ISSN 2723-6471

Rapid Ecosystem-Driven Deep Learning: On-Device Grain Type Classification and Authentication using iOS Swift and Core ML

Trianggoro Wiradinata[1,*]

[1]*Universitas Ciputra Surabaya, CitraLand CBD Boulevard, Surabaya 60219, Indonesia*

(Received: February 25, 2026; Revised: April 20, 2026; Accepted: July 5, 2026; Available online: July 20, 2026)

Abstract

The 2025 Indonesian rice scandal highlighted major shortfalls in food security and the pressing need for robust, data-based verification of authenticity. The goal of this work is to design a fast, lightweight, and fully offline rice grain classification and verification system that can run directly on consumer mobile hardware. The basic idea to overcome the technical bottleneck of deploying complex computer vision models on edge devices is to use macOS and Apple's unified ecosystem as a rapid prototyping and deployment platform. The deliberate avoidance of fragmented and high-latency workflows such as external environments (e.g. TensorFlow or PyTorch) and intermediate formats (e.g. ONNX) is mentioned. The study contributes a streamlined pipeline that incorporates an Image Feature Print V1 feature extractor, natively trained with Create ML on a publicly available dataset of 75,000 balanced images of five rice varieties (Arborio, Basmati, Ipsala, Jasmine, and Karacadag), directly into a native iOS application built with SwiftUI. The novelty of this approach is the usage of native tools like VisionKit and Core ML, which enables the complete elimination of third-party bridging code that normally bloats the binary overhead. The results show excellent edge efficiency on an iPhone 15 with a median prediction rate of 2.75 ms, an initial load time of 0.54 ms and a compilation latency of 3.66 ms. Moreover, the results reveal that by employing aggressive data augmentations, including the addition of visual noise, blur, exposure adjustment, flipping and rotation to ensure robustness, and by deliberately not cropping to preserve absolute grain dimensions, the model achieved a remarkable overall accuracy of 97% with an ultra-compact deployment footprint of only 66 KB. These metrics demonstrate that a fast, fully offline and privacy-preserving verification system is well within reach with modern consumer hardware.

Keywords: Swiftui, Core ML, Create ML, Grain Classification, On-Device Inference, Computer Vision

1. Introduction

Rice adulteration is part of a wider global food fraud problem where economically motivated adulteration threatens consumers, producers, and the integrity of the supply chain worldwide [1]. In Southeast Asia rice authenticity is another concern as premium regional varieties such as Vietnam's ST25 rice from the Mekong Delta need to be protected from mislabeling and authenticity risks [2]. This problem is very relevant to Indonesia, where rice is a basic commodity that underlies food security, national stability and public welfare [3]. The country's oversight system was jolted awake by the widespread adulterated rice scandal in 2025 when the Criminal Investigation Agency and the Attorney General's Office investigated and uncovered a counterfeiting scheme involving over 200 brands. A worrying 86% of these brands were below national quality standards and 60% were sold above government price caps. The crisis, with an estimated economic loss of IDR 99 trillion per year, demonstrates the urgent and undeniable need for a reliable data-driven verification mechanism to protect consumers. Recent research has also shown that rice adulteration in Indonesia affects consumer perception and purchasing decisions, and raises concerns about the liability of business actors and government supervision in cases of adulterated rice [4], [5].

Regarding the limitations of conventional inspection, most inspection currently takes place through manual observation and only a small part of the testing is done in laboratories. Laboratory test methods are precise, but too slow and not practical for widespread distribution. This method is also not suitable for continuous quality surveillance once the product is in the market. It's a perfect example of the gap between regulatory oversight and technological capabilities that can be narrowed down by the use of computer vision technology that uses machine learning. Driven by recent advances in machine vision for rice classification, work in computer vision has shown promise in automating rice grain

***Corresponding author: Trianggoro Wiradinata (twiradinata@ciputra.ac.id)**
DOI: https://doi.org/10.47738/jads.v7i3.1414

ISSN 2723-6471

classification. Studies using 2D and 3D imaging have achieved accuracies above 95% [6], [7], [8], [9], [10], [11]. Models such as CNN and CNN-LSTM have better robustness to lighting and background variation, but they are limited to use outside of laboratory conditions due to difficult integration challenges and high cost [8], [12].

Moving toward real-time, scalable verification, the case of blended rice has highlighted the urgent need for a practical, low-cost, and field-implementable automated rice verification system for continuous monitoring across the supply chain. Modern implementations should combine two elements: image-based morphological and color analysis to verify rice grain type and purity, and 2D or 3D scanning to check grain uniformity. Integrating these two components into a practical, easy-to-use, and low-cost system can improve inspection quality and act as a proactive prevention method.

To achieve this, this study proposes a modern and efficient alternative that leverages Apple's iOS framework for an on-device inference. By using SwiftUI for user interface design, AVFoundation for high-quality imaging and video recording, VisionKit for image preprocessing and segmentation, Create ML for dataset curation and model training, and Core ML for on-device inference, the system simplifies the entire computer vision workflow for rice grain classification. Thus, real-time rice grain type inspection can be performed using mobile devices such as an iPhone or iPad, significantly reducing reliance on laboratory infrastructure, cloud hyperscalers, SaaS and ensure convenience for operators across the supply chain, including consumers.

Leveraging machine learning capabilities directly on these portable devices offers a promising pathway for rapid, lightweight and low-cost rice verification. However, while the final goal is to deploy this technology for field-level adulteration detection, the current study serves strictly as a phase-one proof-of-concept. By utilizing a benchmark dataset with controlled, uniform backgrounds, this research focuses specifically on validating the technical feasibility and computational efficiency of the native iOS Core ML pipeline. Testing in real-world environments such as identifying mixed grains or processing images with diverse lighting and complex backgrounds is deliberately reserved for future iterations. By establishing this baseline architectural pipeline, this study demonstrates how modern consumer hardware can serve as a foundational step toward bridging the gap between academic innovation and practical food security monitoring.

2. Literature Review

Rice grain classification is a complex problem encompassing interrelated dimensions, such as morphological characterization (size and shape) and varietal identity. A dominant trend in the literature is the use of two- and three-dimensional image-based shape analysis techniques combined with machine learning models to differentiate rice grain varieties [13]. Based on these previous findings, this synthesis aims to (a) describe a framework for rice grain type classification, (b) review visual shape-based feature extraction methods, and (c) evaluate computer vision and machine learning approaches to existing classification strategies. The evolution of this reviewed literature reveals a critical bottleneck: the increase in the classification accuracy is accompanied by the growing unsustainability of the underlying infrastructure for practical field-level deployment. Very few studies tackled the simplicity of the deployment workflow. In this work, the feasibility of fully on-device deployment of complex classification models is explored to overcome these hardware limitations and infrastructure dependencies. Specifically, this research uses native iOS machine learning frameworks to build a lightweight mobile application that can function independently. This moves the paradigm away from high-latency preprocessing and heavy cloud-hosted GPU serving and offers a deployable solution that is not dependent on sophisticated external infrastructure.

2.1. Grain-Type Classification Schemes and Market Relevance

Rice grain is broadly categorised into three types, namely long-grain, medium-grain and short-grain. The classification is based on the ratio of the length to the width of the grain, which is closely related to the texture of the rice grain after cooking. This classification is important for marketing strategies, as the shape of the rice grain can give practical clues to its final quality. For example, it has been shown that the length and width of grain affect the cooking results and performance of rice during processing and therefore can be used to classify rice to serve various consumer and industrial needs [14]. Added to other features like rice grain density, if it is fully filled, the uniformity of its internal structure, categorisation has been a recent development. These factors affect the total quality of rice. The present 2D and 3D imaging technology and structured-light imaging have made it possible to determine these characteristics more accurately and thus the classification results have become more precise [6].

ISSN 2723-6471

2.2. Imaging Modalities and Feature Extraction

Most of the studies on rice grain classification are based on the extraction of important features like shape, colour, texture, and statistical descriptors using two-dimensional (2D) image analysis [15]. For example, the previous studies combined features such as shape and colour, and then used a Support Vector Machine (SVM) model to increase the accuracy of rice grain classification. Singh & Chaudhury [11] also used texture features and wavelet-based descriptors for classifying individual rice grains. In their study, they have stated that a combination of different types of features can improve the model performance in predicting rice grain varieties. In general, several previous studies have demonstrated that 2D imagery with appropriate feature extraction techniques and ML algorithms can be utilised to classify rice grain varieties with high accuracy.

Currently, both 2D and 3D image technology are capable of large-scale segmentation and classification based on important features. A digital image processing (DIP) pipeline for rice grain quality evaluation by automatic segmentation and geometric feature extraction was proposed by Kurade et al. [16]. Computer vision was employed by Kiratiratanapruk et al. [9] to classify 14 rice varieties, showing automation of all stages from preprocessing to classification. These studies provide a concrete example of the application of image-based systems for rice grain classification to detect fraud in rice grain distribution and trade.

2.3. Machine Learning Models for Rice Grain and Variety Classification

Ensemble-based machine learning approaches are essentially about combining a number of models to work together, rather than just using a single model. This approach often can provide more consistent and accurate classification results. This is clearly demonstrated in the study of Setiawan & Oumarou [17] in which the bagging-based ensemble method was able to achieve better performance in identifying rice grain varieties especially in terms of accuracy, precision, recall and F-measure. Simply put, ensembles combine many individual predictions to create a single one, which significantly improves the overall accuracy of the model for predictions on new, unseen data and reduces the chances for localised classification errors.

Furthermore, feature selection and optimisation are extremely important because ML models tend to perform better when only relevant features are used. A lot of features can make the model slow, complex and even less accurate by including unnecessary or confusing information. The best features from the groups of morphology, colour and texture are selected using binary particle swarm optimisation (BPSO) [15] based on an optimisation approach that simulates the behaviour of animal herds. So, they did a good job at improving accuracy and removing less important features.

The results are consistent with the work of Thangavel & Sakthipriya [18] where the quality of the combination of features is more important than the number of features. The classification results are significantly improved when the selected features truly reflect the important characteristics of the rice grains and are applied with a proper ML model. This approach enhances both the accuracy and the generalisability of the model. Overall, the studies have shown that successful classification of rice grains depends not only on the ML algorithm used, but also on the way we select the best features and combine the advantages of multiple models. This opens the door for smarter, faster and more reliable machine learning systems for the food industry, as well as for the development of rice grain varieties with better characteristic combinations.

2.4. Visual Property of Rice Grain and Other Things to Consider in Classification

While many image-based methods focus on visual attributes, chemical attributes such as amylose content are also important factors that influence rice texture and consumer preferences. However, this biochemical analysis is a separate topic and needs to be discussed in depth which is beyond the scope of this study. However, some studies have suggested that the combination of visual features with chemical or spectroscopic data could provide a more complete evaluation of quality [19], [20]. Such hybrid approaches could be better positioned as a further development direction rather than the primary focus on image utilisation in rice classification and quality assessment. A comprehensive summary of these foundational studies, including their key findings and operational limitations, is compiled in table 1.

Table 1. Summary of Previous Studies

Source	Research Title	Main Findings	Limitation
[19]	Potential Applications of Computer Vision in Quality Inspection of Rice: A Review	Identified that visual traits (color, shape, size) are effective but lack unified automation standards	Limited to static, non-automated lab setups with high human dependency

[9]	Development of Paddy Rice Seed Classification Process using Machine Learning Techniques for Automatic Grading Machine	Successfully automated mechanical grading of 14 rice seed varieties with ~90% accuracy	Heavily reliant on custom, bulky industrial conveyor/sorting machinery
[15]	Combining Binary Particle Swarm Optimization With Support Vector Machine for Enhancing Rice Varieties Classification Accuracy	Proved that removing redundant visual features boosts SVM variety classification accuracy to 95.8%	High mathematical preprocessing latency required to select optimal feature vectors
[13]	A comparative study of state-of-the-art deep learning architectures for rice grain classification	ResNet50 achieved the highest overall variety classification accuracy at 98.6%	Huge model size (>25M parameters) requiring high-end server GPU hosting
[20]	Rice quality recognition method based on multimodal model	Proved that fusing optical and non-optical traits resolves visual ambiguities in highly similar varieties	Extremely high hardware barrier; no framework provided for lightweight mobile execution

3. Methodology

This paper describes the design and development of an efficient rice grain classification application by fast training and deployment of a machine learning model using Apple's iOS framework. Design and development. Preparing a dataset, training the model with Apple Create ML, and embedding the model in a SwiftUI app. The aim of using the Apple iOS ecosystem is to enable fast and easy application development while preserving classification accuracy. Figure 1 shows the flowchart of the rapid on-device grain classification methodology. It outlines the five stages from dataset acquisition to native iOS deployment.

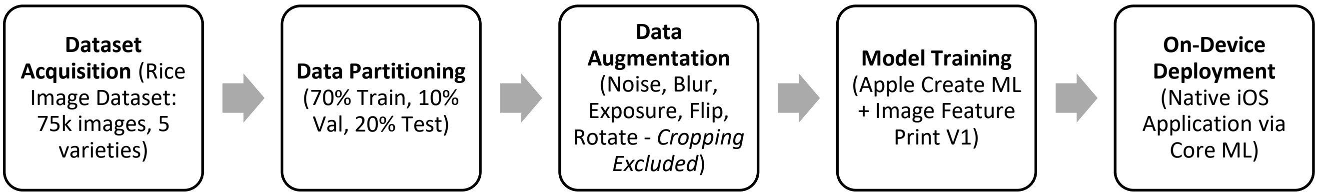

Figure 1. Development Methodology

The development workflow is considered extremely streamlined as it completely bypasses the complex intermediate conversion steps that are typically required when adapting external models for iOS. The normal pipeline is for developers to train a model in a framework like PyTorch, export it to an intermediate format like ONNX, and run secondary conversion scripts to produce a Core ML file, often requiring custom bridging headers for tensor inputs. This study, by comparison, makes use of Create ML, which directly generates a .mlmodel file that can be deployed, avoiding the third-party conversion bottlenecks and thus providing a smoother transition between training and application integration.

3.1. Dataset Preparation

The dataset used to train the machine learning model is the "Rice Image Dataset", which is a public dataset consisting of 75,000 images provided by Koklu et al. [10]. The dataset consists of images of five types of rice: Arborio, Basmati, Ipsala, Jasmine and Karacadag, each type being represented by 15000 images. The images with the rice grains on a uniform black background are shown in figure 2.

Figure 2. Sample Image from the Dataset Depicting Arborio, Basmati, Ipsala, Jasmine and Karacadag Rice Grains (left to right).

The classification architecture developed in this study is designed under a strict closed-set assumption; i.e., the pipeline is designed to evaluate only input images matching the five predefined rice grain classes. The model does not have an out-of-distribution (OOD) detection layer. Thus, feeding any non-rice object will make the network to force confidence scores over these known classes via nearest visual feature mapping. The dataset was split before training to properly train and evaluate the model, by a custom python script using the `os`, `shutil`, `scikit-learn`, and random libraries to randomly split the 75,000 photos into three unique sets. First, a Training Set of 70% of the data (52,500 photos) was assigned for the model to understand the characteristics of each rice class. Second, a validation set of 10% of the data (7,500 photos) was used during the training process to monitor performance metrics and avoid overfitting. Finally, a Test Set comprising the remaining 20% of the data (15,000 photos) was completely held back from the training phase, to be used to evaluate the model's final performance and classification accuracy independently.

3.2. Model Development

This model development was native to the Apple hardware ecosystem, with an Apple Silicon (M-Series) MacBook Air with 16GB of Unified Memory providing maximum computational efficiency with no need for an external server. The model was trained using the Create ML framework with the Image Feature Print V1 architecture, a feature extraction model that provides an accurate representation of the image data with a light workload. This model can be deployed as a very suitable baseline for the application with high performance and efficient power consumption. To artificially boost the diversity of the dataset and guarantee the robustness of the model against potential real-world variables, a suite of data augmentation techniques was applied directly during the training phase. The parameter augmentations were visual noise, to add random changes in pixels that simulate poor camera quality or low light conditions; blur, to make images less sharp and help the model learn to recognise objects that are out of focus or in motion; exposure, to control the brightness level so the model is accurate in bright and dim environments; flipping, to flip the image horizontally or vertically to train the flexibility of object orientation; and rotation, to rotate the image by a certain angle so that the model can recognise objects from different angles.

A major feature of Apple's Create ML framework is the deliberate abstraction of hyperparameters like batch size, learning rate, etc. to make machine learning development easier for iOS and macOS developers. This means augmentations might be added or removed, and their underlying parameters are controlled automatically by framework default configurations to provide consistency across the ecosystem. Importantly, during the preprocessing step, the crop feature was deliberately omitted from the data augmentation scheme following crucial findings from the initial analysis of the dataset used. The observation shows that some rice varieties have very similar morphological characteristics, as shown visually in figure 3, and the main difference to distinguish the classes is usually their absolute size, i.e., the length and width of the whole rice grain.

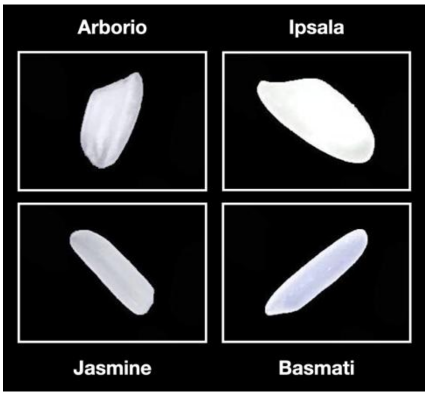

Figure 3. Comparing the similarities of Arborio versus Ipsala and Jasmine versus Basmati Rice Grains, Indicative that Cropping augmentation should be excluded

The use of cropping techniques will result in a loss of scale context, as the model will not have a standard measurement to compare one variety against another. The loss of spatial information creates visual ambiguities where different varieties look similar if only a small part of their area is captured, which is confusing for the classification algorithm and eventually leads to lower generalisation accuracy in the field.

3.3. Application Deployment

The main goal of this work is to focus on the native Apple Swift development workflow, not on external neural network comparisons. Running frameworks like PyTorch or TensorFlow Lite on iOS can often mean complicated bridging or third-party tools. Contrary to that, this study shows that Image Feature Print V1 works natively and seamlessly in the macOS and iOS ecosystem. The resulting .mlmodel file was deployable and was directly integrated into a proof of concept iOS application entirely built on native components: SwiftUI for user interface design, AVFoundation for high-quality imaging and video recording, VisionKit for image preprocessing and segmentation, and Core ML for on-device inference. The entire computer vision workflow for the classification of rice grains is simplified to native components, allowing for the real-time inspection of rice grain types on mobile devices like an iPhone or iPad. This greatly reduces reliance on lab infrastructure or cloud hyperscalers and SaaS, giving convenience and full offline verification to supply chain operators, including consumers. Therefore, benchmarking with baseline architectures like MobileNet or TensorFlowLite is beyond the scope set for this native workflow evaluation.

4. Results and Discussion

The training process was configured to run for a maximum of 50 iterations; however; early monitoring of performance metrics indicated rapid convergence. By the 30th iteration; the model had already achieved a training accuracy of 93.1 percent and a validation accuracy of 97.2 percent; as shown in the training curve generated by Create ML in figure 4 below. This expected discrepancy occurs because aggressive data augmentations such as visual noise and blur during preprocessing are applied exclusively during the training phase to artificially increase the difficulty of the dataset. Consequently; this pattern reflects the model's behavior during training on the baseline dataset. While a stable gap between training and validation metrics indicates proper convergence; a definitive assessment of model robustness would require external cross-dataset validation.

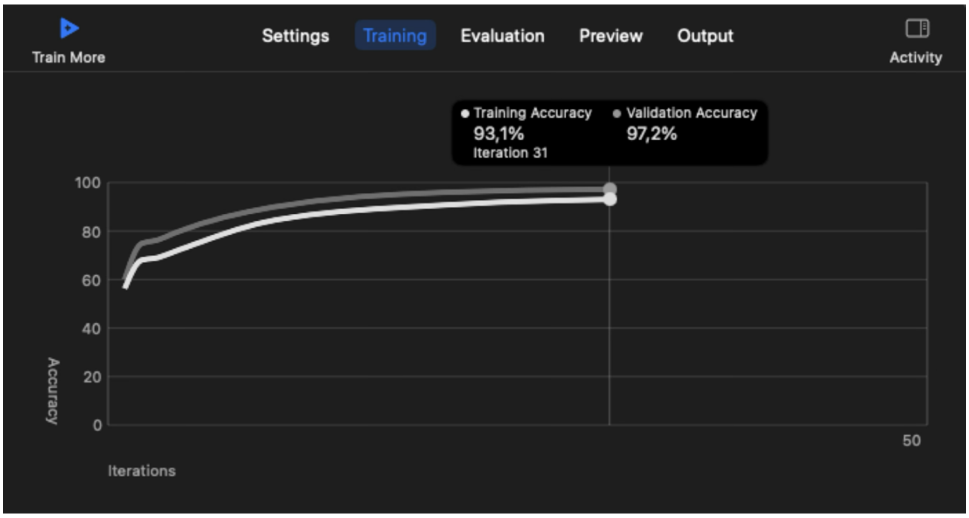

Figure 4. After 31 Iterations, the Create ML model completed training with 93,1% Training Accuracy and 97,2% Validation Accuracy

This early plateau in validation performance reflects the efficiency of the transfer learning approach leveraged by Create ML; while also demonstrating the benefit of high-quality; balanced image data across the five rice grain classes. Furthermore; training was accelerated by Apple Silicon’s Neural Engine on the M4 Pro chip; enabling fast computation per iteration while maintaining energy efficiency. The ability to monitor model performance in real time within the Create ML interface allowed for timely training decisions; avoiding unnecessary computation beyond the optimal convergence point. This balance of speed; accuracy; and robust generalization makes the resulting model highly suitable for real-time mobile deployment using Core ML on iOS devices.

4.1. Model Performance

Finally, the model was evaluated on a previously unseen test set consisting of 15,000 images with equal distribution of 3,000 images of each of the five rice varieties. The model reached testing accuracy of 97 percent, classifying correctly 14,520 images and misclassifying 480 samples. The results show the good generalisation ability of the model outside of the training and validation datasets. As can be seen from the training graph, the learning process stabilised and achieved optimal performance around the 30th iteration, ensuring a balance between learning efficiency and generalisation without significant signs of overfitting. Further results are displayed in table 2.

Table 2. Confusion Matrix of the Create ML Model Trained

Class	Count	True Positives	False Positives	False Negatives	Precision	Recall	F1 Score
Arborio	3000	2871	124	129	96%	96%	0.96

Class							
	Count	True Positives	False Positives	False Negatives	Precision	Recall	F1 Score
Basmati	3000	2907	104	93	97%	97%	0.97
Ipsala	3000	2972	26	28	99%	99%	0.99
Jasmine	3000	2863	125	137	96%	95%	0.96
Karacadag	3000	2907	101	93	97%	97%	0.97

*Confusion Matrix of the Create ML Model Trained showing Excellent F1 Score Ranging between 0.96 and 0.99 for all 5 Classes of Rice Grains.

Detailed analysis of performance for each class showed consistently high precision, recall and F1 scores as can be seen in table 2. The Ipsala class had the most consistent predictions, with 2,972 correct predictions, 26 false positives and 28 false negatives, and precision and recall values of 99 percent. Karacadag and Basmati followed closely behind with 2,907 correct classifications each, maintaining an F1 score above 0.98. The Arborio class performed similarly well with 2,871 correct predictions and an F1 score of 0.98. The Jasmine class was the hardest for the model, with 2,863 correct classifications, 125 false positives and 137 false negatives. This resulted in the lowest precision and recall values among the five classes, but still kept a strong F1 score of 0.96. The highest confusion was between Jasmine and Basmati, with 102 Jasmine images misclassified as Basmati, indicating some visual similarities between samples of these two varieties that may benefit from further data augmentation or feature refinement in future iterations.

The deployment overhead is very small as the custom trained classification layer distributed in the application package is only 66 KB. Traditional architectures such as MobileNetV2 require the entire standalone neural network (13 to 15 MB) to be packed into the binary. The proposed system exploits the native system-level feature extraction infrastructure, decoupling only the top-level classification layer. This modular decoupling decreases the initial download package and structural complexity of the application. The high-test accuracy of the model, coupled with consistent class-level performance metrics, provides a strong foundation for mobile deployment overall. Additional testing in other real-world settings is needed, but these initial results validate the Create ML training pipeline and the feasibility of on-device inference with Core ML in an iOS application.

4.2. Error Analysis

An extensive analysis of the misclassification behaviour of the model shows that the most frequent cause of errors was between the Jasmine and Basmati classes. Specifically, the model misclassified 102 samples of Jasmine as Basmati indicating an overlap in the visual representations learned for these two rice varieties. Despite the misclassification being insignificant with respect to the whaoverall sample size and not materially affecting the model's high accuracy, it is still worth mentioning because of its consistency and underlying reason. Jasmine and Basmati are both long-grain, thin-profiled rices with similar aspect ratios. They can appear nearly identical in 2D images, especially when the lighting and perspective are the same.

This classification challenge indicates that while the model is successful in learning general morphological features such as shape and size, it is not able to classify classes that differ only by subtle textural or proportional cues. The model's performance on more visually distinct varieties such as Ipsala or Karacadag shows that it can generalise well in the presence of strong differences between classes. However, to distinguish Jasmine from Basmati, it likely needs a more sophisticated understanding of fine-grained spatial features, such as subtle variations in the surface texture or micro-patterns that may not be sufficiently represented in the current training data.

This misclassification of the two varieties persists and suggests that there is room for further improvements, such as increasing the image resolution, collecting data under different lighting and orientation conditions, or using more advanced model architectures with attention mechanisms that can detect localised differences. It also opens the door for the exploration of multimodal approaches that integrate image data with other rice grain characteristics (e.g., weight, density, or infrared profile) to improve classification in practical agricultural or industrial applications.

In conclusion, while the confusion between Jasmine and Basmati was statistically insignificant in relation to the overall model performance, it offers a meaningful understanding of the limitations of image-based classification in the context of highly similar object classes. Resolving these nuanced cases will be important to further improve model precision for expert-level grain quality control or large-scale automated sorting systems.

4.3. Application Implementation and Performance

As seen in figure 5 and figure 6, the trained Core ML model was successfully integrated into a SwiftUI-based iOS application. The application offers a simple interface to the user to select an image and get a nearly instant classification.

ISSN 2723-6471

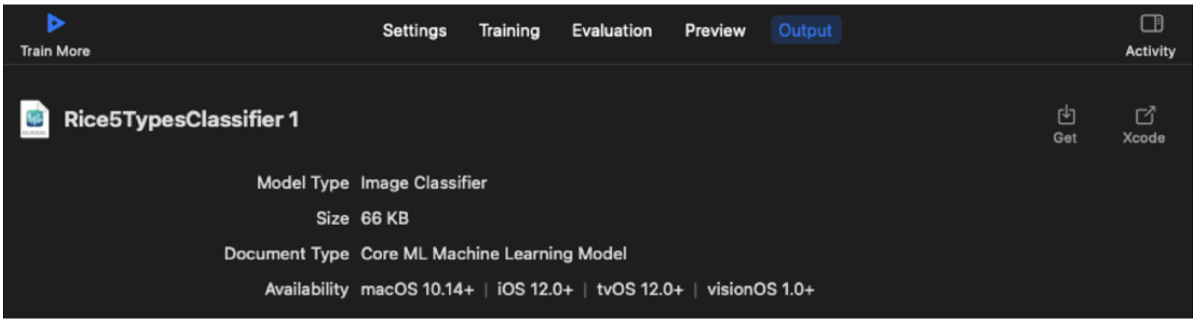

Figure 5. Core ML Model in Overview, Ready for Deployment on Mobile Apps

The application reported a high level of confidence in its predictions. When tested with images from the test set, the confidence scores for correct classification were consistently high across all classes: Arborio achieved 99.70% confidence; Basmati achieved 99.85% confidence; Ipsala achieved 99.99% confidence; Jasmine achieved 99.09% confidence; and Karacadag achieved 99.96% confidence.

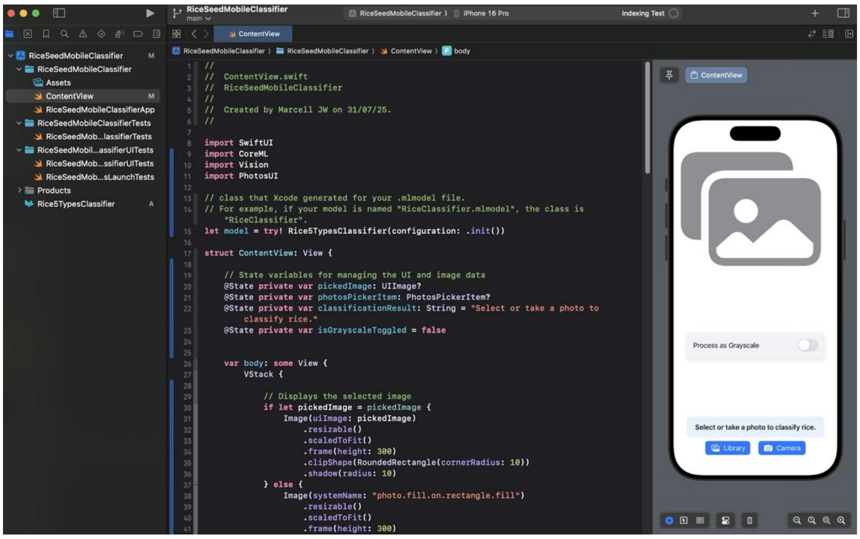

Figure 6. Snippet of Xcode Environment for Developing Simple SwiftUI Application to Classify Rice Grain Images

The high scores indicate that the model is not only accurate but also strong in its predictions, a key feature for a practical utility tool. Below the sample examples of testing all 5 rice grains on the SwiftUI application shown in figure 7.

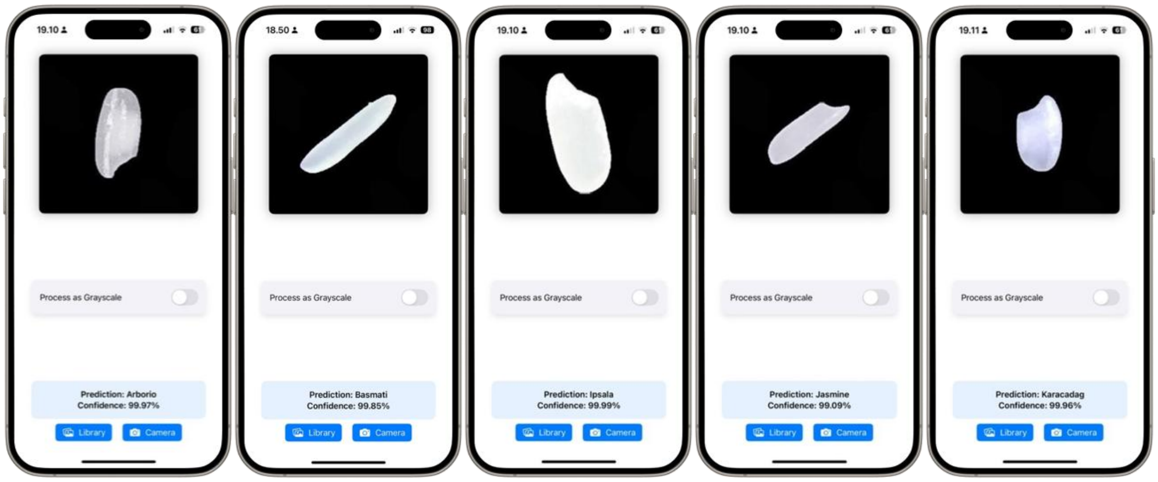

Figure 7. Sample Examples of Testing all 5 Rice Grains on the SwiftUI Application

For real-time application performance evaluation, the deployment was benchmarked natively on an iPhone 15 using Xcode Instruments. Profiling revealed that the embedded payload size for the compiled custom model layer is only 66

KB. However, traditional edge computing pipelines or TensorFlow Lite implementations need to distribute a full 13 to 15 MB model layout into the compiled application package. Common native infrastructure is leveraged for an effective reduction of localised binary overhead of ~99.5%. This compact footprint directly affects the execution efficiency and results in a median on-device prediction time of only 2,75ms per image, an initial load time of 0,54 ms and a compilation time of 3,66 ms as shown in figure 8 below. The profiling data confirms that the native Core ML framework easily fits within on-device cache limits, avoids multi-step preprocessing latency, and facilitates fast load times required for real-time, field-based agricultural inspection.

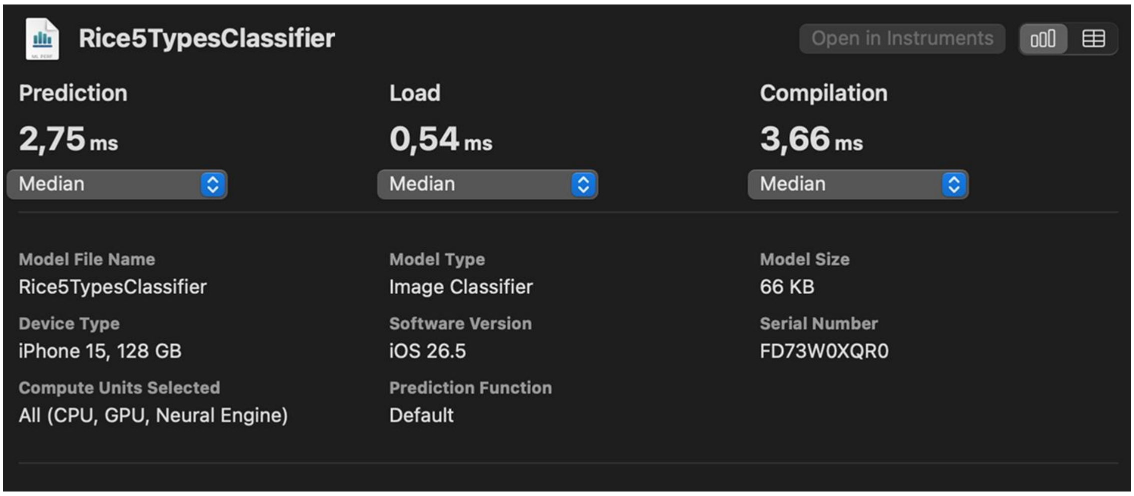

Figure 8. iPhone Model Performance Evaluation from .mlperf tool in the Xcode IDE

Direct profiling of battery consumption is beyond the scope of this proof-of-concept, but the application design is optimised for lightweight operation. We limit the unique footprint to a 66 KB weight layer, ensuring extremely fast and resource-efficient ongoing maintenance and over-the-air (OTA) updates, avoiding the bandwidth and storage expenses of redeploying monolithic, multi-megabyte model binaries.

4.4. Implementation Constraints and Future Research

Future versions can become multimodal by integrating seamlessly with external high-precision IoT peripherals through the native iOS CoreBluetooth and Bluetooth Low Energy (BLE) APIs. For instance, with the addition of external digital micro-scales or macro-lens attachments, the mobile app would be able to directly ingest granular physical and weight data and generate a fully integrated, on-device evaluation pipeline that exceeds the native optical limitations of a smartphone.

5. Conclusion

The development of this proof-of-concept application has important practical implications, especially given recent concerns about food security and market transparency highlighted by the 2025 Indonesian blended rice scandal. Recent studies have revealed that deep learning has been becoming more and more important in food defect detection and quality assurance. For example, it has been applied to detect unsound wheat kernels, indicating the increasing importance of AI-based visual inspection for agricultural and food commodities [21]. Building on this direction, the most important contribution of this work is the validation of a highly efficient deployment pipeline that has the potential to democratise future rice authenticity verification.

By embedding a small, high-accuracy image classification model into a native iOS mobile application, we enable varietal assessment right on the device, without the need for centralised lab tests and cloud dependencies. Previous work has shown that deep learning models can be deployed as real-time applications on smartphones and evaluated using practical metrics such as throughput, frame rate, and device-level resource consumption [22]. Moreover, it has been shown that mobile edge computing can facilitate low latency processing of visual data by bringing computation closer to the point of data acquisition, in-line with the goal of enabling fast on-site inspection [23].

The app enables inspectors, distributors and even end consumers to quickly check rice varieties on-site using a mobile device. Such a capability enhances traceability along the supply chain and enables early detection of adulteration or mislabelling. The solution provides an easy first line of defence and encourages proactive quality control rather than just reactive enforcement mechanisms. Similar smartphone-based deep learning applications have shown the possibility of real-time inspection in real world environment, including defect detection in buildings and road distress detection in mobile phones [24], [25]. Moreover, by running the model on-device, the app can still be used in remote or low connectivity areas, which is a key requirement in agriculture. Local processing also improves privacy, since image data do not have to be sent to a cloud server. This is in line with edge-AI studies showing that mobile on-site processing can reduce cloud dependency, latency, bandwidth requirements, and privacy concerns [25], [26].

Even though the results are promising, the current study has a number of limitations that suggest directions for future research. Initially, the classification model was trained on images captured on a uniform black background. Though this controlled environment is suitable to evaluate the technical feasibility, it is not fully representative to the real-world inspection conditions where rice grains can appear on complex and diverse backgrounds. Second, the dataset used in this study is made up of internationally recognised rice varieties and does not yet represent the full diversity of rice grown and consumed in Indonesia. This limitation is intentional as the main aim of the study is to create a baseline, showing that a high accuracy mobile rice classification application can be developed using the present Core ML framework.

Future research should therefore aim to build a comprehensive dataset of local Indonesian rice grain varieties under more realistic environmental conditions. Inclusion of region-specific varieties would be a substantial improvement in the practical relevance of the application for domestic food monitoring and regulatory enforcement. These datasets combined with background agnostic pre-processing techniques with VisionKit and Core Image would enable retraining or fine-tuning the model while keeping the same Core ML deployment pipeline. Based on this foundational framework, the application can be further developed to incorporate locally sourced data and real-world applications, becoming a scalable and context-sensitive instrument to promote food security, quality assurance, and market integrity in Indonesia.

6. Declarations

6.1. Author Contributions

Conceptualization: D.H., P., and L.T.; Methodology: P.; Software: D.H.; Validation: D.H., P., and L.T.; Formal Analysis: D.H., P., and L.T.; Investigation: D.H.; Resources: P.; Data Curation: P.; Writing Original Draft Preparation: D.H., P., and L.T.; Writing Review and Editing: P., D.H., and L.T.; Visualization: D.H.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

This study would not have been complete without the invaluable contribution of Marcell Jeremy Wiradinata, who developed the mobile application employed in this research. Utilizing SwiftUI and the iOS development framework, he engineered a robust and intuitive platform capable of rapidly scanning and identifying rice grain types with considerable accuracy. This research was financially supported by Universitas Ciputra Surabaya through institutional research funding. Generative AI tools such as ChatGPT and Grammarly were used for grammatical checking and paraphrasing.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] K. Giannakas and A. Yiannaka, "Food fraud: Causes, consequences, and deterrence strategies," *Annu. Rev. Resour. Econ.*, vol. 15, no. October, pp. 85-104, 2023, doi: 10.1146/annurev-resource-101422-013027.

[2] D. T. Bui, T. N. Minh, and L. V. Anh, "Preserving the authenticity of ST25 rice (*Oryza sativa*) from the Mekong Delta: A multivariate geographical characterization approach," *Stresses*, vol. 3, no. 3, pp. 1-12, 2023, doi: 10.3390/stresses3030045.

[3] A. Kusumawardani, B. S. Laksmono, L. Setyawati, and T. E. B. Soesilo, "A policy construction for sustainable rice food sovereignty in Indonesia," *Potravinarstvo Slovak J. Food Sci.*, vol. 15, no. May, pp. 484–496, 2021, doi: 10.5219/1533.

[4] C. A. Glenda and R. Sanjaya, "Consumer perception and purchase decision amid rice adulteration issue: A case study in Bandung, Indonesia," *J. Fokus Manaj.*, vol. 5, no. 4, pp. 939-950, 2025, doi: 10.37676/jfm.v5i4.9844.

[5] E. Maniari, S. M. D. Hutabarat, and M. Sakti, "Business actor's liability and government supervision on the practice of adulterated rice under consumer protection law: Case study of PT Padi Indonesia Maju," *J. Law Polit. Humanit.*, vol. 6, no. 4, pp. 2842–2852, 2026, doi: 10.38035/jlph.v6i4.3348.

[6] S. Huang, Z. Lu, Y. Shi, L. Hu, W. Yang, and C. Huang, "A novel method for filled/unfilled grain classification based on structured light imaging and improved PointNet++," *Sensors*, vol. 23, no. 14, pp. 1-14, 2023, doi: 10.3390/s23146331.

[7] M. J. Iqbal, M. Aasem, I. Ahmad, M. O. Alassafi, S. T. Bakhsh, N. Noreen, and A. Alhomoud, "On application of lightweight models for rice variety classification and their potential in edge computing," *Foods*, vol. 12, no. 21, pp. 1-16, 2023, doi: 10.3390/foods12213993.

[8] P. Jalaparthi, M. Shabbeer, S. Kumar, N. Samala, and A. Hussain, "A fast and intelligent rice quality assessment by combining CNN and Bi-LSTM for accurate rice quality evaluation," *AIP Conf. Proc.*, vol. 3342, no. 1, pp. 1-8, 2025, doi: 10.1063/5.0296679.

[9] K. Kiratiratanapruk, P. Temniranrat, W. Sinthupinyo, P. Prempree, K. Chaitavon, S. Porntheeraphat, and A. Prasertsak, "Development of paddy rice seed classification process using machine learning techniques for automatic grading machine," *J. Sensors*, vol. 2020, no. 1, pp. 1-14, 2020, doi: 10.1155/2020/7041310.

[10] M. Koklu, I. Cinar, and Y. S. Taspinar, "Classification of rice varieties with deep learning methods," *Comput. Electron. Agric.*, vol. 187, no. August, pp. 1-8, 2021, doi: 10.1016/j.compag.2021.106285.

[11] R. Singh and S. Chaudhury, "A cascade network for the classification of rice grain based on single rice kernel," *Complex Intell. Syst.*, vol. 6, no. 2, pp. 321–334, 2020, doi: 10.1007/s40747-020-00132-9.

[12] P. Nagamani, S. Yacoob, G. L. S. Sneha, S. Mahesh, and B. Godavarthi, "SmartRiceQC: Integrating CNN and Bi-LSTM for precise rice quality analysis," in *Smart Computing Paradigms: Advanced Data Mining and Analytics*, Singapore: Springer Nature, vol. 2025, no. May, pp. 267–277, 2025, doi: 10.1007/978-981-96-1981-8_21.

[13] F. Farahnakian, J. Sheikh, F. Farahnakian, and J. Heikkonen, "A comparative study of state-of-the-art deep learning architectures for rice grain classification," *J. Agric. Food Res.*, vol. 15, no. March, pp. 1-10, 2024, doi: 10.1016/j.jafr.2023.100890.

[14] R. Guerra, B. Angira, M. Kongchum, and A. N. Famoso, "Phenotypic and genotypic characterization of grain quality in US and Latin American rice and implications for breeding," *J. Sci. Food Agric.*, vol. 105, no. 13, pp. 7159–7168, 2025, doi: 10.1002/jsfa.14428.

[15] T. T. K. Nga, T. V. Pham, D. M. Tam, I. Koo, V. Y. Mariano, and T. Do-Hong, "Combining binary particle swarm optimization with support vector machine for enhancing rice varieties classification accuracy," *IEEE Access*, vol. 9, no. April, pp. 66062–66078, 2021, doi: 10.1109/ACCESS.2021.3076130.

[16] C. Kurade, M. Meenu, S. Kalra, A. Miglani, B. C. Neelapu, Y. Yu, and H. S. Ramaswamy, "An automated image processing module for quality evaluation of milled rice," *Foods*, vol. 12, no. 6, pp. 1-14, 2023, doi: 10.3390/foods12061273.

[17] R. Setiawan and H. Oumarou, "Classification of rice grain varieties using ensemble learning and image analysis techniques," *Indones. J. Data Sci.*, vol. 5, no. 1, pp. 54–63, 2024, doi: 10.56705/ijodas.v5i1.129.

[18] C. Thangavel and D. Sakthipriya, "Machine learning ensemble classifiers for feature selection in rice cultivars," *Appl. Artif. Intell.*, vol. 38, no. 1, pp. 1-25, 2024, doi: 10.1080/08839514.2024.2394734.

[19] H. Zareiforoush, S. Minaee, M. Alizadeh, and A. Banakar, "Potential applications of computer vision in quality inspection of rice: A review," *Food Eng. Rev.*, vol. 7, no. January, pp. 321–345, 2015, doi: 10.1007/s12393-014-9101-z.

[20] Y. Zhang, X. Xing, L. Zhu, X. Li, X. Y. Xue, C. T. Sun, C. H. Duan, and Y. P. Du, "Rice quality recognition method based on multimodal model," *Food Control*, vol. 181, no. March, pp. 1-14, 2026, doi: 10.1016/j.foodcont.2025.111691.

[21] R. Wang, Z. Zhao, Y. Sun, Y. Ni, J. Ye, M. Zhang, and L. Cui, "Food defect detection technologies based on deep learning and prospects in detection of unsound wheat kernels," *Food Chem.*, vol. 496, no. December, pp. 1-16, 2025, doi: 10.1016/j.foodchem.2025.146910.

[22] A. Sehgal and N. Kehtarnavaz, "Guidelines and benchmarks for deployment of deep learning models on smartphones as real-time apps," *Mach. Learn. Knowl. Extr.*, vol. 1, no. 1, pp. 450–465, 2019, doi: 10.3390/make1010027.

[23] H. Trinh, D. Chemodanov, S. Yao, Q. Lei, and S. Yan, "Energy-aware mobile edge computing for low-latency visual data processing," in *Proc. 2017 IEEE 5th Int. Conf. Future Internet Things Cloud (FiCloud)*, vol. 2017, no. November, pp. 128-133, 2017, doi: 10.1109/FiCloud.2017.13.

[24] H. Perez, J. H. M. Tah, and A. Mosavi, "Deep learning smartphone application for real-time detection of defects in buildings," *Struct. Control Health Monit.*, vol. 28, no. 7, pp. 1-15, 2021, doi: 10.1002/stc.2751.

[25] Y. Hu, N. Chen, Y. Hou, X. Lin, B. Jing, *et al.*, "Lightweight deep learning for real-time road distress detection on mobile devices," *Nat. Commun.*, vol. 16, no. May, Art. no. 4212, pp. 1-13, 2025, doi: 10.1038/s41467-025-59516-5.

[26] H. I. Ustun, M. Bulbul, G. Yolcu Oztel, and V. H. Sahin, "On-device brain tumor classification from MR images using smartphone," *Adv. Intell. Syst.*, vol. 8, no. 1, Art. no. e2500205, pp. 1-13, 2026, doi: 10.1002/aisy.202500205.