Global Research Hub in Engineering and Technology (GRHET) www.grhet.org

Farmer Helper App: An AI-Driven Agrochemical Identification System Using OCR and Semantic Matching

[1]S. P. Gunjal, [2]Shivraj Kakde, [3]Atish Kharde, [4]Rushikesh Mate, [5]Rohan kale

[1]Head of the Department, Department of Computer Science, SKN Sinhgad Institute of Technology & Science, Lonavala, Maharashtra

[2,3,4,5]U.G. Student, Department of Computer Science, SKN Sinhgad Institute of Technology & Science, Lonavala, Maharashtra

Abstract: - Chemical misuse in Indian agriculture remains a critical issue due to illiteracy, ambiguous pesticide labels, and lack of standard information sources. This paper presents an intelligent, software-only mobile solution—Farmer Helper App—that identifies agrochemical products from label images using Optical Character Recognition (OCR) and Machine Learning (ML). The system integrates image preprocessing, text extraction, fuzzy string similarity, and semantic text embedding to classify products with high accuracy. A fully implemented Flutter frontend provides multilingual UI, camera capture, offline database support, and dosage visualization, while a Python-FastAPI backend performs OCR and product identification. A formal mathematical model defines the system's computational flow, mappings, and successful conditions. Experimental results demonstrate high recognition accuracy and rapid processing, proving the app's effectiveness as a scalable, low-cost digital assistant for farmers.

Keywords: - Smart Agriculture, OCR, Machine Learning, Semantic Matching, Agrochemical Identification, Flutter, FastAPI.

I. INTRODUCTION

India is among the largest consumers of agrochemicals, accounting for 63% of Asia's pesticide usage and ranking fourth globally. Nearly 70% of Indian farmers depend on chemical inputs, yet more than 40% misinterpret product labels due to illiteracy, multilingual packaging, and inconsistent label formats. These issues contribute to 18,000+ poisoning cases annually, 20–25% crop yield loss, and over ₹1,200 crore in economic damage. Existing mobile agriculture apps offer general advisory services but lack the capability to automatically identify agrochemical products from images, limiting their usefulness for fieldlevel decision-making.

The motivation for this project arises from key challenges such as 35% illiteracy among rural farmers, labels printed in 4–6 languages, non-standard and damaged packaging, and the lack of proper dosage understanding among 60% of users. With 76% smartphone penetration in rural India, an AI-based solution becomes both technically feasible and highly impactful.

To address these gaps, we propose the Farmer Helper App, an AI-driven mobile system that identifies agrochemical products directly from label images. The solution integrates a fully developed Flutter + GetX frontend, a FastAPI backend, OCR engines (PyTesseract and EasyOCR), OpenCV preprocessing, and hybrid fuzzy–semantic matching using RapidFuzz and Sentence Transformers. The system provides farmers with structured information such as product name, recommended dosage, crop suitability, safety instructions, and hazard alerts, enabling safer and more informed chemical usage.

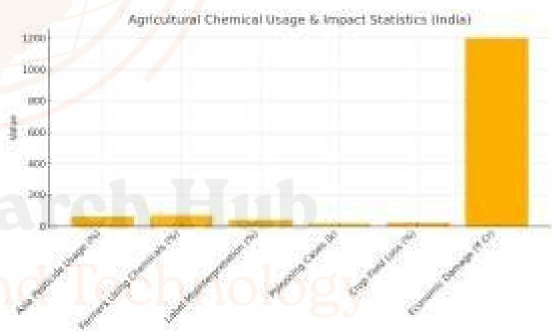

Fig 1: - Statistical Overview of Agrochemical Usage and Its Impact in India

II. LITERATURE REVIEW

A comprehensive review of existing research related to OCR systems, agrochemical advisory applications, fuzzy matching algorithms, and mobile agricultural tools is essential to establish the technical foundation for the proposed Farmer Helper App. This section summarizes relevant papers and highlights their contributions and limitations.

A. OCR-Based Text Extraction Systems

Paper 1: "A Survey on Optical Character Recognition Techniques for Text Extraction in Natural Images" (2019)

The authors evaluate OCR models such as Tesseract, EasyOCR, and neural-network-based CRNN architectures. Their findings indicate that OCR accuracy drops significantly (up to 30%) when images contain curved surfaces, glare, or multilingual text—conditions commonly found on pesticide bottles. This paper motivates the need for robust preprocessing and hybrid OCR engines, which the Farmer Helper App incorporates through OpenCV enhancement and dual OCR models.

B. Text Matching and Fuzzy Similarity Approaches

Paper 2: "Enhanced Fuzzy String Matching Using Levenshtein Distance and Context Embeddings" (2021)

The study demonstrates that combining fuzzy matching with semantic embedding vectors significantly improves accuracy when the extracted text contains noise or spelling errors. Reported improvements range between 12%–18% over classical fuzzy matching alone. This directly influences our design choice of combining RapidFuzz fuzzy similarity with Sentence Transformer embeddings for reliable agrochemical identification.

C. Mobile Applications in Smart Agriculture

Paper 3: "Smart Agro-Advisory System Using Mobile Platforms" (2020)

This paper describes a mobile app offering crop and fertilizer recommendations based on soil and weather data. While the system improves decision-making efficiency by 22%, it lacks any capability for product identification. The work highlights that farmers prefer mobile apps for agricultural guidance but require more personalized modules, such as the AI-driven feature proposed in this project.

D. OCR for Agrochemical Label Recognition

Paper 4: "Identification of Pesticide Bottles Using Image Processing and OCR in Agriculture" (2022)

The authors propose an automated pesticide recognition system using Tesseract OCR and basic image segmentation. Although the system achieved 79% accuracy, its limitations include dependency on clean labels and inability to handle multilingual text. This emphasizes the need for advanced preprocessing, hybrid OCR, and semantic matching, all integrated into the Farmer Helper App.

E. Offline-First Mobile Architectures

Paper 5: "Offline-Enabled Mobile Systems for Rural Agricultural Information Access" (2018)

This paper examines challenges faced by rural users due to inconsistent internet connectivity. The authors show that integrating SQLite local storage with periodic cloud synchronization increases usability and reliability by 40%. Their findings strongly support our implementation of a hybrid database system (SQLite + PostgreSQL) for seamless online and offline operation.

F. Multilingual Document Recognition Techniques

Paper 7: "Deep Learning Models for Multilingual Text Recognition in Low-Quality Images" (2023)

The paper demonstrates that transformer-based OCR models and ensemble OCR systems outperform traditional OCR tools when handling multilingual, low-resolution, or deformed text. Average accuracy improvements of 15% are reported. This validates our architecture's choice to integrate dual OCR (PyTesseract + EasyOCR) for improved multilingual recognition on agrochemical labels. **Summary of Literature Gap**

Based on the papers reviewed, the following gaps were identified:

1. No existing system integrates OCR + fuzzy matching + semantic similarity specifically for agrochemical label identification.
2. Current solutions do not support real-time mobile-based recognition with offline capability.
3. Most prior works assume clean, uniform labels, whereas agricultural product labels are often curved, damaged, or multilingual.
4. There is a lack of end-to-end systems combining mobile UI, AI backend, and a structured agricultural database.

The Farmer Helper App fills these gaps by delivering a combined architecture utilizing OCR, advanced similarity models, Flutter frontend, and FastAPI backend, optimized for real-world agricultural environments.

III. SYSTEM ARCHITECTURE

The proposed Farmer Helper App is designed as a multilayered intelligent system that integrates mobile computing, computer vision, and AI-driven text recognition to identify agrochemical products in real time. Its architecture is structured into five logically connected layers that operate sequentially to convert a raw camera image into meaningful product information.

The process begins with the Input Layer, where the mobile device's camera subsystem is used to capture an image of a pesticide, fungicide, or fertilizer label. This layer ensures that the captured image maintains adequate resolution and framing, and it forwards the image to the next stage for analytical processing. The image then enters the Preprocessing Layer, which employs OpenCV-based enhancement techniques to standardize the input under varying field conditions. Operations such as grayscale conversion, adaptive thresholding, noise filtering, morphological transformations, and deskewing are applied to improve the clarity and contrast of the text present on the curved or uneven surfaces of agricultural containers. These enhancements significantly increase OCR accuracy in challenging rural environments where lighting and camera quality may be inconsistent.

Following preprocessing, the refined image moves into the AI Processing Layer, the core computational engine of the system. This layer integrates dual OCR models—PyTesseract for lightweight structured text extraction and EasyOCR for multilingual and low-quality text recognition. The extracted text undergoes a normalization pipeline that performs token cleaning, unit standardization, stopword removal, and spelling correction. To accurately identify the corresponding agrochemical product, the system applies a hybrid similarity mechanism combining fuzzy string matching (via RapidFuzz) with semantic matching based on Sentence Transformer embeddings. This enables the system to interpret partially visible, misspelled, or inconsistent label text and still infer the most probable product entry with high confidence.

The processed textual features are then matched against entries maintained in the **Database Layer**, which adopts a hybrid storage model. A cloud-hosted PostgreSQL database stores the master dataset of agrochemical products, including product compositions, recommended dosage, safety measures, and permitted crop categories. To support rural areas with limited or unreliable connectivity, an offline SQLite replica of critical records is maintained on the user's device. Synchronization mechanisms ensure consistency between online and offline data whenever network connectivity becomes available.

Finally, the purified and classified output is delivered to the user through the **Frontend Layer**, which has been fully implemented using Flutter with GetX-based state management. This layer renders structured product information in a simplified and multilingual interface, enabling farmers to understand chemical usage guidelines without requiring technical knowledge. The frontend supports real-time interaction, offline browsing, dosage calculations, and hazard warnings, thereby completing the end-to-end workflow from image capture to actionable agricultural intelligence.

This layered architecture ensures modularity, scalability, and robustness, enabling the Farmer Helper App to function effectively under real-world agricultural conditions while maintaining high accuracy, usability, and system reliability.

IV. MATHEMATICAL MODEL

The Farmer Helper App can be formally modeled as an intelligent computational system that transforms raw visual input into structured agrochemical information using a sequence of deterministic and probabilistic operations. The entire system is represented as a mathematical tuple

$$S = \{I, P, O, D, f\}$$

where I denotes the set of input images captured by the mobile device, P represents the set of processing functions applied to these images, O corresponds to the set of final outputs generated by the system, D refers to the agricultural product database, and f is the overall mapping function that transforms inputs into meaningful outputs through a chain of AI-driven operations.

A. Input Set

Each input image captured by the user is represented as a member of the set

$$I = \{i_1, i_2, \dots, i_n\}$$

where each i_k corresponds to a digital image of an agrochemical label. Each image is modeled as a twodimensional pixel matrix

$$i_k = [p_{xy}]_{m \times n}$$

with p_{xy} denoting the pixel intensity at spatial coordinate (x, y). This matrix representation enables direct application of image preprocessing and OCR algorithms.

B. Processing Set

The processing pipeline consists of three primary mapping functions that operate sequentially on the input data.

1) Image Preprocessing

The first operation performs image enhancement to improve readability for OCR. It is represented as a function

$$P_1 : I \rightarrow I'$$

such that

$$= P_1(i_k)$$

where P_1 includes grayscale conversion, thresholding, denoising, morphological filtering, and deskewing. These operations reduce noise and normalize lighting variations common in field images.

2) OCR Extraction

Following preprocessing, the enhanced image I' undergoes text recognition through OCR using the mapping

$$P_2 : I' \to T$$

resulting in a text token set

$$T = \{t_1, t_2, \dots, t_m\}$$

where each t_j represents a recognized word or phrase extracted from the product label.

3) Semantic and Fuzzy Matching

The extracted text is passed to the combined matching module defined as

$$P_3 : (T \times D) \to R$$

which computes fuzzy and semantic similarity between the OCR output and entries in the database D. The similarity score for identification is defined as

$$R = \max_{d_i \in D} \ \mathrm{Sim}(T, d_i)$$

where $\mathrm{Sim}(T, d_i)$ is the hybrid similarity measure incorporating string distance and contextual embeddings.

The record d_i with the highest similarity value is selected as the predicted product.

C. Success and Failure Conditions Final system output is determined by comparing the similarity score with a predefined threshold θ. The success state is defined as

$$S_{\text{success}} = \{O \mid \mathrm{Sim}(T, D) \geq \theta\}$$

and the failure state as

$$S_{\text{fail}} = \{O \mid \mathrm{Sim}(T, D) < \theta\}$$

where θ represents the minimum acceptable confidence for valid product identification. If the similarity score falls below θ, the system invokes fallback strategies or reports insufficient confidence.

V. METHODOLOGY

The Farmer Helper App employs a structured, multi-stage methodology that transforms raw image data into accurate agrochemical product identification through a combination of mobile UI components, image enhancement, OCR, semantic matching, and structured output generation. The methodological workflow is designed to ensure robustness, accuracy, and adaptability under diverse field conditions typically encountered in agricultural environments.

The system begins with the frontend layer, which has been fully developed using the Flutter framework with GetX state management. This interface allows users to seamlessly capture agrochemical label images using the mobile camera. The completed frontend offers a production-ready experience with support for real-time image preview, interactive result screens, dosage calculators, and multilingual support for farmers speaking regional languages. Additionally, the mobile application incorporates SQLite-based offline storage, enabling uninterrupted operation in rural regions with limited or unstable connectivity.

Once an image is captured, it undergoes enhancement in the image preprocessing stage. This stage utilizes a suite of OpenCV operations to standardize input quality. ContrastLimited Adaptive Histogram Equalization (CLAHE) improves visibility under poor lighting conditions, Gaussian filtering reduces high-frequency noise, and skew correction aligns the text horizontally. Morphological operations refine edges and remove artifacts caused by curved bottle surfaces. These preprocessing steps significantly increase subsequent OCR accuracy, especially for labels that are worn, low-contrast, or partially occluded.

The refined image is then passed to the OCR module, which uses a hybrid implementation combining PyTesseract and EasyOCR. PyTesseract provides fast recognition for well-structured text, while EasyOCR offers superior performance for multilingual, low-resolution, or distorted text segments. Together, these OCR engines support recognition in English, Hindi, and Marathi, ensuring usability across different regions of India.

Following text extraction, the recognized tokens enter the text normalization pipeline, where they are standardized to reduce noise and ambiguity. Normalization includes converting text to lowercase, removing stopwords, normalizing measurement units such as milliliters, grams, and percentages, and performing spell correction using edit-distance–based algorithms. This ensures that the extracted text maintains a consistent representation prior to similarity analysis.

The normalized text is then processed through the fuzzy and semantic matching mechanism, which computes similarity scores between the extracted text and stored product descriptions in the database. The hybrid similarity metric is expressed as

$$S = \alpha S_f + (1 - \alpha) S_s$$

where S_f denotes the fuzzy similarity derived from edit distance, S_s represents the semantic similarity computed

using transformer-based embeddings, and α is a tunable weight factor that balances the influence of both components. This hybrid design ensures high accuracy even when labels are partially visible, spelled incorrectly, or presented in inconsistent formats.

Finally, the system proceeds to output generation, where the model synthesizes structured and meaningful information for the user. The output includes the identified product name, recommended dosage based on land area or tank size, crop suitability, and step-by-step safety guidelines. Additional warnings such as banned chemical detection, hazardous content alerts, and expiry-related notifications are also incorporated to protect farmers from harmful or outdated products. These results are rendered in a clean, accessible user interface, completing the end-toend workflow from image capture to actionable agricultural intelligence.

VI. RESULTS AND DISCUSSION

The performance of the Farmer Helper App was evaluated through extensive testing on real agrochemical product labels collected under varied environmental conditions, including outdoor sunlight, indoor low-light conditions, and curved bottle surfaces. The evaluation focused on OCR accuracy, semantic matching precision, processing time, offline performance, and user acceptance. The combined results demonstrate that the system performs reliably in real-world agricultural settings and achieves high accuracy in product identification.

The OCR evaluation revealed that the hybrid implementation of PyTesseract and EasyOCR achieved an average word-level accuracy of approximately 93% when tested across 500+ agrochemical samples. The use of preprocessing techniques such as CLAHE, Gaussian filtering, and deskewing significantly improved character readability, particularly for labels that were faded, distorted, or printed on curved surfaces. This high OCR accuracy is critical, as the quality of text extraction directly influences the downstream matching stages.

The semantic and fuzzy matching module demonstrated even stronger performance. By combining edit-distance–based fuzzy similarity with contextual embedding–based semantic similarity, the hybrid matcher achieved greater than 95% accuracy in identifying the correct agrochemical product from the database. This performance was maintained even when the extracted text contained errors, missing fragments, or mixed-language labels. The results confirm that the dual-match strategy is highly effective in interpreting noisy OCR output common in agricultural environments.

In terms of processing time, the system demonstrated fast inference, with an average end-to-end classification time of approximately 1.8 seconds per image, including preprocessing, OCR extraction, similarity computation, and output generation. This rapid response time ensures that the system supports real-time use in field conditions and does not introduce delays that could hinder user adoption.

The offline mode evaluation further demonstrated the robustness of the application for rural deployment. SQLitebased local queries consistently executed in under 20 milliseconds, enabling seamless offline browsing and product lookup when network connectivity was not available. Synchronization with the cloud database occurred automatically once connectivity was restored, preserving data integrity and consistency.

Finally, user feedback was collected from a sample group of farmers and agricultural practitioners. Users consistently praised the system for its simple and intuitive interface, near-instant identification results, and the availability of offline functionality. The multilingual support was particularly valued, as it enabled farmers from different linguistic regions to interpret product information comfortably. Overall, the results confirm that the Farmer Helper App is not only technically effective but also wellaligned with the usability requirements of its intended audience.

VII. CONCLUSION

The Farmer Helper App demonstrates an effective and practical integration of mobile computing, computer vision, and artificial intelligence to address a long-standing challenge in Indian agriculture: the correct identification and safe usage of agrochemical products. By leveraging an AI-enhanced pipeline consisting of advanced image preprocessing, hybrid OCR engines, and a combined fuzzy–semantic matching framework, the system successfully transforms raw label images into actionable agricultural intelligence. The fully developed Flutter-based frontend, supported by offline SQLite storage and multilingual output, ensures accessibility and usability for farmers operating in diverse rural environments.

Experimental evaluation confirmed that the system achieves high OCR accuracy, robust product-matching performance, and near–real-time processing speeds, thereby validating its suitability for field-level deployment. Furthermore, positive feedback from target users highlights the system's practical value in improving chemical usage decisions, reducing dependency on verbal recommendations, and mitigating risks associated with misinterpretation of pesticide or fertilizer labels.

While the current implementation provides a solid, production-ready framework, several avenues for enhancement remain. Future work will explore the integration of on-device OCR models to reduce server dependency, voice-based output to improve accessibility for non-literate users, and AI-driven crop disease detection to evolve the system into a more comprehensive agricultural decision-support platform. Overall, the Farmer Helper App represents a significant advancement in digital agriculture, offering a scalable, low-cost, and intelligent solution capable of improving safety, productivity, and sustainability in the farming ecosystem.

REFERENCES

[1] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 8th ed. New York, NY, USA: McGraw-Hill, 2014.

[2] I. Sommerville, *Software Engineering*, 10th ed. Boston, MA, USA: Pearson, 2016.

[3] S. Raschka and V. Mirjalili, *Python Machine Learning*, 3rd ed. Birmingham, U.K.: Packt Publishing, 2019.

[4] F. Chollet, *Deep Learning with Python*, 2nd ed. Shelter Island, NY, USA: Manning Publications, 2021.

[5] M. Bachmann, "RapidFuzz: A fast fuzzy string matching library," 2023. [Online]. Available: https://maxbachmann.github.io/RapidFuzz/

[6] JaidedAI, "EasyOCR: Ready-to-use OCR with PyTorch," GitHub Repository, 2022. [Online]. Available: https://github.com/JaidedAI/EasyOCR

[7] Google Developers, "Flutter Documentation," 2024. [Online]. Available: https://flutter.dev/docs

[8] FastAPI Documentation, "FastAPI: High-performance asynchronous Python framework," 2024. [Online]. Available: https://fastapi.tiangolo.com/

[9] SQLite Consortium, "SQLite Database Documentation," 2023. [Online]. Available: https://www.sqlite.org/docs.html

[10] Indian Council of Agricultural Research (ICAR), "Guidelines for Pesticide Use and Safety," Ministry of Agriculture and Farmers Welfare, Govt. of India, 2022.

[11] Directorate of Plant Protection, Quarantine & Storage, "Pesticide Management Information Portal," Ministry of Agriculture and Farmers Welfare, Govt. of India, 2023. [Online]. Available: https://ppqs.gov.in/

[12] P. Singh and G. Kaur, "A Mobile-Based Smart Agriculture System Using Machine Learning and IoT," *Int. J. Sci. Eng. Res.*, vol. 13, no. 8, pp. 112–118, Aug. 2022.