additive identification system implements a dual-pathway approach combining deterministic database lookups with probabilistic AI extraction. The direct lookup pathway performs exact and fuzzy string matching against SAFI, handling common variations such as parenthetical descriptors (*e.g.*, "citric acid

(preservative)") and alternative naming conventions. This pathway includes specialized handling for color additives (mapping colloquial names like "Red 40" to official nomenclature "FD&C RED No. 40") and flavor categories (distinguishing between "natural flavors," "artificial flavors," and combinations thereof).

The AI-powered extraction pathway utilizes the on-device Llama 3.2 3B model[42] with carefully crafted prompts optimized for additive identification. The prompt engineering process involved iterative refinement to minimize false positives while maintaining high recall for less common additives. The system prompt explicitly defines inclusion criteria (preservatives, emulsifiers, stabilizers, artificial colors, *etc.*) and exclusion criteria (basic ingredients like flour, water, and sugar) to guide the model's extraction behavior. Post-processing validates AI-extracted additives against SAFI, filtering results through cosine similarity thresholds (empirically set at 0.244) to balance precision and recall.

2.3.3 Agent 3: contextualized explanation generator. The explanation generator leverages the quantized Llama 3.2 3B model to transform technical additive information into consumer-friendly explanations tailored to specific food contexts. The generation pipeline receives three inputs: (1) the identified additives with their technical effects from the knowledge base, (2) the food category classification result, and (3) the original ingredient list for maintaining contextual accuracy. The system implements a streaming generation approach using MLX's token-by-token output, achieving 13–30 tokens per second depending on device capabilities and thermal conditions.

Prompt engineering for the explanation generator followed evidence-based principles for technical communication to lay audiences. The system prompt instructs the model to use exact additive names as they appear in the original ingredients, explain each additive's function in 1–2 sentences using plain English, focus on why additives are used rather than chemical properties, and group similar additives when appropriate. The prompt explicitly incorporates food context, adjusting explanations based on product category—for instance, explaining citric acid as a flavor enhancer in beverages *versus* a chelating agent in canned vegetables. Temperature parameter tuning (set to 0.3) balances creativity with factual consistency, preventing hallucination while maintaining natural language flow.

2.4 On-device large language model deployment

2.4.1 Model quantization and compression. Deploying Llama 3.2 3B on mobile devices requires utilizing a pre-quantized model optimized for edge deployment. We obtained the 4-bit quantized version of Llama-3.2-3B-Instruct from Hugging Face's model repository,[43] specifically compiled for Apple's MLX framework. This pre-quantized model employs group-wise quantization with a group size of 32 for weights and 8-bit per-token dynamic quantization for activations, maintaining separate scaling factors to preserve relative weight magnitudes within each group. The quantization achieves a 3.6x compression ratio, reducing the model from 6.4 GB in its original 32-bit floating-point format to 1.8 GB in 4-bit integer representation.

The quantized model weights are distributed in MLX's native format, optimized for memory-mapped loading on Apple Silicon devices. This format enables rapid initialization without loading the entire model into RAM simultaneously, leveraging the unified memory architecture of modern iOS devices. The model bundle includes metadata headers specifying quantization parameters, vocabulary mappings, and architectural configuration. Memory mapping reduces application launch time from 12 seconds to 3 seconds on iPhone 14+ devices while maintaining a peak memory footprint of 2.32 GB during inference operations. The bundled model path "Llama-3.2-3B-Instruct-4bit" is embedded directly in the application resources, eliminating the need for runtime downloads.

2.4.2 MLX framework integration. Apple's MLX framework provides hardware-accelerated ML operations specifically optimized for Apple Silicon's unified memory architecture. Our implementation leverages MLX's lazy evaluation graph construction, enabling efficient memory usage through automatic operation fusion and in-place tensor modifications. The inference pipeline implements custom MLX operations for 4-bit dequantization, performed just-in-time during matrix multiplications to minimize memory bandwidth requirements.

Performance profiling using Xcode Instruments revealed critical optimization opportunities in the attention mechanism. We implemented Flash Attention-inspired optimizations within MLX constraints, including chunked attention computation to maintain activation tensors within the Neural Engine's 20 MB scratchpad memory. The model operates with a maximum token generation limit of 1000 tokens per request. The implementation achieves 15.2% GPU utilization and 24% CPU utilization during inference, with the display consuming the remaining computational resources for UI updates.

2.5 Knowledge retrieval and embedding system

2.5.1 Embedding generation and indexing. The embedding system implements a hybrid approach combining character-level and semantic features optimized for food additive nomenclature. The embedding function generates 384-dimensional vectors through three complementary mechanisms: (1) character-based features capturing morphological patterns in chemical names, (2) position-weighted word embeddings emphasizing initial terms that often indicate primary substances, and (3) chemical pattern detection identifying common suffixes and functional groups. This multi-faceted approach addresses the challenge of matching varied nomenclatures, from systematic IUPAC names to common trade names and colloquial descriptions.

The embedding generation algorithm processes each additive name through text normalization (lowercasing and tokenization) before feature extraction. For character-level features, the algorithm iterates through each character position and applies trigonometric transformations (using sine functions) weighted by word position and character ASCII values to generate embedding activations. Word-level features

incorporate position weighting, where words appearing earlier in the additive name receive higher weights, reflecting the convention that primary substances typically appear first. Chemical pattern features activate specific embedding dimensions based on detection of common chemical suffixes including "acid", "ate", "ine", "ium", "ide", "oxy", "meth", "eth", and "prop", enabling the system to recognize chemical families and functional groups.

Vector normalization ensures consistent similarity metrics across the embedding space. Each embedding undergoes L2 normalization, projecting vectors onto the unit hypersphere to enable cosine similarity calculations using simple dot products. The normalized embeddings are serialized to JSON format with 16-bit float precision, reducing storage requirements to 18.08 MB for the complete SAFI while maintaining sufficient precision for similarity calculations. The embedding index loads into memory during application initialization, enabling sub-millisecond similarity searches without disk I/O operations.

2.5.2 Similarity search and retrieval optimization. The retrieval system implements a two-stage search process optimizing for both precision and recall in additive identification. The first stage performs exact string matching against substance names and alternative nomenclature. This stage handles straightforward cases including exact matches, case-insensitive variations, and parenthetical removal (*e.g.*, matching "citric acid" when searching for "citric acid (preservative)"). Color additives receive special handling through a comprehensive mapping table translating colloquial names to FDA nomenclature.

The second stage employs embedding-based similarity search for fuzzy matching when exact matches fail. The system computes cosine similarity between query embeddings and all database embeddings using vectorized operations accelerated by the Accelerate framework's vDSP functions. A dynamic thresholding mechanism adjusts the similarity cutoff based on query characteristics: shorter queries (under 10 characters) require higher similarity scores (%3e0.5) to prevent false positives, while longer chemical names accept lower thresholds (% 3e0.244) to accommodate nomenclature variations. The retrieval system also implements query expansion for common additive categories, automatically searching for related terms when queries match category patterns (*e.g.*, expanding "artificial colors" to include specific FD&C dyes).

2.6 User interface and interaction design

2.6.1 Camera-based ingredient scanning. The optical character recognition (OCR) system for ingredient label scanning utilizes Apple's Vision framework with VNRecognizeTextRequest configured for maximum accuracy. The implementation addresses specific challenges in food label typography, including small font sizes, curved surfaces, and variable lighting conditions. The recognition pipeline employs multiple preprocessing steps: automatic perspective correction for angled captures, adaptive histogram equalization for contrast enhancement, and text region detection to isolate ingredient lists from other label information.

Post-processing of OCR output addresses common recognition errors specific to food additives. The system implements a domain-specific spell correction algorithm using Levenshtein distance weighted by character confusion probabilities derived from empirical OCR error analysis. Common misrecognitions (*e.g.*, "1" *vs.* "l" and "0" *vs.* "O") receive special handling when occurring within known additive names. The correction algorithm maintains a confidence threshold, flagging uncertain recognitions for user verification rather than silently introducing errors. Ingredient list detection employs keyword spotting for markers like "INGREDIENTS:", "CONTAINS:", or "MADE WITH:", automatically extracting relevant text regions while filtering nutritional information and marketing claims.

2.6.2 Progressive disclosure interface. The user interface implements progressive disclosure principles to manage information complexity while maintaining accessibility for diverse user groups. The initial view presents three primary interaction modes: camera scanning for immediate analysis, random sampling for exploratory learning, and manual text entry for precise control.

The results presentation employs a three-tier information architecture accommodating different user expertise levels. The first tier displays identified additives with single-sentence purpose descriptions suitable for general consumers. The second tier, revealed through expandable sections, provides technical effects, alternative names, and regulatory classifications for users seeking deeper understanding. The third tier, accessed *via* CFR links, connects to federal regulations for professional users requiring legal documentation. This graduated approach prevents information overload while ensuring comprehensive access for specialized needs. Animation transitions (SwiftUI's withAnimation) provide visual continuity between states, with spring animations (response: 0.3, damping fraction: 0.7) creating responsive feedback for user interactions.

2.6.3 Feedback collection and iterative improvement. While FAL operates entirely on-device to ensure user privacy during real-time usage, we implemented comprehensive feedback mechanisms to support continuous improvement without compromising this privacy commitment. During the beta testing phase, we utilized Apple's TestFlight platform, which enabled users to directly report errors, technical difficulties, and usability concerns. This feedback proved invaluable for refining the application; for instance, users requested more detailed UI instructions, leading to an enhanced onboarding experience, and highlighted the need to remove technical RAG explanations in favor of consumer-friendly language. Most significantly, beta testers identified false positive cases where sample ingredient lists in prompts were mistakenly classified as user inputs due to the LLM's limited context window. We addressed this through prompt engineering refinements that explicitly separate examples from user data, substantially reducing false positive rates. Post-release, we leverage Apple's App Analytics platform to monitor app performance metrics (crash rates and launch times), user engagement patterns (feature usage and session duration), and device compatibility. Additionally, we maintain in-app anonymous feedback submission and monitor App Store reviews to identify emerging

issues or enhancement opportunities. This multi-channel feedback approach, combined with our modular architecture, enables rapid iteration cycles, demonstrated by our ability to implement and validate the prompt engineering improvements, without the complexities of server-side updates or data migration concerns inherent in cloud-based solutions.

2.7 Privacy-preserving architecture

2.7.1 On-device processing implementation. The privacy architecture ensures complete data isolation through exclusive on-device processing, with no network communication modules implemented for data transmission. All application components, including the quantized language model, SAFI and GBFPD, embedding indices, and classification models, reside within the application's sandboxed container, totaling 2.05 GB in the installed application. The architecture explicitly excludes networking frameworks beyond system-level APIs required for CFR documentation links, which open in external Safari instances without transmitting user data.

Privacy protection extends to the application's data persistence layer, which employs Core Data with SQLite backing stores that benefit from iOS's default device encryption when the device is locked. User interaction history, including scanned ingredients and generated explanations, remains confined to the device's encrypted storage partition. The application deliberately avoids using UserDefaults for sensitive data, with plans to leverage the Keychain Services API for any authentication tokens that may be required for future premium features. The application implements a privacy-first architecture with no third-party analytics or tracking frameworks integrated into the codebase.

2.7.2 Storage and memory management. The application implements aggressive memory management strategies to operate within iOS memory constraints while processing LLM. The memory management system employs a three-tier caching hierarchy: (1) active tensors in GPU memory for immediate computation, (2) model weights managed by MLX's memory system for efficient access, and (3) complete model weights memory-mapped from storage for on-demand loading. The MLX framework's lazy evaluation enables automatic memory pressure responses, deallocating intermediate tensors when system memory warnings occur.

Core data optimization for history storage implements batch faulting and relationship prefetching to minimize memory overhead when displaying analysis history. The fetch request configurations use optimized fetch requests to minimize memory overhead when displaying analysis history to load only displayed attributes, with full additive details loaded on-demand through relationship traversal. The OCR system for ingredient scanning processes images directly without persistent storage, converting captured images immediately to text through the Vision framework's VNRecognizeTextRequest. After text extraction, the original image data are released from memory, maintaining only the extracted ingredient text for analysis. This approach eliminates the need for image caching infrastructure while ensuring efficient memory utilization during the scanning process.

2.8 Code of Federal Regulations integration

The CFR integration system processes regulatory references from SAFI to generate direct links to electronic CFR documentation. The implementation parses the 20 CFR reference columns (Reg add01 through Reg add20) from SAFI, extracting valid regulation codes while filtering placeholder values and malformed entries. The parsing algorithm handles multiple reference formats including standard citations (*e.g.*, "184.1095") and special annotations for prohibited substances under 21 CFR 189. URL generation for CFR links follows the eCFR's REST API structure, constructing paths incorporating title (always 21 for food regulations), chapter (I), subchapter (B), part numbers, and section identifiers. The system implements fallback strategies for invalid section references, linking to part-level documentation when specific sections are unavailable. Special handling addresses regulation 170.3 (general food additives), which links to part 170 rather than a specific section due to its broad applicability across multiple additive categories. Once users tap the CFR links after each identified additive, an external Safari instance will lead to a detailed webpage.

2.9 Application-level optimizations

Application-level optimizations address the unique constraints of mobile deployment while maintaining a responsive user experience. The startup sequence implements lazy loading for all components except the primary view controller, deferring model initialization until first use. This approach reduces initial launch time to under 1 second while displaying the interface, with background loading completing within 3 seconds on iPhone 14+ devices. Subsequent launches leverage iOS's application suspension, maintaining model state in memory when possible to eliminate reload overhead.

The user interface employs predictive prefetching for likely user actions, pre-warming the OCR pipeline when the camera button becomes visible and pre-generating embeddings for the text field content during typing pauses. SwiftUI's task priority system ensures UI updates receive precedence over background processing, with inference operations using Task detached with priority "background" to prevent interface stuttering. Memory pressure responses implement graceful degradation, first clearing image caches, then unloading the classification model, and finally reducing LLM context length, ensuring core functionality remains available even on memory-constrained devices.

2.10 System validation and performance evaluation

2.10.1 Food category classifier validation. The food category classifier underwent comprehensive validation using a test set comprising 3136 randomly selected products from the GBFPD. The validation dataset was processed through an automated analysis pipeline that parsed model outputs including entry IDs, product names, brand information, ground truth categories, ingredient lists, and the classifier's top-3 predictions with confidence scores. The validation protocol assessed both top-1 and top-3 accuracy metrics, with top-3

accuracy serving as the primary evaluation criterion given the inherent subjectivity in food categorization boundaries. For each test sample, the system recorded the model's confidence scores and identified unknown tokens, ingredients not present in the DistilBERT-base vocabulary during training. Error analysis categorized misclassifications by ground truth category to identify systematic patterns in classifier performance. The system tracked confidence score distributions separately for correct and incorrect predictions to assess model calibration. Unknown token analysis quantified the frequency of out-of-vocabulary ingredients across the test set, providing insights into potential vocabulary limitations affecting classification performance. Confidence thresholds for triggering user confirmation in the deployed system were empirically determined through this validation process, balancing accuracy requirements with user experience considerations.

To benchmark the specialized DistilBERT classifier against general-purpose large language models, we evaluated GPT-4o (gpt-4o-2024-08-06) on the same 108 samples used for additive identification validation (Section 2.10.2). The GPT-4o API was queried with a system prompt defining it as a food science expert and a user prompt requesting top-3 category predictions based on ingredient lists, with responses structured in JSON format for consistent parsing. GPT-4o then self-evaluated its predictions against ground truth categories using a separate prompt that assessed top-1 and top-3 accuracy. This comparison provides context for the performance of task-specific models *versus* general-purpose AI agents in food categorization tasks.

2.10.2 Additive identification validation. Additive identification accuracy will be validated through manual review of 108 randomly sampled ingredient lists from the GBFPD. The validator will use SAFI as the reference standard to manually identify all food additives present in each ingredient list, distinguishing them from basic ingredients (flour, water, sugar, salt, oil, *etc.*). For each ingredient list, the validator will create a ground truth list of additives and compare it against the system's output. Three metrics will be calculated: (1) precision – the percentage of system-identified additives that are correctly identified as additives, (2) recall – the percentage of actual additives that the system successfully identifies, and (3) F1-score – the harmonic mean of precision and recall. The validator will categorize errors into two types: false positives (basic ingredients incorrectly identified as additives) and false negatives (missed additives). Alternative chemical names will be accepted as correct if they refer to the same substance according to FDA documentation, and naturally occurring compounds will only be considered additives if they are intentionally added for technological function rather than naturally present in the food matrix.

To benchmark the hybrid additive identification system against general-purpose large language models, we evaluated GPT-4o (gpt-4o-2024-08-06) on the same 108 samples and ground truth annotations. The GPT-4o API was queried with a system prompt ("Identify all food additives from an ingredient list") and a user prompt requesting identification of all food additives from each ingredient list with responses structured in JSON format containing an array of additive names. For each sample, GPT-4o's identified additives were compared against the manually annotated ground truth using the same evaluation criteria applied to FAL: precision (percentage of GPT-4o-identified additives that are correct), recall (percentage of true additives identified by GPT-4o), and F1-score. Alternative chemical names were accepted as correct matches according to FDA documentation. This comparison provides context for the performance of hybrid database-AI approaches *versus* pure general-purpose AI agents in additive identification tasks.

2.10.3 Explanation quality assessment. Explanation quality will be evaluated using a structured rubric applied by six independent evaluators: three professional evaluators with food science or nutritional science expertise and three general consumer representatives. Each evaluator will assess 108 randomly selected explanations (same as section 2.10.2) across three dimensions using a 5-point scale (1 = poor, 2 = fair, 3 = good, 4 = very good, and 5 = excellent). The evaluation criteria are: (1) factual accuracy – whether the explanation correctly describes the additive's function and properties according to FDA documentation, (2) clarity – whether a typical consumer without a scientific background can understand the explanation, and (3) contextual relevance – whether the explanation appropriately describes the additive's role in the specific food product rather than providing generic information. The nutritional science graduate student will focus primarily on factual accuracy, verifying explanations against FDA technical effect descriptions and identifying any scientific inaccuracies or misleading statements. The consumer evaluator will emphasize clarity and usefulness, assessing whether explanations use accessible language, avoid unnecessary technical jargon, and provide meaningful information for food purchasing decisions.

3 Results and discussion

3.1 System performance analysis

FAL was evaluated on two representative Apple devices to assess its computational efficiency and resource utilization: a MacBook Air with an M1 chip (16 GB unified memory, macOS 15.6.1) and an iPhone 14 (iOS 18.6.2). Performance metrics were captured using Xcode Instruments during typical usage scenarios including model initialization, additive identification, and explanation generation. Fig. 3 and Fig. 4 illustrate the Xcode Instrument analysis of the application on MacBook Air and iPhone 14, respectively.

3.1.1 Memory usage patterns. Memory consumption analysis revealed distinct operational phases with characteristic usage patterns across both platforms. On the MacBook Air, model initialization established a stable baseline of 1.82 GB (11.4% of total system memory) with brief peaks reaching 1.87 GB during loading operations. This efficient initialization profile demonstrates the effectiveness of memory-mapped model loading and the quantized model's reduced footprint.

During active processing phases involving additive identification and LLM generation, memory usage exhibited a characteristic dual-peak pattern. The first peak corresponded to additive identification operations, while the second, higher peak of 2.36

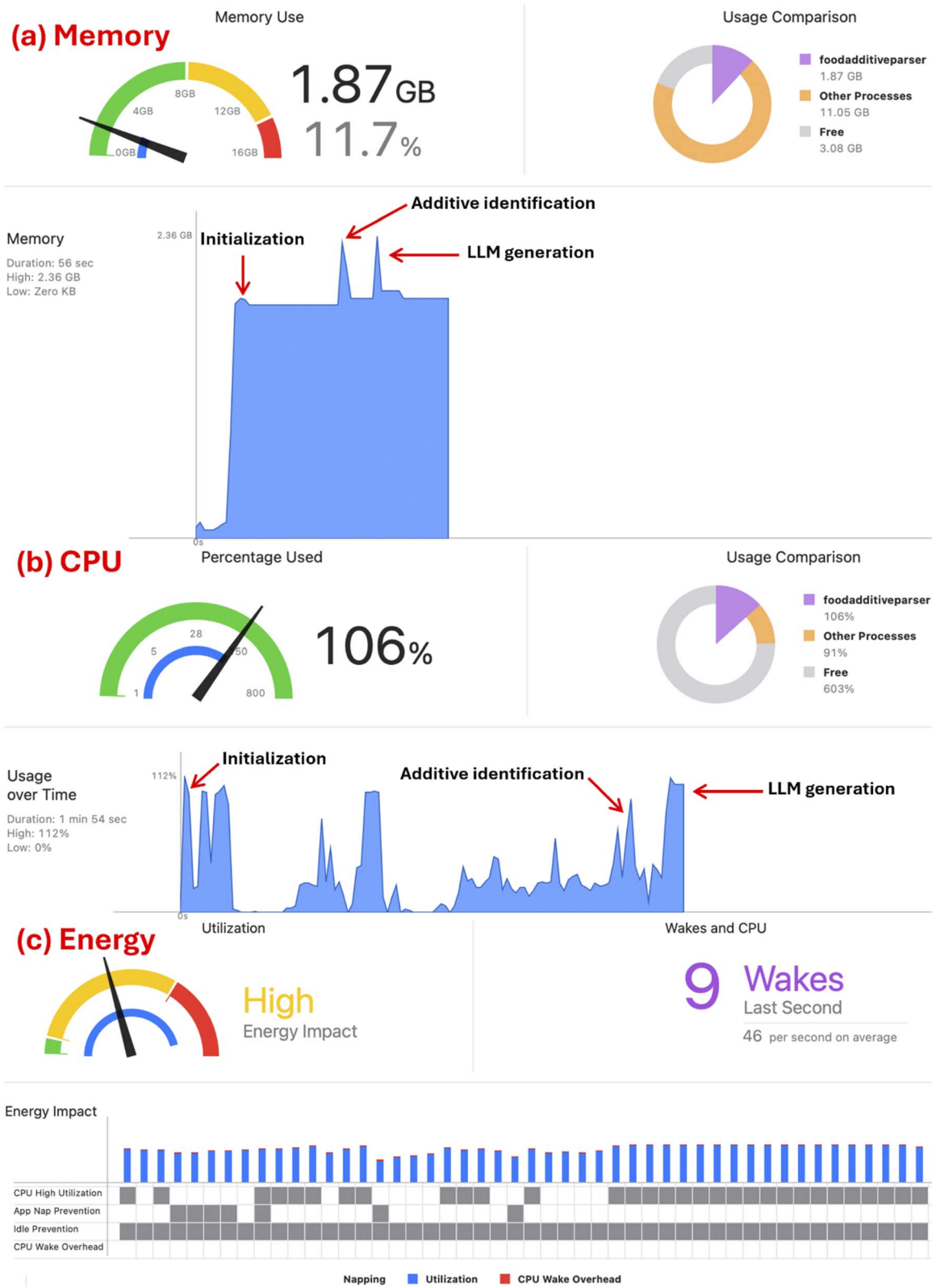

Fig. 3 Xcode Instrument analysis of the application on MacBook Air. (a) Memory usage patterns. (b) CPU utilization characteristics. (c) Energy impact.

GB occurred during explanation generation. This generation peak represents the maximum memory demand of the system, encompassing the loaded model, active computation graphs, and intermediate tensor storage. Following generation completion, memory usage stabilized at 1.87 GB, indicating successful cleanup of temporary computational structures.

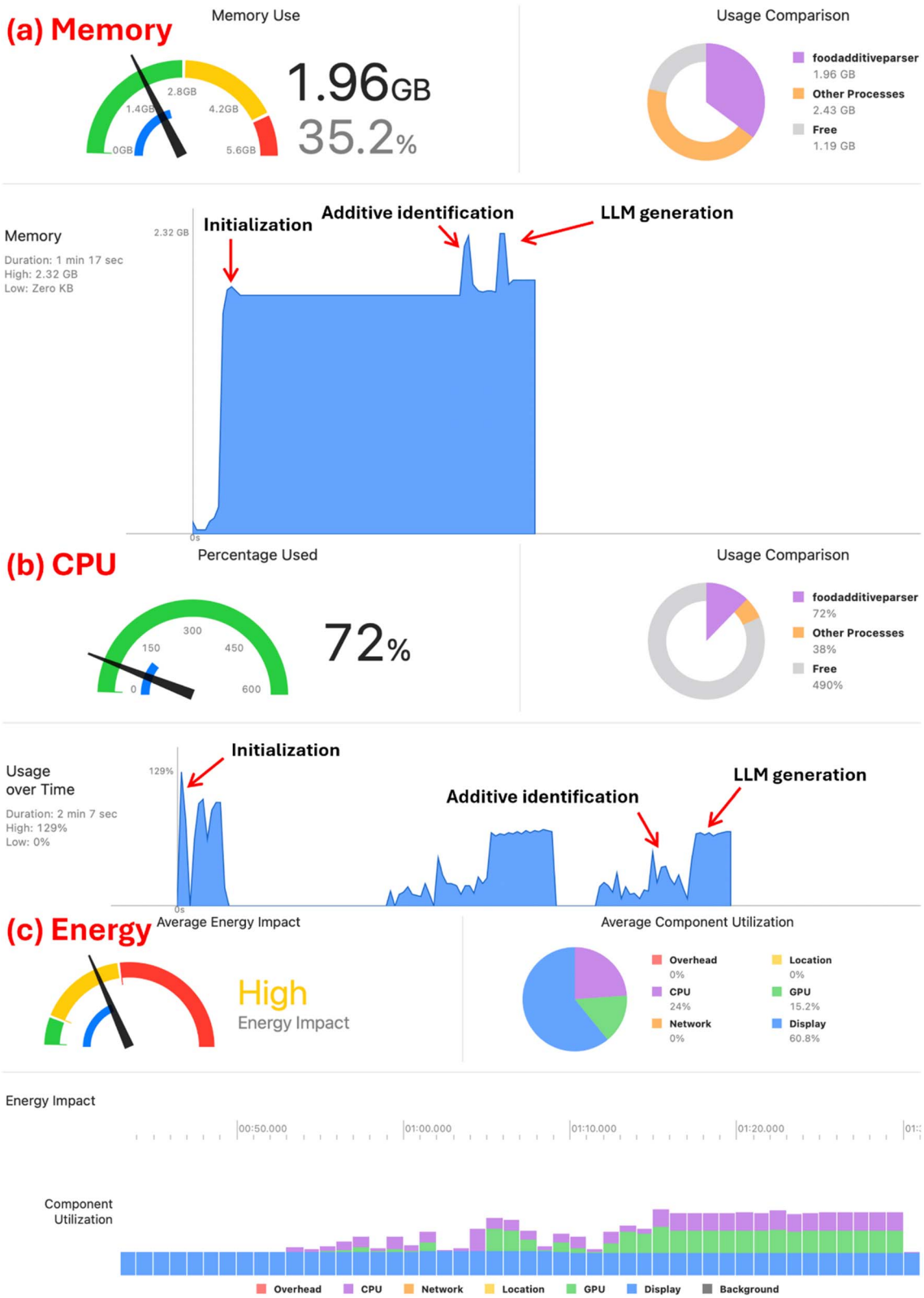

Fig. 4 Xcode Instrument analysis of the application on iPhone 14. (a) Memory usage patterns. (b) CPU utilization characteristics. (c) Energy impact.

iPhone 14 performance demonstrated similar patterns with platform-appropriate scaling. Model initialization required 1.84 GB of stable memory (33% of total system memory) with peaks reaching 1.92 GB. The higher percentage utilization on iPhone reflects the device's 6 GB total memory compared to the MacBook's 16 GB, yet the absolute memory requirements remained

remarkably consistent. During processing, the iPhone exhibited the same dual-peak pattern with a maximum of 2.32 GB during generation, stabilizing at 1.96 GB post-processing.

The consistency of absolute memory requirements across platforms validates the quantized model's efficiency and the MLX framework's optimization. The peak memory usage of approximately 2.3 GB across both devices demonstrates successful resource management within mobile hardware constraints while maintaining full functionality.

3.1.2 Latency analysis. Workflow latency was measured across 10 independent runs on iPhone 14 to quantify the execution time for each FAL component from app initialization through explanation generation. Table 1 presents the mean execution times with standard deviations for each workflow stage. The total workflow latency from app initialization to explanation generation initialization averaged 15.8 seconds, with model loading operations (7.3 seconds combined for the DistilBERT classifier and Llama 3.2 3B) representing the largest latency component during initial app launch. Subsequent app uses benefit from iOS application suspension, maintaining models in memory and eliminating reload overhead. OCR processing (0.932 seconds) and food category classification (0.112 seconds) demonstrated efficient execution, while additive identification (3.488 seconds) showed higher variability (±0.407 seconds), reflecting the complexity of hybrid database-AI matching across diverse ingredient lists.

3.1.3 CPU utilization characteristics. CPU utilization patterns revealed significant computational demands during LLM operations, with distinct differences between platforms. On the MacBook Air, with 800% total CPU capacity available (8-core configuration), typical LLM generation sustained 106% utilization with initialization peaks reaching 112%. This utilization pattern indicates effective multi-core scaling, with the quantized model successfully leveraging multiple processing cores for parallel computation.

iPhone 14 CPU usage showed more conservative patterns, with typical generation loads of 72% against 600% total available capacity (6-core configuration). Initialization peaks reached 129%, higher than generation loads, suggesting that model loading operations place greater instantaneous demands on CPU resources than sustained inference. The lower sustained utilization during generation reflects the iPhone's thermal management optimizations and the efficiency gains from the Neural Engine integration.

The utilization patterns demonstrate successful adaptation to hardware constraints while maintaining performance targets. The MacBook's higher sustained utilization leverages available thermal headroom and power resources, while the iPhone's more conservative approach balances performance with battery life and thermal constraints.

3.1.4 Energy consumption and component distribution. Energy consumption analysis classified both platforms as "High Energy Impact" during active LLM operations, reflecting the computational intensity of on-device language model processing. On the iPhone 14, detailed component analysis revealed energy distribution across CPU (24%), GPU (15.2%), and display operations (60.8%). The predominant display energy consumption corresponds to real-time UI updates during streaming text generation and the visual complexity of the progressive disclosure interface.

MacBook Air energy patterns showed 9 wake events in the last second with an average of 46 wake events per second during active generation. These wake patterns indicate intensive computational activity balanced with efficient scheduling, preventing unnecessary background processing while maintaining responsive user interaction.

The GPU utilization of 15.2% reflects a deliberate optimization choice in memory management. The GPU chunk size was constrained to 20 MB to balance peak memory usage against inference speed. Increasing GPU memory allocation would improve token generation rates but results in non-linear increases in peak memory consumption, potentially exceeding device capabilities during complex analyses.

To provide detailed quantitative characterization of energy consumption patterns, we conducted 60-second Power Profiler measurements in iOS developer mode on iPhone 14 ($n = 5$ runs) capturing complete FAL workflows from initialization through explanation generation completion. Fig. 5 illustrates a representative power consumption profile with the CPU profile over the measurement period, demonstrating the temporal dynamics of energy utilization across different workflow stages. Table 2 summarizes the component-specific energy metrics. The analysis revealed average power usage of 40.260 ± 5.141% per hour with display brightness maintained at 71.6 ± 2.5%. Apple's impact metrics, which range from 0–15 (low), 15–40 (medium), to >40 (high impact), showed a CPU impact of 6.160 ± 0.559, a GPU impact of 18.660 ± 2.248, and a display impact of 5.140 ± 0.371. Note that Apple intentionally reports energy consumption using relative impact metrics rather than absolute

Table 1 Latency analysis of FAL workflow components on iPhone 14 ($n = 10$ runs)

Component	Mean (seconds)	Std dev (seconds)
Additive embeddings loading	0.853	0.025
CFR data loading	0.238	0.008
Food category classifier model loading	2.399	0.123
Llama-3.2-3B-Instruct-4bit model loading	4.893	0.135
OCR processing	0.932	0.027
Food category classification	0.112	0.046
Additive identification	3.488	0.407
Explanation generation initialization	2.847	0.106

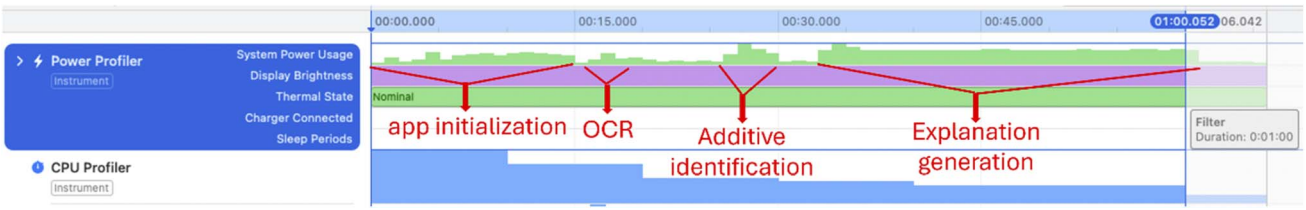

Fig. 5 Representative 60-second power monitoring and CPU usage profiles from app initialization through explanation generation completion.

units (*e.g.*, watts or milliampere-hours) because power consumption varies significantly across device models, thermal conditions, and usage contexts, making standardized comparisons challenging. The GPU impact of 18.660 positions FAL within the medium range despite the computational demands of on-device LLM inference, while the CPU impact of 6.160 indicates efficient computational resource utilization. The display impact of 5.140 corresponds to progressive disclosure interface updates during streaming text generation. Critically, networking impact remained at 0 across all measurements, validating the complete offline operation of the privacy-preserving architecture. The power usage rate of 40.3% per hour indicates approximately 2.5 hours of sustained continuous operation, though typical real-world usage involves brief, intermittent queries rather than continuous processing.

3.1.5 Cross-platform performance implications. Comparative analysis between platforms reveals both the capabilities and constraints of on-device LLM deployment across the Apple ecosystem. The MacBook Air's superior memory capacity (16 GB *vs.* 6 GB) enables more aggressive memory utilization strategies, while both platforms achieve similar absolute memory requirements, validating the universal applicability of the quantized model approach.

The iPhone's higher memory utilization percentage (33% *vs.* 11.4% during initialization) demonstrates successful optimization for resource-constrained environments. Despite operating closer to hardware limits, the iPhone maintains full functionality without performance degradation, indicating robust memory management and efficient model quantization.

CPU utilization differences reflect platform-specific optimization strategies. The MacBook's higher utilization leverages available thermal and power headroom for maximum performance, while the iPhone's conservative approach prioritizes sustained operation and battery efficiency. Both approaches achieve target token generation rates of 13–30 tokens per second, ensuring consistent user experience across devices.

Table 2 Energy consumption analysis of FAL on iPhone 14 ($n = 5$ runs)

Metric	Mean	Std dev
Power usage (% per hour)	40.260	5.141
Display brightness (%)	71.6	2.5
CPU impact	6.160	0.559
GPU impact	18.660	2.248
Display impact	5.140	0.371
Networking impact	0	0

The energy analysis reveals the fundamental trade-offs of on-device AI processing. While classified as high energy impact, this approach eliminates network dependencies, ensures complete privacy, and provides instantaneous responses. The display-dominated energy consumption suggests that interface optimizations could significantly improve overall efficiency without compromising core AI functionality.

These performance characteristics validate the technical feasibility of sophisticated on-device AI for consumer applications while highlighting the platform-specific optimizations necessary for effective deployment across diverse hardware configurations.

3.2 Food category classification performance

3.2.1 Training performance and convergence. The DistilBERT-base food category classifier was trained on 90% (943K) of samples across 257 food categories using an A100 GPU on Google Colab. Training completed in 74.9 minutes over 5583 steps across 3 epochs, achieving a throughput of 621 samples per second and 1.243 training steps per second. The model demonstrated consistent convergence throughout training. Training loss decreased from 1.427 (step 500) to 0.341 (step 5500), while validation loss declined from 1.179 (step 500) to 0.339 (step 5500). Fig. 6 presents the training and validation curves. The parallel reduction in both losses without significant divergence indicates effective learning without overfitting. The validation loss consistently remained below or comparable to the training loss after step 1000, suggesting appropriate regularization through the implemented weight decay (0.01) and mixed precision training. Evaluation on the held-out test set of 10% (104K) samples yielded a top-1 accuracy of 90.8%, demonstrating strong classification performance across the diverse food categories.

3.2.2 On-device validation results and error analysis. The food category classifier achieved an on-device top-1 accuracy of 69.1% (2166/3136) and top-3 accuracy of 87.2% (2735/3136) on the validation dataset. The 18.1 percentage point improvement from top-1 to top-3 accuracy demonstrates the value of presenting multiple category options to users, particularly given the subjective nature of food categorization boundaries where multiple categories may legitimately apply to a single product.

Confidence calibration analysis revealed well-calibrated model behavior with clear separation between correct and incorrect (top-1) predictions in Fig. 7. Correct predictions exhibited an average confidence of 0.852, while incorrect predictions averaged 0.549, indicating the model's ability to

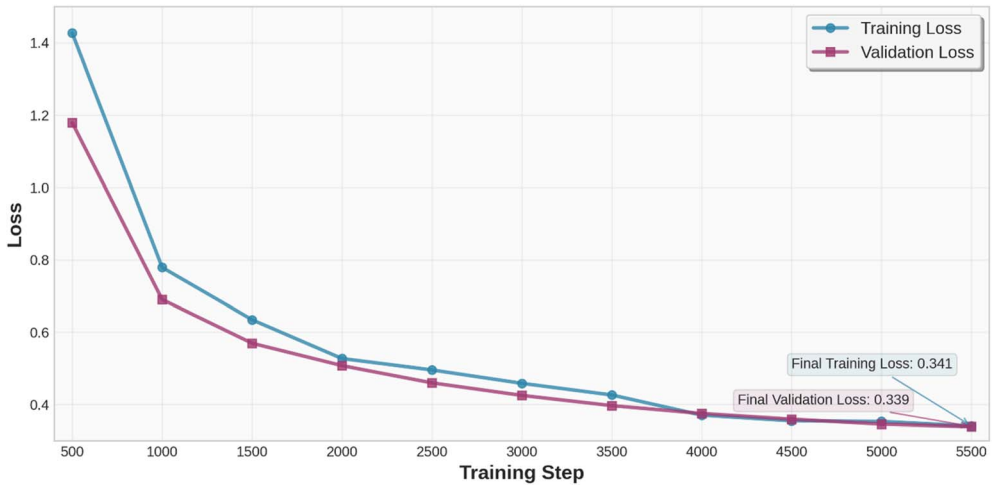

Fig. 6 Training and validation loss curves for the food category classifier.

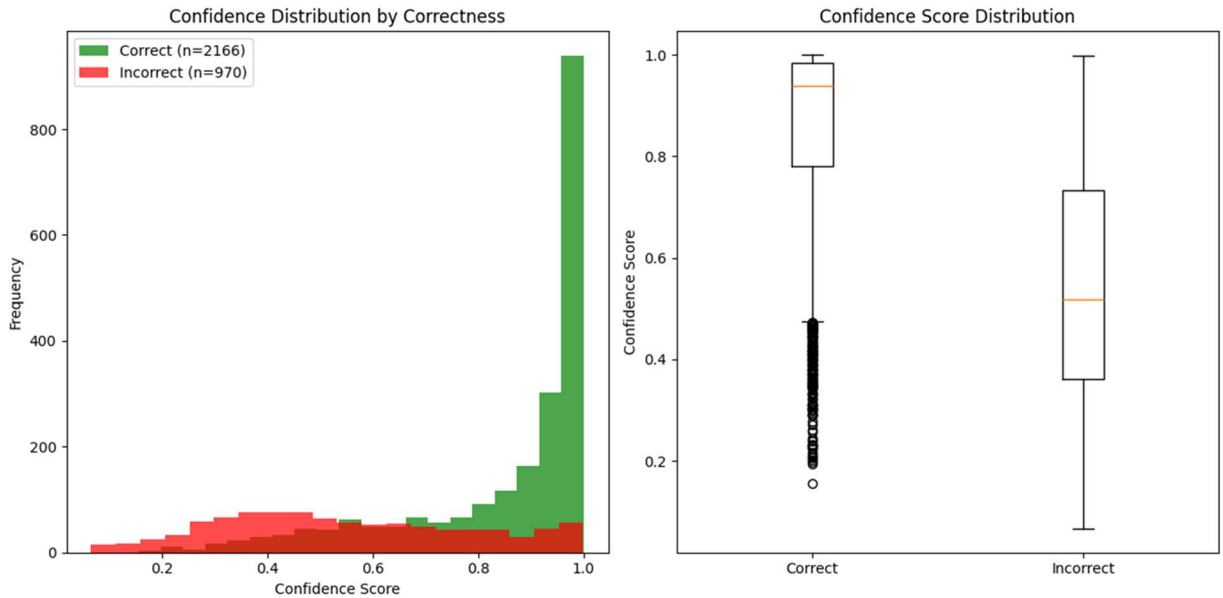

Fig. 7 The food category classifier on-device confidence calibration analysis.

distinguish between confident correct classifications and uncertain cases. The confidence distribution analysis showed most correct predictions clustered at high confidence scores (0.8–1.0), with a small number of low-confidence correct predictions appearing as outliers. This distribution pattern supports the implementation of confidence-based user confirmation thresholds in the deployed system.

Category-specific performance identified systematic challenges in certain food categories. Detailed accuracy-by-category performance can be found in SI Fig. S1. The highest error rates occurred in "Soda" (104 errors), "Popcorn, Peanuts, Seeds & Related Snacks" (35 errors), and "Yogurt" (34 errors). The high error rate for soda products likely reflects ingredient list similarities with other beverage categories, while snack food errors suggest overlapping ingredient profiles across snack subcategories. Yogurt classification challenges may stem from the diverse range of yogurt-based products that blur category boundaries with desserts and beverages.

Unknown token analysis revealed vocabulary limitations affecting model performance. The validation set contained substantial numbers of out-of-vocabulary ingredients, with most samples (>85%) containing ≥5 unknown tokens. The most frequent unknown tokens included nutritional additives (niacin: 584 occurrences, mononitrate: 471 occurrences, riboflavin: 358 occurrences) and common food processing ingredients (starch: 454 occurrences, citric: 439 occurrences, lecithin: 269 occurrences). This pattern indicates that the DistilBERT vocabulary, trained primarily on general text corpora, lacks comprehensive coverage of food industry terminology, particularly nutritional supplements and specialized food additives.

The validation results demonstrate acceptable performance for consumer-facing food categorization while highlighting

areas for potential improvement. The 87.2% top-3 accuracy provides sufficient reliability for the application's use case, where users can select from multiple category suggestions. The clear confidence calibration enables effective implementation of uncertainty-based user confirmation, enhancing system reliability in ambiguous cases.

3.2.3 Comparison with GPT-4o. Comparative evaluation of GPT-4o on the 108-sample validation set yielded 52.78% top-1 accuracy (57/108) and 70.37% top-3 accuracy (76/108). These results are substantially lower than the DistilBERT classifier's performance of 69.1% top-1 and 87.2% top-3 accuracy on the full validation dataset. The performance gap demonstrates that task-specific fine-tuning on food product data provides significant advantages over general-purpose language models for food categorization, despite GPT-4o being orders of magnitude larger than the deployed DistilBERT model (134 MB). The specialized classifier's superior performance, combined with its compact size enabling on-device deployment, validates the architectural choice of domain-specific model fine-tuning over reliance on general-purpose AI agents for this application.

3.3 Additive identification validation results

The additive identification system was evaluated on 108 randomly selected ingredient lists from the GBFPD containing 833 manually annotated food additives. The system achieved an overall precision of 0.728, a recall of 0.788, and an F1-score of 0.757, correctly identifying 652 additives while generating 244 false positives and missing 175 true additives. Per-entry performance showed considerable variability, with a mean precision of 0.744 ± 0.250, a recall of 0.808 ± 0.209, and an F1-score of 0.749 ± 0.201, as illustrated in the distribution of metrics across entries (Fig. 8). The system demonstrated a tendency toward over-identification, predicting 900 additives compared to the 833 ground truth annotations. Error analysis revealed that the most frequent false positives were legitimate additives, including sodium benzoate (33 occurrences), citric acid (15), and xanthan gum (13), suggesting potential inconsistencies in ground truth annotation criteria. False negatives were dominated by nutritional supplements and alternative nomenclature, with iron (15 occurrences), folic acid (9), and baking soda (8) being most commonly missed. The higher recall compared to precision indicates the system successfully captures most additives present while generating additional candidates that may require verification, representing acceptable performance for consumer-facing applications where missing additives could be more problematic than providing comprehensive information.

Comparative evaluation of GPT-4o on the same 108 samples using structured prompts for additive identification yielded overall precision of 0.841, recall of 0.762, and F1-score of 0.800, with per-entry means of 0.838 ± 0.189 (precision), 0.787 ± 0.206 (recall), and 0.793 ± 0.173 (F1-score). GPT-4o correctly identified 635 additives with 120 false positives and 198 false negatives, compared to FAL's hybrid system identifying 652 correct additives with 244 false positives and 175 false negatives. While GPT-4o demonstrated higher precision (0.841 *vs.* 0.728), FAL achieved slightly higher recall (0.788 *vs.* 0.762), resulting in comparable F1-scores (0.757 *vs.* 0.800). The performance differences reflect trade-offs in precision-recall balance, with GPT-4o showing more conservative identification (fewer false positives) while FAL's hybrid approach captures more true additives at the cost of additional false positives. Despite GPT-4o being orders of magnitude larger and requiring cloud-based processing, the comparable performance validates FAL's hybrid database-AI architecture for on-device deployment in consumer applications.

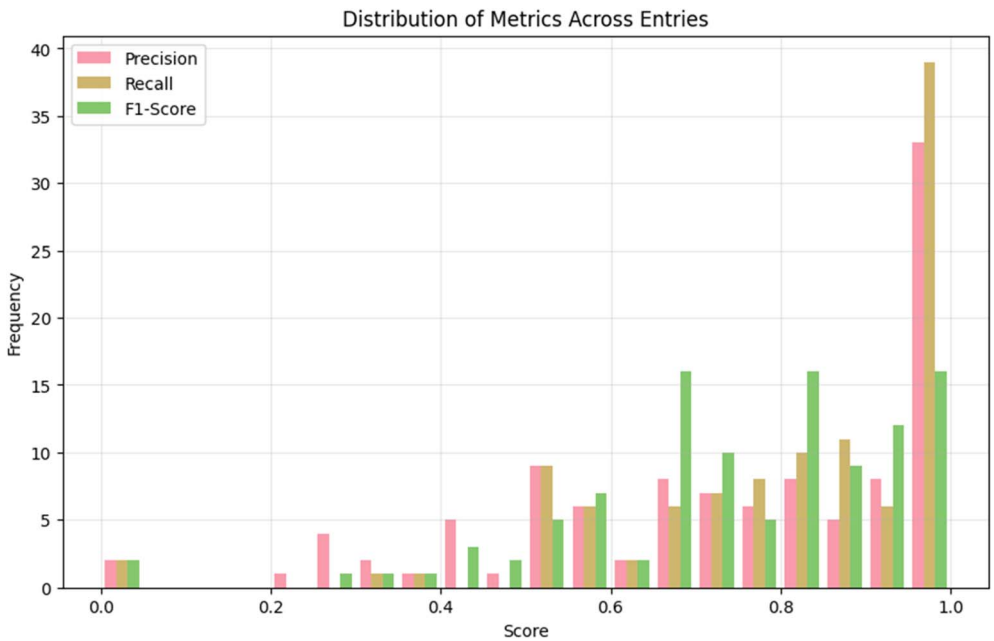

Fig. 8 Distribution of metrics of additive identification across 108 random entries from the GBFPD.

3.4 Explanation quality assessment results

The explanation quality assessment was conducted by six independent evaluators across 108 randomly selected explanations using a 5-point rubric (1 = poor to 5 = excellent) with results shown in Fig. 9. Consumer evaluators ($n = 3$) provided ratings of 4.145 ± 1.091 for factual accuracy, 4.299 ± 0.973 for clarity, and 4.235 ± 0.935 for contextual relevance. Professional evaluators ($n = 3$) assigned ratings of 3.728 ± 0.904 for factual accuracy, 3.877 ± 0.831 for clarity, and 4.302 ± 0.772 for contextual relevance. Both evaluator groups rated contextual relevance highest, indicating successful product-specific tailoring rather than generic additive descriptions. Professional evaluators showed greater rating consistency (standard deviations of 0.772–0.904) compared to consumer evaluators (0.935–1.091), reflecting more homogeneous expert assessment criteria *versus* varied consumer perspectives. The expanded evaluator pool demonstrates acceptable explanation quality for consumer-facing food additive education, with all metrics averaging above 3.7/5.0, balancing scientific precision with accessibility requirements.

3.5 Case study

A case study of FAL on identifying and explaining additives for packaged biscuits is illustrated in Fig. 10 and SI demo video.mp4. The app employed the OCR technique to automatically extract the ingredient list text in the image as below: "whole grain blend (rolled oats, rye flakes), enriched flour (wheat flour, niacin, reduced iron, thiamin mononitrate [VITAMIN B1], riboflavin [VITAMIN B2], folic acid), sugar, canola oil, whole grain wheat flour, malt syrup (from corn and barley), baking soda, salt, soy lecithin, disodium pyrophosphate, datem, ferric orthophosphate (iron), niacinamide, pyridoxine hydrochloride (VITAMIN B6), riboflavin (VITAMIN B2), thiamin mononitrate (VITAMIN B1)".

Then, food category classifier predictions included "Processed Cereal Products", "Cookies & Biscuits", and "Biscuits/Cookies" with confidence scores of 21.8%, 20.9%, and 16.6%, respectively. After the user selected the "Cookies & Biscuits" category, the app identified additives from the ingredient list including canola oil, DATEM, disodium pyrophosphate, ferric orthophosphate, folic acid, iron, niacin, niacinamide, pyridoxine hydrochloride, riboflavin, soy lecithin, vitamin b1, vitamin b2. For each identified additive, the app showed the matched additive name in SAFI, the "used for" technical effects, other names of the additive, and the CFR link for the additive. For example, disodium pyrophosphate was matched to sodium acid pyrophosphate in SAFI, and it has a range of technical effects, *e.g.* leavening agent for cookies. Disodium pyrophosphate has other names like diphosphoric acid. The CFR for disodium pyrophosphate is 21 CFR 182.1087.

The LLM agent grouped the additives with similar technical effects to provide clear and user-friendly explanations as below: "Let's break down the additives in this cookie recipe. I'll explain what each one does in simple terms. Emulsifiers: (1) CANOLA OIL: acts as an emulsifier, helping mix oil and water-based ingredients together and making the dough easier to work with. (2) SOY LECITHIN: also helps mix oil and water-based ingredients, creating a smooth dough. Stabilizers and thickeners: (1) CANOLA OIL: besides being an emulsifier, it also helps stabilize the dough and keep it moist. (2) BAKING SODA: helps release carbon dioxide gas, causing the dough to rise and giving the cookies their light texture. Coloring agents: (1) RIBOFLAVIN (VITAMIN B2): acts as a coloring agent, giving the cookies a yellow color. (2) DISODIUM PYROPHOSPHATE: can also act as a coloring agent, although its primary function is as a leavening agent. Flavor enhancers: (1) NIACIN: provides a flavor and nutritional boost to the cookies. (2) NICOTINAMIDE:

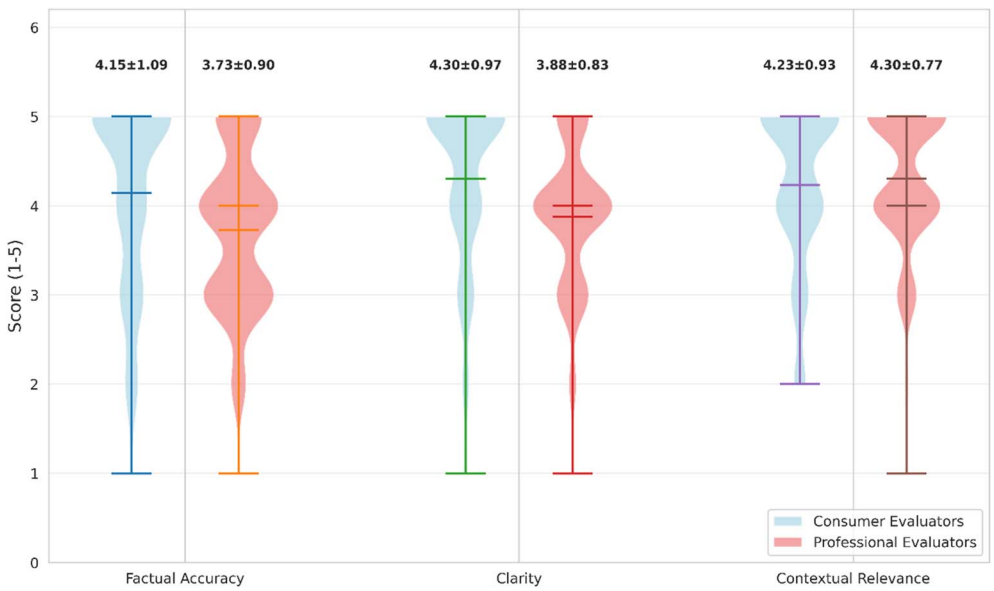

Fig. 9 Explanation quality assessment results from consumer ($n = 3$) and professional ($n = 3$) evaluators. Violin plots show the distribution of ratings across 108 explanations for three evaluation criteria: factual accuracy, clarity, and contextual relevance. Values above each plot indicate mean ± standard deviation.

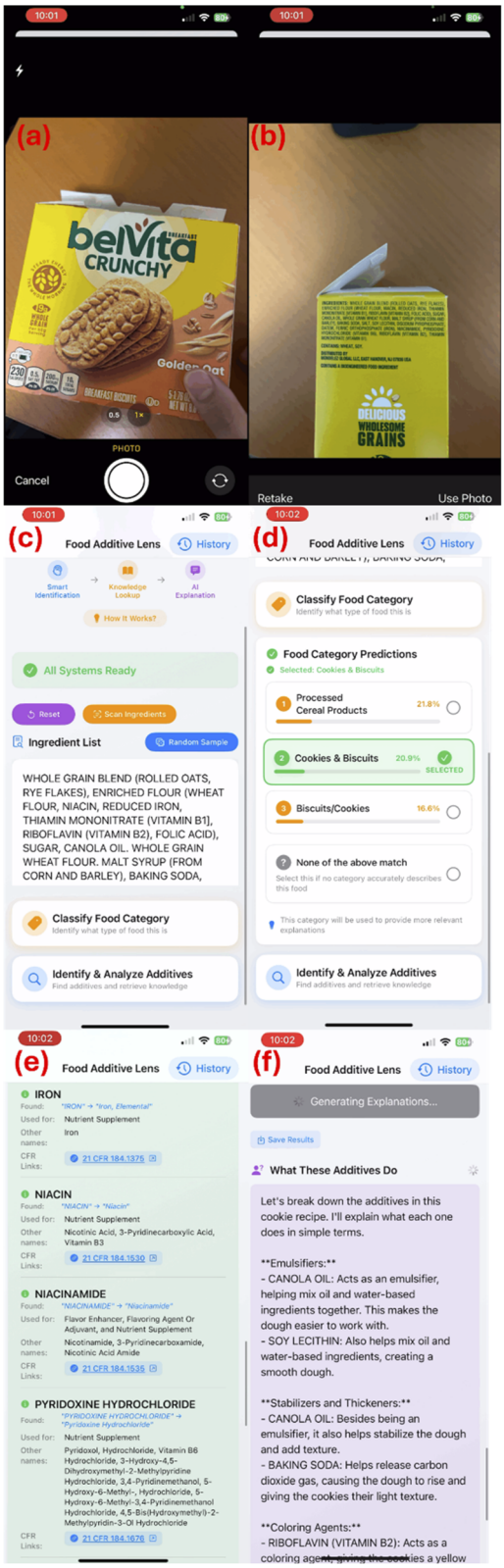

Fig. 10 FAL screenshots for analyzing the ingredient list of packaged biscuits. (a) Package cover. (b) Package ingredient list. (c) OCR-scanned ingredient list text. (d) Food category prediction. (e) Identified additives. (f) Explanation generation.

enhances flavor and serves as a nutrient supplement. (3) PYRIDOXINE HYDROCHLORIDE: acts as a flavor enhancer and nutrient supplement. (4) VITAMIN B1: contributes to flavor and serves as a nutrient supplement. (5) RIBOFLAVIN: also serves as a flavor enhancer and nutrient supplement. Nutrient supplements: (1) FERRIC ORTHOPHOSPHATE: provides iron, an essential nutrient. (2) FOLIC ACID: acts as a nutrient supplement, providing folate. (3) IRON: serves as a nutrient supplement, providing iron. (4) NIACIN: provides a nutrient supplement, niacin. (5) PYRIDOXINE HYDROCHLORIDE: acts as a nutrient supplement, providing vitamin B6. (6) RIBOFLAVIN: provides a nutrient supplement, riboflavin. These additives work together to create a delicious and nutritious cookie".

The case study demonstrates successful end-to-end performance across all three AI agents. The OCR system accurately extracted the complete ingredient list from the biscuit packaging, the food category classifier correctly identified "Cookies & Biscuits" among its top-3 predictions (20.9% confidence), and the additive identification system successfully identified 13 additives with proper SAFI matching and regulatory links. The explanation generator effectively organized these additives into functional categories (emulsifiers, stabilizers, coloring agents, flavor enhancers, and nutrient supplements) with consumer-friendly explanations, validating the system's ability to transform complex ingredient information into accessible, science-based consumer education.

4 Conclusions

This work demonstrates the successful deployment of sophisticated on-device AI for food additive education through FAL, addressing the critical disconnect between scientific knowledge and consumer understanding. The application's three-agent architecture achieved robust performance metrics: 87.2% top-3 accuracy in food categorization, a 0.757 F1-score in additive identification, and consumer-acceptable explanation quality scores averaging 3.5/5.0.

By utilizing a pre-quantized 4-bit version of Llama 3.2 3B, the system achieves a generation speed of 13–30 tokens/second on consumer smartphones, validating the practical deployment of large language models without compromising functionality or privacy.

The integration of authoritative databases (FDA's SAFI and USDA's GBFPD) through retrieval-augmented generation establishes a paradigm for grounding AI responses in regulatory sources, critical for health-related applications where misinformation poses genuine risks. The system's complete offline operation eliminates privacy concerns inherent in cloud-based solutions, addressing a primary barrier to digital health tool adoption. Performance consistency across devices (iPhone 14 and MacBook Air M1) with a peak memory usage of 2.36 GB demonstrates practical deployment feasibility across the iOS ecosystem.

Several limitations warrant acknowledgment. The DistilBERT vocabulary's limited coverage of food-specific terminology (>85% of samples containing ≥5 unknown tokens)

suggests potential improvements through domain-specific pretraining. The additive identification system's tendency toward over-identification (precision: 0.728) may require refinement for professional applications. Explanation quality assessment, while encouraging, relied on limited evaluators and would benefit from larger-scale user studies.

Future work should explore expansion to nutrition claims analysis, integration with electronic health records for personalized dietary guidance, and adaptation for international regulatory frameworks. The modular architecture facilitates extension to other food science domains including allergen detection, sustainability metrics, and nutritional optimization. The demonstrated success of on-device AI for complex scientific communication suggests broader applications in combating misinformation across health and science domains while preserving user privacy and autonomy.

Author contributions

Yihang Feng: writing – review & editing, writing – original draft, validation, software, resources, methodology, investigation, formal analysis, data curation, and conceptualization. Yi Wang: writing – review & editing, methodology, investigation, and data curation. Xinhao Wang: writing – review & editing and methodology. Bo Zhao: writing – review & editing and methodology. Jinbo Bi: writing – review & editing and conceptualization. Song Han: writing – review & editing, supervision, and conceptualization. Yangchao Luo: writing – review & editing, supervision, funding acquisition, and conceptualization.

Conflicts of interest

There are no conflicts to declare. The Food Additive Lens app is free to download from App Store.

Data availability

The up-to-date GBFPD is available at https://fdc.nal.usda.gov/download-datasets. The up-to-date SAFI is available at https://www.hfpappexternal.fda.gov/scripts/fdcc/index.cfm?set=FoodSubstances. All codes are available *via* GitHub with Apache License 2.0 at https://github.com/yih-f/Food-Additive-Lens and its linked Zenodo repository at https://doi.org/10.5281/zenodo.18193643, including the application Swift script, food category classifier training and on-device evaluation scripts, food additive embedding and identification validation scripts, explanation quality assessment script, and prompts for LLM.

Supplementary information: high-resolution food category classification accuracy results and the demo video of food additive lens are provided. See DOI: https://doi.org/10.1039/d5dd00444f.

Acknowledgements

We gratefully acknowledge the Institute for the Advancement of Food and Nutrition Sciences (IAFNS) for providing the 2025 Summer Research Fellowship funding that supported this research. We extend special thanks to Trish Zecca and Caitlin Karolenko from IAFNS for their participation in beta testing and valuable feedback that significantly improved the application's functionality and user experience.

References

1 Y. Kwon, R. López-García, S. Socolovsky and B. Magnuson, Global regulations for the use of food additives and processing aids, *InPresent Knowledge in Food Safety*, Academic Press, 2023, pp. 170–193.
2 GiGAFACT, Does the US allow 10,000 additives into food? [Internet], 2024, [cited 2025 Aug 31]. Available from: https://gigafact.org/fact-briefs/does-the-us-allow-10000-additives-into-food/.
3 U.S. Food and Drug Administration, Substances added to food (formerly EAFUS) [Internet], 2025, [cited 2025 Aug 31]. Available from: https://www.hfpappexternal.fda.gov/scripts/fdcc/index.cfm?set=FoodSubstances.
4 P. Pressman, R. Clemens, W. Hayes and C. Reddy, Food additive safety: A review of toxicologic and regulatory issues, *Toxicol. Res. Appl.*, 2017, **1**, DOI: 10.1177/2397847317723572.
5 International Food Information Council, 2024, *IFIC food & health survey*, [Internet]. 2024 [cited 2025 Aug 31]. Available from: https://ific.org/wp-content/uploads/2025/07/2024-IFIC-Food-Health-Survey.pdf.
6 R. G. Xiong, J. Li and H. B. Li, Connections Between Food Additives and Psychiatric Disorders, *Psychiatr. Times*, 2024, **41**(4).
7 R. Diyab, J. Grgurevic and R. Roy, Exploring nutrition misinformation on social media platforms, *Proc. Nutr. Soc.*, 2025, **84**, E8.
8 T. Lu, Z. Mo, F. He, Y. Wang, Z. Yu, L. Li and P. Wall, Unlocking the potential of social media on food additives for effective science communication, *npj Sci. Food*, 2024, **8**(1), 100.
9 A. Moreira da Silva and M. J. Barroca, Addressing Chemophobia: Bridging Misconceptions in Food Chemistry, *Appl. Sci.*, 2025, **15**(11), 6104.
10 J. Aschemann-Witzel, P. Varela and A. O. Peschel, Consumers' categorization of food ingredients: Do consumers perceive them as 'clean label'producers expect? An exploration with projective mapping, *Food Qual. Prefer.*, 2019, **71**, 117–128.
11 International Food Information Council, From "chemical-sounding" to "clean": consumer perspectives on food ingredients [Internet], 2021, [cited 2025 Aug 31]. Available from: https://ific.org/wp-content/uploads/2025/04/Food-Ingredients-LSI-Survey.May-2021.pdf.
12 G. Ares, A. N. Giménez, F. Bruzzone, L. Vidal, L. Antúnez and A. Maiche, Consumer visual processing of food labels: results from an eye-tracking study, *J. Sens. Stud.*, 2013, **28**(2), 138–153.
13 L. Machín, L. Antúnez, M. R. Curutchet and G. Ares, The heuristics that guide healthiness perception of ultra-

processed foods: a qualitative exploration, *Public Health Nutr.*, 2020, **23**(16), 2932–2940.
14 C. O. Werle, C. Gauthier, A. P. Yamim and F. Bally, How a food scanner app influences healthy food choice, *Appetite*, 2024, **200**, 107571.
15 S. Gioia, I. M. Vlasac, D. Babazadeh, N. L. Fryou, E. Do, J. Love, R. Robbins, H. S. Dashti and J. M. Lane, Mobile apps for dietary and food timing assessment: evaluation for use in clinical research, *JMIR Form. Res.*, 2023, **7**, e35858.
16 G. Tangari, M. Ikram, K. Ijaz, M. A. Kaafar and S. Berkovsky, Mobile health and privacy: cross sectional study, *BMJ*, 2021, **373**, n1248.
17 Q. Grundy, K. Chiu, F. Held, A. Continella, L. Bero and R. Holz, Data sharing practices of medicines related apps and the mobile ecosystem: traffic, content, and network analysis, *BMJ*, 2019, **364**, l920.
18 N. Zadushlivy, R. Biviji and K. S. Williams, Exploration of reproductive health apps' data privacy policies and the risks posed to users: qualitative content analysis, *J. Med. Internet Res.*, 2025, **27**, e51517.
19 A. Oliveira, A. L. Amaro and M. Pintado, Impact of food matrix components on nutritional and functional properties of fruit-based products, *Curr. Opin. Food Sci.*, 2018, **22**, 153–159.
20 E. Chazelas, M. Deschasaux, B. Srour, E. Kesse-Guyot, C. Julia, B. Alles, N. Druesne-Pecollo, P. Galan, S. Hercberg, P. Latino-Martel and Y. Esseddik, Food additives: distribution and co-occurrence in 126,000 food products of the French market, *Sci. Rep.*, 2020, **10**(1), 3980.
21 L. Wu, C. Zhang, Y. Long, Q. Chen, W. Zhang and G. Liu, Food additives: From functions to analytical methods, *Crit. Rev. Food Sci. Nutr.*, 2022, **62**(30), 8497–8517.
22 J. M. Aguilera, Food matrices as delivery units of nutrients in processed foods, *J. Food Sci.*, 2025, **90**(2), e70049.
23 Y. Qiao, X. Li, D. Wiechmann and E. Kerz. (Psycho-) Linguistic Features Meet Transformer Models for Improved Explainable and Controllable Text Simplification. *arXiv*, 2022, preprint, arXiv:2212.09848, DOI: 10.48550/arXiv.2212.09848.
24 Z. Li, S. Belkadi, N. Micheletti, L. Han, M. Shardlow and G. Nenadic, Large Language Models for Biomedical Text Simplification: Promising But Not There Yet, *arXiv*, 2024, preprint, arXiv:2408.03871, DOI: 10.48550/arXiv.2408.03871.
25 L. Liu, H. An, P. Chen and L. Ye, A Contemporary Overview: Trends and Applications of Large Language Models on Mobile Devices, *arXiv*, 2024, preprint, arXiv:2412.03772, DOI: 10.48550/arXiv.2412.03772.
26 Y. Wang, J. Wang, W. Zhang, Y. Zhan, S. Guo, Q. Zheng and X. Wang, A survey on deploying mobile deep learning applications: A systemic and technical perspective, *Digital Communications and Networks*, 2022, **8**(1), 1–7.
27 S. Han, H. Mao and W. J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, *arXiv*, 2015, preprint, arXiv:1510.00149, DOI: 10.48550/arXiv.1510.00149.
28 X. Wang and W. Jia, Optimizing edge AI: a comprehensive survey on data, model, and system strategies, *arXiv*, 2025, preprint, arXiv:2501.03265, DOI: 10.48550/arXiv.2501.03265.
29 P. V. Dantas, W. S. Da Silva, L. C. Cordeiro and C. B. Carvalho, A comprehensive review of model compression techniques in machine learning, *Appl. Intell.*, 2024, **54**(22), 11804–11844.
30 Apple Open Source, Apple project MLX [Internet], 2025, [cited 2025 Aug 31]. Available from: https://opensource.apple.com/projects/mlx/.
31 Google AI Edge, MediaPipe solutions guide [Internet], 2025, [cited 2025 Aug 31]. Available from: https://ai.google.dev/edge/mediapipe/solutions/guide.
32 Z. Li, Z. Wang, W. Wang, K. Hung, H. Xie and F. L. Wang, Retrieval-augmented generation for educational application: A systematic survey, *Comput. Educ. Artif. Intell.*, 2025, **14**, 100417.
33 O. Ayala and P. Bechard, Reducing hallucination in structured outputs via Retrieval-Augmented Generation, *InProceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Industry Track, 2024, 6, pp. 228–238.
34 K. Shuster, S. Poff, M. Chen, D. Kiela and J. Weston, Retrieval augmentation reduces hallucination in conversation, *arXiv*, 2021, preprint, arXiv:2104.07567, DOI: 10.48550/arXiv.2104.07567.
35 B. Murdoch, Privacy and artificial intelligence: challenges for protecting health information in a new era, *BMC Med. Ethics*, 2021, **22**(1), 122.
36 C. Wang, J. Zhang, N. Lassi and X. Zhang, Privacy protection in using artificial intelligence for healthcare: Chinese regulation in comparative perspective, *Healthcare*, 2022, **10**(10), 1878.
37 N. Yadav, S. Pandey, A. Gupta, P. Dudani, S. Gupta and K. Rangarajan, Data privacy in healthcare: in the era of artificial intelligence, *Indian Dermatol. Online J.*, 2023, **14**(6), 788–792.
38 X. Wang, Z. Tang, J. Guo, T. Meng, C. Wang, T. Wang and W. Jia, Empowering edge intelligence: A comprehensive survey on on-device ai models, *ACM Comput. Surv.*, 2025, **57**(9), 1–39.
39 D. Dhagarra, M. Goswami and G. Kumar, Impact of trust and privacy concerns on technology acceptance in healthcare: an Indian perspective, *Int. J. Med. Inf.*, 2020, **141**, 104164.
40 D. Indrawan, D. S. Kusumo and S. Y. Puspitasari, Analysis of the implementation of mvvm architecture pattern on performance of ios mobile-based applications, *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, 2023, **8**(1), 59–65.
41 V. Sanh, L. Debut, J. Chaumond and T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, *arXiv*, 2019, preprint, arXiv:1910.01108, DOI: 10.48550/arXiv.1910.01108.
42 H. Touvron, T. Lavril, G. Izacard, X. Martinet, M. A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar and A. Rodriguez, Llama: Open and efficient foundation language models, *arXiv*, 2023 preprint arXiv:2302.13971, DOI: 10.48550/arXiv.2302.13971.
43 S. M. Jain, *Introduction to Transformers for NLP, With the Hugging Face Library and Models to Solve Problems*, 2022.