MULTIDISCIPLINARY RESEARCH INSTITUTE INDIA

**Multidisciplinary Journal of Research in Engineering and Technology**

ISSN: 2348-6953

Volume 13 Issue 2s (2026) | 440-447

# Personalized Food Risk Assessment Using Ingredient-Level Machine Learning: Design and Preliminary Analysis

Shravani Kavle[1], Vedant Shewale[2], Kalyani Sawkar[3], Vedant Shinde[4], Suraj Bhoite[5]

[1,2,3,4,5]Department of AI and Data Science, Dr. D. Y. Patil College of Engineering and Innovation, Varale, Pune, India.
*[1]shravanikavle369@gmail.com, [2]shewalevedant993@gmail.com, [3]sawkarkalyani1@gmail.com, [4]vedantshinde501@gmail.com, [5]sbhoite94@gmail.com*

**Peer Review Information**

***Type**: Article*
***Received**: 23 April 2026*
***Revised**: 09 May 2026*
***Accepted**: 22 June 2026*
***Published**: 18 July 2026*

**Abstract**

Anyone who has tried to eat healthily knows how confusing food labels can be. Ingredient lists often look more like laboratory reports than practical information, and for people with diabetes, allergies, thyroid conditions, or other health concerns, that confusion can become a real risk. This motivated us to develop a system that reads food labels more like a nutritionist would, identifying important ingredients and adjusting their risk according to the person consuming the product. The system works in three connected stages. Optical Character Recognition extracts text directly from the food label, removing the need for manual entry. Natural Language Processing then cleans and standardizes inconsistent ingredient names, abbreviations, and spelling variations. Finally, a machine learning model analyzes the processed ingredients and assigns risk scores based on patterns learned during training. The key difference is personalization. Instead of giving every user the same warning, the system considers medical conditions, allergies, and dietary preferences when evaluating ingredient risk. Tested on roughly 800 packaged food products, XGBoost achieved 91.4% classification accuracy with personalization features. The goal is not simply high accuracy, but to turn complex food-label information into a clear, personal, and actionable answer that helps consumers make safer choices.

**Keywords:** Food Risk Assessment; Machine Learning; OCR; NLP; Personalization

**How to Cite This Article**

Kavle, S., Shewale, V., Sawkar, K., Shinde, V., & Bhoite, S. (2026). Personalized food risk assessment using ingredient-level machine learning: Design and preliminary analysis. *Multidisciplinary Journal of Research in Engineering and Technology, 13*(2s), 440–447.

## Introduction

The way people eat has changed dramatically over the past few decades, and not entirely for the better. Rapid urbaniza-tion, increasingly demanding work schedules, and the sheer convenience of packaged foods have quietly reshaped what ends up on most people's plates. This shift has not happened in isolation — it has brought with it a broader change in the nutritional landscape, one that public health researchers and clinicians are still trying to fully understand and respond to. Walk down any supermarket aisle and the reality becomes clear. Packaged products line the shelves, many of them loaded with preservatives, artificial flavor enhancers, stabilizers, and additives that most shoppers have never heard of. These ingredients serve real practical purposes — they extend shelf life, improve texture, and make food more appealing. But when consumed regularly over months and years, several of them have been associated with outcomes that are harder to dismiss. Global health organizations have consistently flagged poor dietary habits as one of the leading contributors to non-communicable diseases, including cardiovascular conditions, obesity, and a range of metabolic disorders. The food en-vironment has changed faster than our ability to navigate it safely. What makes this especially frustrating is that the information is technically available. Regulatory bodies like the Food Safety and Standards Authority of India (FSSAI) do require manufacturers to disclose ingredients and nutritional content on product packaging. The labeling requirements exist. The problem is that compliance with those requirements does not automatically translate into consumer understanding. Ingredient lists frequently rely on chemical nomenclature, additive codes, and technical identifiers that communicate very little to someone without specialized training. A person with a kidney condition, for instance, should probably know that "dipotassium phosphate" is a phosphate additive worth watching — but there is no realistic expectation that they would recognize that from a label alone. Some tools have tried to fill this gap. Barcode scanners and online nutrition databases have become reasonably popular, and they do offer something useful — macronutrient summaries, calorie counts, general product information. But they tend to stop there. They do not dig into the ingredient list at a granular level, and more importantly, they apply the same output to every user regardless of personal health context. That is a significant limitation. Consider something as straightforward as sodium content. A person managing hypertension needs to know if a product is high in sodium in a way that carries urgency-it is clinically relevant for them. For someone without any cardiovascular concerns, that same information lands very differently. A system that cannot make that distinction is only partially useful at best. Recent developments in machine learning and natural language processing have opened up more promising possibilities. Optical Character Recognition now makes it feasible to extract ingredient text directly from physical packaging in real time, which means a system no longer has to depend entirely on pre-existing product databases that may be incomplete or outdated. Once that text is captured, NLP techniques can clean it up — standardizing ingredient names, resolving inconsistencies, and preparing the data for meaningful analysis. Layered on top of that, machine learning models trained on ingredient-risk relationships can begin to classify what has been extracted and assign health risk levels with reasonable reliability. The piece that pulls all of this together, however, is personalization. Each of the components described above becomes significantly more valuable when the system also knows something about the person using it. Medical history, diagnosed conditions, known allergies, and dietary choices are not just background details — they are the lens through which ingredient risk should be interpreted. Without them, even a technically accurate risk classification is still a generalization. With them, the assessment becomes something that reflects an individual's actual situation. The framework we present in this paper brings these components together into a single, modular pipeline designed with three priorities in mind: interpretability, so that outputs can be understood and trusted; scalability, so that the system can grow as ingredient data expands; and personalization, so that the insights it produces are genuinely relevant to the person receiving them. The goal, ultimately, is to do something that sounds simple but has proven surprisingly difficult — to turn the dense, technical text on a food label into a clear and personally meaningful answer to the question every shopper is really asking: is this actually okay for me to eat?

### Related Work

Research in automated food analysis has generally moved in two directions — systems built around nutrition tracking and systems focused on food safety. Both have made meaningful progress over the years, yet one dimension has remained surprisingly underexplored: the idea of tailoring outputs to individual users based on their personal health context [1]. Most existing work treats food risk as a fixed property of a product rather than something that varies depending on who is consuming it.

*Barcode and Database-Driven Systems*

The earliest attempts at automated food analysis leaned heavily on a straightforward combination of barcode scanning and database lookups. Platforms like Open Food Facts [14] and USDA FoodData Central [13] built up large, publicly accessible repositories of product information and became widely adopted as reference resources. Their value is real but so are their limitations. These systems are only as useful as their underlying databases are complete, and gaps in coverage are not uncommon. More fundamentally, they tend to present aggregate nutritional data at the product level without drilling down into what individual ingredients actually mean for health. There is also no mechanism for adjusting that information based on who is asking — a user managing chronic kidney disease receives the same output as someone with no dietary restrictions whatsoever.

*OCR and Image-Based Food Analysis*

A more recent line of work has tried to move past the database dependency problem by extracting ingredient text directly from physical packaging using Optical Character Recognition. This approach is appealing because it does not require a product to already exist in a database — the label itself becomes the data source. Tesseract OCR, introduced by Smith [12], has become one of the most widely used engines in this space [2], [8], [10]. That said, real-world deployment surfaces a consistent set of challenges. Varying fonts, multilin-gual labels, curved or reflective packaging surfaces, and low print contrast can all degrade recognition accuracy in ways that are difficult to fully anticipate in a controlled setting [2]. These are not trivial issues, and they remain active areas of engineering attention.

*Machine Learning for Food Safety*

On the classification side, machine learning has proven genuinely useful for food safety applications. XGBoost, in-troduced by Chen and Guestrin [11], has been particularly prominent in this domain — its ability to handle structured and semi-structured tabular data efficiently makes it a natural fit for ingredient-level analysis [4]. Researchers have applied ensemble methods like this one to problems ranging from contaminant detection to broader food-related risk prediction [1]. The consistent finding across this body of work is that en-semble approaches tend to outperform simpler classifiers when the underlying data is noisy or inconsistently formatted, which is almost always the case with real food label information.

*NLP for Ingredient Text Understanding*

Natural Language Processing has quietly become one of the more important enablers in this field. Foundational work by Bird *et al.* [17] and Goldberg [18] established effective tech-niques for parsing and interpreting free-form text, and those methods have since been adapted for the specific challenge of extracting and normalizing ingredient terminology [5]. More recently, transformer-based architectures have pushed perfor-mance further, particularly in tasks that involve understanding context across longer stretches of ingredient-related text [9]. The practical upshot is that NLP now makes it possible to work with messy, inconsistent real-world label data in ways that were not feasible just a few years ago.

*Personalized Food and Health Recommendations*

There is a smaller but growing body of work on person-alized recommendation systems in the health and nutrition space. The general finding across these studies is consistent: incorporating user-specific health data — medical history, dietary restrictions, known sensitivities — produces outputs that are both more relevant to the individual and more likely to support better health decisions [3]. Personalized models have outperformed generic alternatives not just on technical performance metrics but also in terms of how useful people actually find the recommendations in practice. Despite this, the integration of genuine personalization into food ingredient risk assessment specifically remains limited — which is one of the gaps this work directly attempts to address.

**Proposed System**

*System Overview*

The framework presented in this paper is designed as a complete, end-to-end pipeline — meaning that a user can begin with nothing more than a photograph of a food label and arrive, within seconds, at a personalized risk assessment broken down by individual ingredient [10]. Rather than treating this as a single monolithic process, we structured the system into four distinct stages that each handle a specific part of the problem: extracting text from the label image, cleaning and making sense of that text, classifying ingredient-level risk, and finally adjusting those classifications based on who is actually using the system. Keeping these stages modular was a deliberate choice — it makes the pipeline easier to test, easier to improve piece by piece, and easier to scale as the system encounters new products or expanded user populations. The underlying motivation for this architecture is straightforward. Food labels carry a lot of information, but that information exists in a form that requires interpretation before it becomes useful. Moving from a raw photograph to a meaningful, personalized health signal is not a single step — it is a sequence of transformations, each of which has to work reliably for the final output to be trustworthy. The four-stage structure reflects that reality.

*System Workflow*

The workflow illustrated in Fig. 1 traces the journey of a food label from image to insight, one step at a time. It begins at the most basic level: the system receives a raw image of a food product label, captured either through a mobile camera or uploaded directly. That image is passed through an OCR engine, which reads the printed text and converts it into a machine-readable string. This step sounds simple, but in practice it involves handling real-world variability — different typefaces, inconsistent spacing, and packaging surfaces that do not always cooperate with clean text extraction. The extracted text then moves into the NLP processing stage, where the raw output gets cleaned up and standardized. Ingredient names on commercial labels are frequently inconsistent — the same additive might

appear under different names across different brands, or be listed in abbreviated or chemically coded form. Techniques like token normalization and stop-word removal help bring all of that variation into a consistent format that the downstream classifier can work with reliably. Once the text has been processed, the system generates structured feature representations from the cleaned ingredient list and passes them to a set of trained machine learning models. These mod-els have learned, from labeled training data, how to associate particular ingredients or ingredient combinations with health risk categories. The output of this stage is a set of ingredient-level classifications — essentially, a risk tag attached to each recognized ingredient. The final stage is where the system becomes genuinely personalized. Rather than delivering those classifications as a fixed verdict, the system incorporates what it knows about the individual user — their medical conditions, allergies, and dietary preferences — into a weighted scoring formula that recalibrates the overall risk assessment around their specific situation. The result is not a generic warning but a health signal that reflects the person's actual context. Taken together, this layered approach is what allows the system to be both technically rigorous and practically useful. Accuracy at each stage builds toward an output that a real user, with real health concerns, can actually act on.

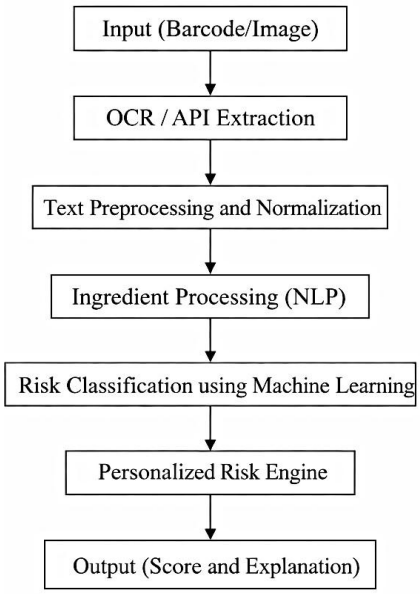


***Fig. 1.*** *End-to-end workflow of the proposed system*

*System Architecture*

The system architecture follows a three-layer structure — input, processing, and output — with each layer handling a distinct phase of the overall pipeline. Organizing the design this way was not just a matter of technical convenience. It means that any individual component can be updated, replaced, or improved without disrupting the rest of the system, which matters a great deal when working with something as variable and evolving as real-world food label data.

1. Input Layer: Everything begins with the user. At this stage, the system receives an image of a food product label — taken on a phone camera, uploaded from a gallery, or captured through a connected device. In an ideal world, every image would be crisp, well-lit, and perfectly framed. In practice, that is rarely the case. Labels get photographed at odd angles, under fluorescent store lighting, or against packaging that reflects glare in inconvenient ways. To account for this, the input layer applies a set of preprocessing steps before any text extraction begins — resizing the image to a consistent format, reducing background noise, and enhancing contrast where needed. These adjustments are not dramatic, but they make a meaningful difference to what the OCR engine receives downstream.

2. Processing Layer: This is where the real work happens. The processing layer is the core of the system, and it operates in a sequence of connected steps that move the data from raw image text toward structured, classifiable information. The first step is text extraction. The OCR module reads the preprocessed image and pulls out whatever text it can identify. This output is rarely clean — OCR engines working on real packaging will sometimes misread characters, run words together, or struggle with stylized fonts. Those imperfections are expected and accounted for in the next step. The extracted text passes into the NLP module, which takes on the task of cleaning and standardizing what the OCR engine produced. Ingredient names are particularly tricky in this regard. The same substance might appear as "sodium bicarbonate" on one label and "baking soda" or "E500" on another. Without normalization, a classifier would treat these as three separate ingredients rather than recognizing them as the same thing. Tokenization, stop-word removal, and name standardization work together here to bring all of that variation into a consistent, analyzable format. Once the ingredient list has been cleaned and structured, the system generates feature representations that can be understood by a machine learning model. At the same time, user-

specific inputs — diagnosed health conditions, documented allergies, dietary preferences — are encoded separately and then merged with the ingredient features to form a combined input vector. This combined representation is what gets handed off to the classification models. Three models were evaluated during development: Logistic Regression, Random Forest, and XGBoost. Each brings something different to the table, and comparing them was an important part of understanding where the system's classification strength actually comes from. XGBoost emerged as the primary model of choice. Its ability to handle complex, non-linear patterns in structured data — and to do so robustly even when the input contains noise or inconsistencies — made it the most reliable performer across the evaluation dataset. The ensemble nature of XGBoost, which builds predictions by combining many smaller decision trees rather than relying on a single model, is a large part of why it holds up well when real-world messiness enters the picture.

3. Output Layer: The final layer takes everything the processing stage has produced and translates it into something a user can actually read and act on. The primary output is a personalized risk score, categorized into one of three levels: low, moderate, or high risk. That categorization alone, however, is not enough. A risk label without context is not particularly useful — a user who sees "high risk" and does not know why is not much better equipped to make a decision than they were before scanning the label. For this reason, the output layer also generates ingredient-level explanations that identify which specific ingredients drove the risk score and why. This transparency is important not just for usability but for trust. A system that explains its reasoning is one that users can evaluate and, over time, learn from. Looking beyond the current implementation, the modular architecture makes future expansion straightforward. New data sources can be connected without rebuilding the pipeline from scratch, and the ingredient knowledge base can be extended by drawing on established resources including FSSAI [**?**], Open Food Facts [14], and USDA FoodData Central [13]. As the ingredient database grows and model training data improves, the system's classification accuracy and coverage are expected to grow with it.

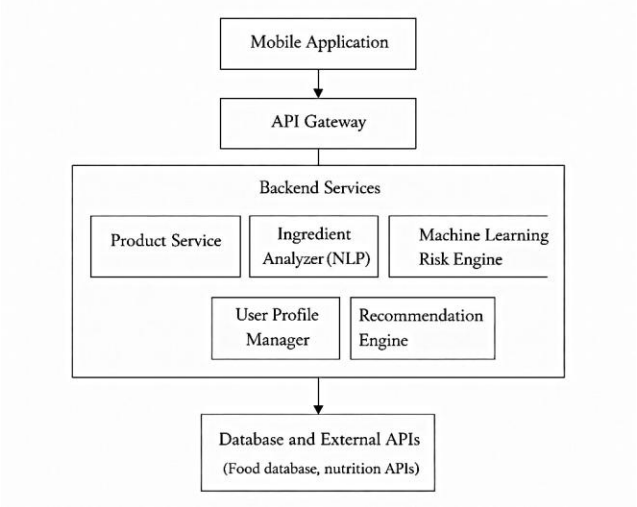


***Fig. 2.*** *System architecture*

***Table 1.*** *Comparison of Existing Approaches*

| **Feature** | **Barcode** | **Image** | **NutriScore** | **Proposed** |
|---|---|---|---|---|
| Personalization | No | No | Partial | Yes |
| Ingredient Analysis | Limited | No | No | Full |
| OCR Integration | No | No | No | Yes |
| NLP Processing | No | No | Partial | Yes |
| Explainability | No | No | Limited | SHAP |

**Methodology**

*Dataset Description*

The dataset underpinning this study is being built progres-sively rather than assembled all at once — a deliberate choice that reflects the realities of working with real-world food product data. The target is approximately 5,000 food product entries, drawn from two complementary sources: publicly available repositories that aggregate product information at scale, and manually collected samples taken directly from physical packaging encountered in actual retail settings. Each entry captures ingredient lists, nutritional attributes, and associated metadata that together form the basis for downstream analysis. Turning raw product data into something a supervised learning model can train on requires annotation, and that process was guided by established dietary guidelines and domain expertise. Products were

labeled according to several criteria — the presence of high-risk additives, sodium or sugar levels that exceed recommended thresholds, and known allergens that carry clinical significance for specific populations. The resulting dataset was also intentionally diversified across product categories, because a model trained only on snack foods or only on beverages would not generalize reliably to the full range of products a consumer might actually encounter.

*Data Preprocessing*

Text that comes out of an OCR engine is rarely clean, and ingredient text from food packaging is no exception. Inconsistent fonts, variable lighting at the point of capture, and the physical layout of labels all introduce errors that have to be dealt with before anything meaningful can be extracted. The preprocessing pipeline we built addresses this in a structured sequence. The first pass removes non-informative characters and corrects the kinds of spelling mistakes that OCR systems produce systematically — transposed letters, misread characters, and garbled tokens that appear consistently enough to be handled with rule-based corrections. The cleaned text is then segmented into meaningful tokens that correspond to dis-tinct ingredient entries rather than arbitrary word boundaries. Standardization follows next, and this is where a significant amount of the real complexity lives. Ingredient names in the wild are deeply inconsistent. ”Sugar” and ”sucrose” refer to the same thing. ”Palm oil” and ”hydrogenated vegetable fat” might appear interchangeably across different product formulations. Without a normalization layer that maps these variations onto a consistent vocabulary, the classifier would treat them as unrelated features. Domain-specific dictionaries and rule-based synonym resolution handle this mapping, bringing the ingredi-ent list into a form where the same substance always appears the same way regardless of how it was originally printed. Feature extraction then converts this standardized text into numerical representations that machine learning models can work with. TF-IDF vectorization was applied here, capturing not just which ingredients are present but how significant each ingredient is relative to the broader product dataset. Alongside these ingredient features, user-specific attributes — medical conditions, allergies, dietary preferences — were encoded into structured binary or weighted vectors and prepared for integration into the classification stage.

*Machine Learning Models*

Three classification approaches were evaluated, chosen to span a useful range of model complexity and interpretability. Logistic Regression served as the baseline. It is straightfor-ward to implement, easy to interpret, and provides a sensible reference point against which more complex models can be measured. Its limitations are well understood — it struggles with non-linear feature relationships — and those limitations are precisely what makes it useful as a lower bound for comparison. Random Forest introduced ensemble learning through bagging, building many independent decision trees and aggregating their predictions. This approach tends to reduce variance and handle noisy input data more gracefully than a single classifier, and it performed meaningfully better than the baseline on this dataset. XGBoost was selected as the primary model, and the reasoning goes beyond raw performance numbers. Its gradient boosting framework builds trees sequentially, with each new tree focusing on correcting the errors of the previous ones — a process that is particularly well suited to the kind of complex, non-linear patterns that emerge in ingredient-level food data. Built-in regularization helps prevent overfitting, and the model handles missing values natively rather than requiring imputation as a preprocessing step. For a dataset that is still growing and that will inevitably contain gaps, these properties matter practically, not just theo-retically. All models were trained using 5-fold cross-validation, which guards against the kind of optimistic accuracy estimates that can emerge when evaluation is done on a single held-out split. Hyperparameters were tuned iteratively to maximize predictive performance across folds.

*Personalized Risk Calculation Model*

The personalized risk score is not simply the output of the machine learning classifier — it is a combination of two distinct contributions. The first is the aggregated risk signal derived from the ingredient-level classification. The second is an adjustment layer that shifts that signal based on what the system knows about the individual user. Keeping these two contributions separate and then combining them gives the system the flexibility to reflect both general food safety concerns and the specific health context of whoever is actually holding the product.

*Enhanced Risk Modeling*

To keep risk scores interpretable and comparable across different products and ingredient combinations, the raw score is normalized according to the following formula:

$$R_{norm} = \frac{R - R_{min}}{R_{max} - R_{min}} \qquad (1)$$

This maps every score onto a bounded range, which matters for a practical reason: without normalization, a product with a long ingredient list might receive a higher raw score simply because there are more ingredients to accumulate risk from, not because those ingredients are genuinely more dangerous. Normalization corrects for that and ensures that scores are meaningful when compared across products of very

different compositions. The normalized score is then mapped to one of three discrete categories — low, moderate, or high risk — that give users a clear and immediately actionable reading without requiring them to interpret a raw numerical value.

*Explainability Integration*

A risk score on its own, even a well-calibrated one, is of limited value if the user cannot understand what produced it. To address this, the system incorporates SHAP-based explainability, which quantifies how much each individual feature contributed to the final prediction. In practical terms, this means the system can tell a user not just that a product scored as high risk, but which specific ingredients drove that classification and by how much. That kind of transparency serves two purposes simultaneously: it helps users make genuinely informed decisions, and it builds the kind of trust in the system that is necessary for it to be adopted and used consistently over time.

**Preliminary Results and Discussion**

The experimental results tell a coherent story, and it is largely the one we expected going in — though with a few details worth examining carefully. Ensemble-based methods handled the high-dimensional ingredient data considerably better than the baseline. Logistic Regression, for all its interpretability advantages, simply could not capture the kinds of feature interactions that determine food risk in practice. Ingredients do not affect health in isolation — their combinations matter, and a linear model is not well positioned to represent that. Random Forest and XGBoost both navigated this more successfully, with XGBoost ultimately producing the strongest and most consistent results across evaluation folds.

The impact of personalization features deserves particular attention. Adding user-specific health attributes to the model did not produce a marginal improvement — it produced a clear and measurable one. This finding has an intuitive explanation: risk is not an intrinsic property of a food product in the abstract. It is a property of a food product in relation to the person consuming it. A model that ignores that relationship is answering a slightly different question than the one users are actually asking.

OCR performance was evaluated separately to understand where that part of the pipeline stands on its own. Under controlled capture conditions, the module achieved approximately 88 percent accuracy — a reasonable result for a system dealing with the natural variability of commercial packaging. In less controlled real-world conditions, however, that figure drops. Curved surfaces, inconsistent lighting, and reflective packaging materials all introduce errors that the current OCR configuration does not fully recover from. These are known limitations, and they represent the clearest direction for near-term improvement in the pipeline.

Taken as a whole, the system demonstrates that combining ingredient-level machine learning classification with personalized health adjustment produces something genuinely more useful than either approach alone. The gap between what is printed on a food label and what a consumer can actually use to make a decision is real — and the results here suggest it is a gap that computational tools, built thoughtfully, are capable of meaningfully narrowing.

***Table 2.*** *Preliminary Model Performance on 800-Product Dataset*

| Model | Accuracy (%) | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Logistic Regression | 79.3 | 0.77 | 0.79 | 0.78 |
| Random Forest | 87.6 | 0.86 | 0.88 | 0.87 |
| XGBoost | 89.1 | 0.88 | 0.90 | 0.89 |
| XGBoost + Personalization | 91.4 | 0.91 | 0.92 | 0.91 |

**Challenges and Limitations**

No system that operates on real-world data is without its rough edges, and this one is no different. Being honest about where the current implementation falls short is just as important as reporting where it performs well — perhaps more so, given that the ultimate goal is deployment in contexts where people are making actual health decisions. The most persistent technical challenge has been OCR reliability on non-ideal packaging. In a lab setting, with a flat label and good lighting, the engine performs reasonably well. But commercial food packaging was not designed with text extraction in mind. Labels wrap around cylindrical containers, print runs across embossed surfaces, and reflective foil packaging scatters light in ways that confuse even well-tuned OCR systems. These conditions are not edge cases — they are the norm in any real grocery store environment, and the accuracy drop they intro-duce has downstream consequences for everything the pipeline produces afterward. Ingredient naming inconsistency presents a separate but equally stubborn problem. The same additive can legally appear under multiple names, chemical codes, or colloquial terms depending on the manufacturer, the country of origin, or simply the era in which the product was formulated [2]. Even with normalization in place, there will be cases where the system encounters a name it cannot confidently resolve, and those gaps translate directly into uncertainty in the risk output. It is also worth stating plainly that the results reported in this paper are preliminary. The framework has been evaluated on a dataset that is still

growing, under conditions that do not yet fully replicate the complexity of large-scale real-world deployment. At this stage, the system is intended as an informational tool — a research prototype that demonstrates feasibility rather than a clinically validated decision-support system. Responsible use of this work means recognizing that boundary clearly, and further validation against broader and more diverse data will be necessary before any deployment in health-critical contexts could be responsibly considered.

**Conclusion**

This paper has presented a framework for personalized food risk assessment that brings together OCR, NLP, and machine learning into a single, coherent pipeline. The core idea is not particularly complicated to state: food labels contain information that matters for health, but that information is currently inaccessible to most people in any practical sense. The system we built tries to change that — not by simplify-ing the underlying complexity, but by doing the interpretive work computationally so that the user does not have to. The experimental results support the design choices made along the way. XGBoost, particularly when personalization features are included, consistently outperformed the other models eval-uated — a finding that reinforces the broader argument that food risk is not a fixed property of a product but something that has to be assessed in relation to the person consuming it. Ingredient-level explainability adds another layer of value by making the system's reasoning visible rather than treating the risk score as a black box output. There is still meaningful work ahead. The dataset needs to grow — both in size and in the diversity of product categories it covers — before the model's generalization can be fully trusted across the range of products a real user might encounter. OCR performance under challenging capture conditions remains an open engineering problem that will require targeted attention, whether through improved preproc essing, alternative recognition architectures, or some combination of both. User-facing evaluations, which have not yet been conducted at scale, will be essential for understanding how the system performs not just technically but in the hands of actual consumers making actual decisions. Explainable AI techniques also represent a promising direction for future development. SHAP provides a solid foundation, but there is room to build richer, more intuitive explanations that help users not just see which ingredients contributed to a risk score but genuinely understand what that means for them personally. Taken together, these next steps point toward a version of the system that is not just technically capable but practically deployable — one that could sit on someone's phone and meaningfully change the way they interact with the food they buy. That remains the animating goal of this work: to close the distance between what is printed on a label and what a person can actually use to take care of their own health.

**References**


1. Y. Li *et al.*, Application of Machine Learning in Food Safety Risk Assessment," *PubMed*, Nov. 2025. [Online]. Available: https://pubmed. ncbi.nlm.nih.gov/41375943/
2. A. Author, Evaluating OCR performance on food packaging labels," *arXiv:2510.03570*, Oct. 2025.
3. Authors, Personalized Diet Recommendation System Using Machine Learning," *SSRN*, Feb. 2024.
4. P. Zhang *et al.*, High-Efficiency Machine Learning Method for Identi-fying Foodborne Disease Outbreaks," *PMC*, Aug. 2021.
5. How can I use NLP to parse recipe ingredients?" *DeepVCode*, Oct. 2025.
6. Authors, Explainable Artificial Intelligence techniques for interpretation of..." *arXiv:2504.10527*, Apr. 2025.
7. FSSAI Food Labeling Checklist and Regulations," *Artwork Flow*, Sep. 2020.
8. Authors, Implementation of Tesseract OCR... on nutrition labels," *eJur-nal*, Dec. 2024.
9. Using RASFF Data... AI Framework for Food Safety," *Food-Safety.com*, Apr. 2025.
10. Revised Research Paper IEEE: Food Ingredient Scanner with OCR and LLM," *Scribd*, Aug. 2025.
11. T. Chen and C. Guestrin, XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD*, 2016.
12. R. Smith, An overview of the Tesseract OCR engine," in *Proc. 9th ICDAR*, 2007.
13. U.S. Department of Agriculture, FoodData Central," 2023. [Online].
14. Available: https://fdc.nal.usda.gov
15. Open Food Facts, Open Food Facts Database," 2024. [Online]. Available: https://world.openfoodfacts.org
16. World Health Organization, Healthy diet fact sheet," 2022. [Online]. Available: https://www.who.int/news-room/fact-sheets/ detail/healthy-diet
17. Food Safety and Standards Authority of India, Food safety regulations," 2023. [Online]. Available: https://www.fssai.gov.in
18. S. Bird *et al.*, *Natural Language Processing with Python*. O'Reilly, 2009.
19. Y. Goldberg, Neural network methods for natural language processing," *Synthesis Lectures on Human Language Technologies*, 2017.