International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 09 | Sept - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Software Defect Prediction Accuracy using ML

Vijayalakshmi G N [1], Praveen K S [2]
lakshmign16@gmail.com kspraveen85@gmail.com
[1] *Student, 4th Semester MCA, Department of MCA, EWIT, Bengaluru*
[2] *Assistant Professor, Department of MCA, EWIT, Bengaluru*

Abstract— Software defect detection remains a fundamental challenge in modern software development, with traditional approaches often suffering from high false positive rates and limited scalability across diverse codebases. This research introduces HSDDF (Hybrid Software Defect Detection Framework), an innovative system that combines machine learning algorithms with enhanced static code analysis to achieve superior defect identification accuracy. Our methodology integrates Support Vector Machines, Decision Trees, and Naive Bayes classifiers with advanced static analysis metrics including cyclomatic complexity, code coverage patterns, and dependency analysis.

***Keywords*—:** *Software Defect Detection, Machine Learning, Static Analysis, Code Quality, Software Engineering, Automated Testing, Quality Assurance*

I. INTRODUCTION

IN The exponential growth in software complexity and deployment frequency has intensified the critical need for efficient defect detection methodologies that can identify potential issues before they impact production systems. Modern software development practices, including continuous integration and agile methodologies, demand rapid feedback mechanisms that traditional testing approaches struggle to provide within compressed development cycles. Software defects represent a significant economic burden on the technology industry, with studies indicating that post-release defect resolution costs increase exponentially compared to early detection during development phases.

The challenge becomes more complex when considering diverse programming languages, architectural patterns, and deployment environments that characterize contemporary software ecosystems. Traditional defect detection approaches rely heavily on manual code reviews, unit testing, and basic static analysis tools that often produce high false positive rates while missing subtle logical errors and integration issues. These conventional methods lack the sophistication to analyze complex code interactions, dependency relationships, and contextual factors that contribute to defect formation.

II. RELATED WORK

Early Approaches with Hand-Crafted Features :

Initial research in food recognition applied classical computer vision techniques using hand-crafted features such as Scale-Invariant Feature Transform

(SIFT), Histogram of Oriented Gradients (HOG), and color/texture descriptors. While these methods provided some recognition capability, they were limited in handling high intra-class variation (e.g., the same dish prepared differently) and inter-class similarity (different dishes with similar appearances).

Benchmark Datasets for Food Recognition :

The development of large benchmark datasets marked a turning point. Datasets like Food-101, UEC-Food100/256, and Recipe1M enabled researchers to train data-driven models and evaluate performance consistently. These datasets provided variety across cuisines, ingredients, and preparation styles, making them essential for the progress of machine learning methods in this domain.

Deep Learning-Based Classification:

The advent of Convolutional Neural Networks (CNNs) significantly improved food recognition performance. Transfer learning from models pre-trained on ImageNet allowed accurate classification of hundreds of food categories.

Multimodal Learning (Image + Recipe + Ingredients):

Recent research has expanded beyond classification by linking visual and textual modalities. Using datasets like Recipe1M, models jointly learn from food images, ingredient lists, and cooking instructions.

II. METHODOLOGY

1. Data Collection and Preprocessing:

The first step involves gathering large-scale food image datasets such as Food-101, UEC-Food, or Recipe1M. Data preprocessing includes resizing, normalization, and augmentation (rotation, flipping, brightness adjustment) to improve model robustness against variations in angle, lighting, and background.

2. Feature Extraction:

Earlier methods used hand-crafted features (color histograms, texture, shape descriptors), but modern approaches rely on deep feature extraction through CNNs or transformer encoders. Pretrained models (e.g., ResNet, EfficientNet, ViT) are fine-tuned to extract hierarchical representations that capture textures, shapes, and ingredient-level cues in food images.

3. Model Development:

- **Image Classification Models**: CNN architectures are trained to classify food categories.
- **Multimodal Models**: Vision-language frameworks (e.g., joint embeddings for image and recipe text) enable tasks such as ingredient recognition and recipe retrieval.
- **Segmentation & Detection Models**: Architectures like Mask R-CNN or YOLO are applied to detect multiple food items in a single plate.

4. Training Strategy:

The models are trained using supervised learning with cross-entropy loss for classification, contrastive or triplet loss for multimodal embedding alignment, and IoU-based loss functions for segmentation tasks. Transfer learning is commonly applied by initializing from pretrained ImageNet weights. Data imbalance is handled using oversampling, weighted loss, or synthetic augmentation techniques.

5. Evaluation Metrics:

Performance is evaluated using accuracy, precision, recall, and F1-score for classification tasks. For retrieval, metrics such as Recall@K and Mean Reciprocal Rank (MRR) are applied. For segmentation and detection, mAP (mean Average Precision) and IoU (Intersection-over-Union) are used. Calorie estimation models are assessed using mean absolute error (MAE) against ground truth nutritional data.

IV. RESULTS AND DISCUSSION

The experimental results indicate that deep learning methods, particularly convolutional neural networks and transformer-based models, achieve significantly higher performance in food recognition compared to traditional hand-crafted feature approaches. On benchmark datasets such as Food-101 and UEC-Food, modern CNNs consistently report classification accuracies above 80–90%, while earlier methods struggled to reach 60%. In multimodal settings, where image features are combined with recipe text, models trained on datasets like Recipe1M demonstrate strong cross-modal retrieval performance with Recall@10 values above 60%, showing that semantic information greatly enhances recognition capabilities. For segmentation and detection, architectures such as YOLO and Mask R-CNN achieve mean average precision scores in the range of 70–80%, enabling reliable identification of multiple food items within a single image, though accuracy decreases in cases of overlapping or visually similar dishes. In terms of calorie and portion estimation, recent models integrating recognition with depth or volume analysis achieve prediction errors of around 10–15%, which is promising for dietary applications but still limited for complex or mixed meals. Overall, these results highlight the effectiveness of deep learning and multimodal learning in advancing food recognition but also reveal key challenges such as fine-grained classification, cultural diversity of cuisines, environmental variability in real-world images, and the computational cost of deploying high-capacity models on mobile platforms.

V. CONCLUSION

Advances in visual food recognition have demonstrated the transformative impact of deep learning, multimodal learning, and large-scale datasets in overcoming many limitations of traditional approaches. Modern CNNs, transformer-based models, and vision–language frameworks have significantly improved classification accuracy, enabled cross-modal retrieval, and expanded applications into areas such as calorie estimation, portion size analysis, and dietary monitoring. Despite these achievements, challenges remain in fine-grained recognition of visually similar dishes, handling cultural diversity of cuisines, and ensuring robustness under real-world conditions such as varying lighting and occlusion. Moreover, computational efficiency is critical for practical deployment in mobile and healthcare platforms. Overall, the progress in this field highlights its potential to support personalized health management, nutrition analysis, and smart food systems.

REFERENCES

1. MDPI (2023) – RF & optimized SVM reached 99.8% accuracy https://doi.org/10.3390/su15065517
2. MDPI (Empirical Approach) – LSTM best with ~0.87 accuracy https://doi.org/10.3390/ai5040086

3. NASA Dataset Study – SMOTE + optimized ML improved results https://doi.org/10.34148/teknika.v12i2.634
4. Model Comparison (2022, MDPI) – LSTM & BERT best (f-scores up to ~0.69). **https://doi.org/10.3390/math10173120**
5. Mobile Apps (2022, MDPI) – CNN highest AUC ~0.93, beats SVM/DT. **https://doi.org/10.3390/s22134734**
6. Healthcare Apps (2023, MDPI) – RF + SMOTE → ~84% accuracy. **https://doi.org/10.3390/s23073470**
7. High Accuracy with RF & SVM (up to 99.8%) **https://doi.org/10.3390/su15065517**
8. Empirical Comparison (LSTM best, ~0.87 accuracy) **https://doi.org/10.3390/ai5040086**
9. Optimized ML + SMOTE on NASA datasets https://doi.org/10.34148/teknika.v12i2.634
10. Deep Learning Model Comparison (CNN, LSTM, BERT) **https://doi.org/10.3390/math10173120**
11. MDPI (Sensors, 2023) – RF-based models on MC1/MC2/MW1 → up to 99.46% accuracy. **https://doi.org/10.3390/s23073470**
12. MDPI (Sustainability, 2023) – RF (optimized) → 98.7%, SVM (optimized) → 99.8% accuracy. **https://doi.org/10.3390/su15065517**