Mushroom Classification Using Handcrafted Features: Evaluating the Impact of Image Processing

Thi-Thu-Hong Phan *

Tuan-Minh Do

Ngoc-Bao-Quang Nguyen

Nam-Thuan Trinh

Artificial Intelligence Department, FPT University, Da Nang, Vietnam

March 19, 2025

*Corresponding author

Email address: hongptt11@fe.edu.vn (Thi-Thu-Hong Phan)

Contents

Abstract

This study presents a robust framework for mushroom species classification, addressing the need for precise identification in food safety and mycological research. Two datasets with varying background preprocessing levels are analyzed to assess classification performance. A diverse set of handcrafted features, including GIST, LBP, GLCM, Hu Moments, HSV/BGR histograms, and color-based features, are extracted, with Random Forest and LightGBM employed for feature selection. Traditional machine learning models, such as Decision Tree, Random Forest, LightGBM, XGBoost, CatBoost, and Extra Trees, are evaluated. Results show that Extra Trees achieves 93.08% accuracy on the fully preprocessed dataset, while LightGBM performs best on the less refined one with 63.33% accuracy, highlighting the impact of preprocessing quality. While the overall accuracy remains limited, these findings provide insights into the challenges of mushroom classification and suggest directions for improving feature extraction and model performance.

Keywords: Mushroom classification, handcrafted feature extraction, machine learning, Random Forest, LightGBM, XGBoost, CatBoost, Extra Trees, Decision Tree, multi-class classification.

1 Introduction

Mushrooms play a crucial role in ecological systems and have significant culinary, medicinal, and economic value. However, the accurate identification of mushroom species remains a challenging task due to their high morphological variability and the subtle differences between species. Traditional identification methods, which rely on expert judgment and manual observation, are not only time-consuming but also prone to inconsistencies and human error.

The advent of computer vision and machine learning has paved the way for automated classification systems that offer a more objective and efficient alternative. In this study, we propose a novel framework for mushroom classification that leverages a robust image preprocessing pipeline, advanced feature extraction, and embedded feature selection methods, followed by the application of state-of-the-art machine learning algorithms.

Our approach starts with a sophisticated background removal process designed to isolate the mushroom from distracting elements in the image. This is followed by the extraction of discriminative features that capture both global and local morphological characteristics. To ensure that only the most relevant attributes contribute to the classification, embedded feature selection techniques are applied to reduce dimensionality and

enhance model interpretability.

For the classification task, we evaluate the performance of six machine learning models: LightGBM, XGBoost, CatBoost, Random Forest, Extra Trees, and Decision Tree. These models were chosen for their proven ability to handle high-dimensional data and to model complex, non-linear relationships that are inherent in mushroom images.

The subsequent sections of this paper provide a comprehensive overview of the research process. The methodology, including data collection, preprocessing techniques, feature extraction, and model training, is described in detail. Following this, the experimental results and comparative analysis of the models are discussed. Lastly, the conclusion summarizes the key findings and suggests directions for future research.

2 Related work

In recent years, various methodologies have been explored for mushroom classification, leveraging both traditional machine learning techniques and advanced deep learning models. Ottom et al. [1] conducted a study employing machine learning techniques for mushroom classification. They extracted eigenfeatures and histogram features from images, incorporating dimensional information such as cap diameter and stem dimensions. The study utilized algorithms like Decision Trees, k-Nearest Neighbors (k-NN), and Support Vector Machines (SVM), achieving an accuracy of 72% with the Decision Tree classifier. Preechasuk et al. [2] developed a Convolutional Neural Network (CNN) model to classify 45 types of mushrooms, aiming to enhance the safety of mushroom consumption. Their approach achieved precision, recall, and F1 scores of 0.78, 0.73, and 0.74, respectively. Another study proposed a CNN model for classifying 103 mushroom species, utilizing images captured in natural habitats. The model achieved an accuracy of 96.70%, with high precision, recall, and F1 scores for each class, marking a 4.4% improvement over previous approaches. However, deep learning models typically require large annotated datasets and substantial computational resources. Given the scope of this study, we explore traditional machine learning (ML) approaches, leveraging handcrafted features such as Color histograms, Local Binary Patterns (LBP), and Gray-Level Co-occurrence Matrix (GLCM), Hu moments,... . Inspired by ML-based studies, we ensure robust feature extraction techniques to enhance classification performance while maintaining

interpretability and computational efficiency.

3 Methodology

The aim of this study is to identify Mushroom images and classify it into multi categories using machine learning techniques. Our approach consists of several key stages: **preprocessing**, **data augmentation**, **feature extraction and combination**, **model training**, and **performance evaluation**. The overall workflow is illustrated in Figure 1.

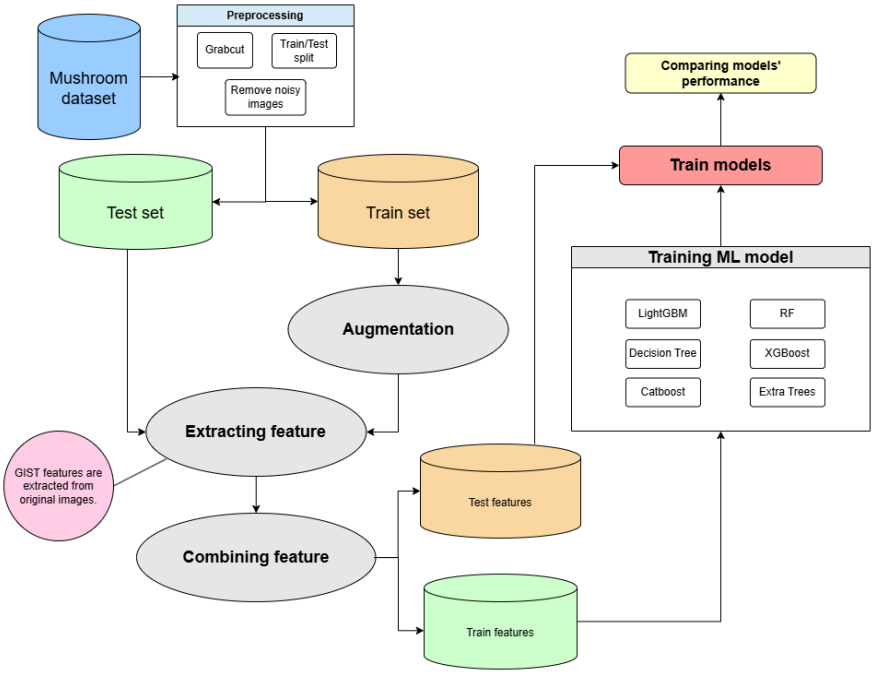

Figure 1: Overview of the workflow.

3.1 Data preprocessing

In mushroom classification tasks, removing the background is essential to eliminate irrelevant noise and highlight the key features of the mushroom, which are critical for accurate species identification.

- **Foreground Extraction with GrabCut:** A bounding box is initialized around the region containing the mushroom. The GrabCut algorithm [3] iteratively refines foreground and background models, effectively isolating the mushroom from its surroundings and discarding non-essential background details.

- **Object Cropping:** After background removal, the image is converted to a binary mask to identify the primary object—the mushroom. The largest contour, corresponding to the mushroom, is detected and used to crop the image, ensuring that only the significant area (the mushroom) remains for further feature extraction and classification.

By isolating the mushroom and discarding extraneous background information, this approach reduces noise and improves the accuracy of subsequent feature extraction and classification stages.

3.2 Feature Extraction

Feature extraction plays a critical role in transforming raw image data into meaningful numerical representations. Extracting features such as texture, shape, and color distribution enables the machine learning models to capture the key morphological characteristics of mushrooms, thereby distinguishing between different species, even when the differences are subtle. The resulting feature set not only reduces the complexity of the original data but also enhances the accuracy and robustness of the classification system. Based on the unique characteristics of each dataset, features are optimally selected to best match the nature of the data and the specific requirements of the classification task.

- **GIST Descriptor:** The GIST descriptor, developed by Oliva and Torralba [6], is designed to capture the overall spatial structure of an image rather than focusing on local details. This makes it particularly useful for mushroom classification, where the general shape and layout of the mushroom can be more informative than fine details. The feature extraction process begins by converting the image to grayscale and performing preprocessing to minimize the effects of illumination variations. The image is then decomposed into multiple frequency components using Gabor filters applied in eight different orientations. The final GIST descriptor is derived from the energy spectrum of these filter responses, with dimensionality reduction achieved by averaging over a 4×4 grid for each filter orientation, resulting in a 128-dimensional feature vector.

- **SIFT Descriptor:** Scale-Invariant Feature Transform (SIFT), proposed by Lowe [7], is a robust feature extraction method known for its ability to identify key points

in an image regardless of changes in scale, rotation, or lighting conditions. In the context of mushroom classification, SIFT can help identify distinctive features such as the shape of the cap, surface texture, or stem patterns. The extraction process involves detecting key points using a scale-space approach, assigning a dominant orientation to each point for rotation invariance, and constructing a 128-dimensional feature vector based on the distribution of gradient magnitudes and orientations in the surrounding region. These characteristics make SIFT particularly useful for distinguishing mushroom species with similar appearances but different structural patterns.

- **ORB Descriptor:** Oriented FAST and Rotated BRIEF (ORB), developed by Rublee et al [8], is an efficient alternative to SIFT and SURF, offering high-speed feature detection and description. ORB combines the FAST key point detector with the BRIEF descriptor, modified to account for orientation, ensuring robustness to rotation. In mushroom classification, ORB can be utilized to capture local texture and shape features that differentiate species. The algorithm detects key points using FAST, computes a local orientation based on intensity gradients, and applies a modified BRIEF descriptor to generate a binary feature vector. Due to its computational efficiency and reliable performance, ORB is particularly suitable for real-time mushroom classification applications or implementation on mobile devices.

- **Hu Moment:** Hu Moments, introduced by Hu (1962) [9], consist of seven invariant features that remain unchanged under image scaling, rotation, and translation. In mushroom classification, Hu Moments capture the overall shape characteristics of mushrooms, such as the shape of the cap, stem, and edges. After detecting the contour or object region through segmentation methods, the Hu Moments are computed to form a feature vector that helps distinguish mushrooms with different shapes.

- **Contour Detection:** Contour Detection Approach, popularized by X. Y. Gong, H. Su, D. Xu et al. [10], is used to detect and extract object boundaries in an image. For mushroom classification, morphological features like area, perimeter, convexity, and aspect ratio of the contours are extracted to recognize different mushroom types based on their cap and stem shapes.

- **Color Basics:** As proposed in [11], the basic descriptor aims to capture essential characteristics of rice seeds and is divided into three main categories: i) Morphological features: These features describe the geometric properties of the rice seeds, such as area, length, width, length-width ratio, major axis, minor axis, convex hull area, and convex hull perimeter. ii) Color features: Derived from the color channels of rice grain images, these features include the mean and square root of mean values for the red (R), green (G), and blue (B) channels. iii) Texture features: These features quantify the textural properties of rice grains and include metrics such as mean, standard deviation, uniformity, and third moment.

- **Color Histogram:** Color Histogram, introduced by SP. Liu, J.-M. Guo, K. Chamnongthai and H. Prasetyo [13], describes the distribution of color values in an image, capturing the overall color information of mushrooms. By dividing the color values into bins, the histogram represents the frequency of each color level, making it resistant to image rotation and translation. It is suitable for classifying mushroom types with clear color variations.

- **LBP (Local Binary Pattern):** Local Binary Pattern (LBP), developed by Ojala, Pietikäinen, and Harwood (1996) [12], is a simple yet effective texture descriptor. In mushroom classification, LBP encodes the grayscale intensity differences between a central pixel and its neighboring pixels into a binary value. The LBP histogram is then computed and used as an input feature for the model, helping distinguish mushrooms with different surface patterns or textures.

- **GLCM (Gray Level Co-occurrence Matrix):** GLCM, proposed by Haralick et al. (1973) [14], is based on the co-occurrence matrix of gray levels that describes the spatial relationship between pixel intensities at specific distances and angles. Features like contrast, correlation, energy, and homogeneity are extracted from GLCM to differentiate mushrooms with complex surfaces or fine-grained textures.

3.3 Feature Selection Techniques

Feature selection is a critical preprocessing step in constructing effective classification models, particularly when dealing with high-dimensional datasets such as those encountered in mushroom classification. By identifying and retaining only the most relevant

features, our approach reduces model complexity, minimizes overfitting, and enhances interpretability, while simultaneously lowering computational overhead.

Unlike PCA or Mutual Information, which do not leverage feature importance directly from classification models, embedded methods such as Random Forest and LightGBM allow feature selection to be performed within the training phase. This ensures that the most discriminative features are retained based on their actual contribution to classification performance, rather than relying solely on statistical relationships.

In this study, we employ an embedded feature selection strategy that integrates the selection process directly within the model training phase. This approach ensures that feature importance is evaluated in relation to its impact on classification performance. Specifically, we utilize the following two methods:

- **Random Forest:** This ensemble-based method computes feature importance by measuring the reduction in impurity when a feature is used for splitting. Its ability to naturally handle high-dimensional data and its robustness against overfitting make it well-suited for our dataset. Additionally, Random Forest does not assume feature independence, which is beneficial given the complex relationships among mushroom characteristics.

- **LightGBM:** As a gradient boosting framework optimized for efficiency and scalability, LightGBM assesses feature relevance based on its contribution to the model's predictive accuracy. Compared to other boosting methods, LightGBM is significantly faster while maintaining high accuracy, making it particularly advantageous for large datasets with diverse feature distributions.

The combination of these two methods allows us to balance interpretability and performance. While Random Forest provides a stable and intuitive ranking of features, LightGBM optimally identifies features that enhance predictive power. By leveraging both approaches, we ensure that the selected features effectively capture the intrinsic morphological and spectral characteristics of mushrooms.

In the subsequent sections, we further analyze the impact of feature selection on model performance.

3.4 Machine Learning Models

To classify mushroom species accurately, we employed and evaluated a diverse set of machine learning models, including Random Forest (RF), Extra Trees (ET), LightGBM, XGBoost, CatBoost, and Decision Tree (DT). These models were chosen for their ability to handle high-dimensional, non-linear data while providing reliable classification performance.

- **Decision Tree (DT):**are among the most popular machine learning models for classification tasks[15]. They split data into subsets based on feature values, forming a tree-like structure where each node represents a feature, each branch represents a decision rule, and each leaf represents a class label. The splits are determined using criteria such as Gini impurity or information gain. Decision trees are also foundational components for advanced ensemble methods like RF and gradient-boosting machines.

- **Random Forest (RF):** proposed by Breiman[16], RF is an ensemble learning method that constructs a multitude of decision trees, each trained on a different bootstrap sample of the data. During the construction of each tree, a random subset of features is selected at each node to determine the best split. The chosen split from these random features creates the branches of the decision tree. For predicting new data points, RF aggregates the predictions from all its trees, typically using a majority vote, where the class label predicted by the most trees becomes the final prediction for the new data point.

- **Extra Trees (ET):** is an ensemble learning technique that constructs multiple decision trees during training and combines their predictions for classification or regression tasks[17]. The "extreme" aspect of this method lies in introducing randomness during tree construction, particularly in determining split points. Unlike Random Forests (RF), which rely on bootstrapped replicas, ET uses the entire training dataset to create the trees. Furthermore, ET considers all features but selects a random split point within each subset. This increased randomness in split point selection helps reduce the model's variance, potentially enhancing generalization performance on unseen data.

- **LightGBM:** LightGBM is an optimized gradient boosting method for computational speed and efficiency [18]. The technique constructs weak decision trees, where each new tree is added to correct the errors of previous trees. LightGBM applies optimizations such as Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundle (EFB) to reduce complexity while maintaining high performance. Designed to handle large and numeric data, LightGBM-powered computational analysis increases scalability and improves aggregation efficiency.

- **Extreme Gradient Boosting(XGB):** or XGBoost is an optimized distributed gradient boosting method[19]. This technique involves sequentially adding weak learners (decision trees) to the ensemble, with each new learner addressing the errors made by its predecessors. XGBoost includes regularization techniques to prevent overfitting and supports distributed computing for scalability. Optimized for speed and efficiency, XGBoost is well-suited for handling large datasets.

- **CatBoost (CB):** is a machine learning algorithm that utilizes gradient boosting on decision trees. Unlike traditional methods that require extensive preprocessing to convert categorical features into numerical representations, CatBoost handles categorical features directly, eliminating the need for extensive data preprocessing. It is particularly effective at mitigating overfitting in decision trees, making it a robust option for classification tasks involving complex datasets[20].

3.5 The Proposed Method

In this study, we refine the traditional feature extraction approach to improve classification accuracy by integrating adaptive transformations and optimized parameter tuning. Unlike conventional methods, our approach emphasizes robustness and computational efficiency through targeted enhancements.

- **Optimized Image Augmentation:** To better generalize across variations, we apply controlled augmentation, including horizontal flipping ($p = 0.5$) and rotational transformations ($\pm 10^{\circ}$) with a probability of 0.3.

- **Enhanced Gabor Feature Extraction:** We employ a bank of Gabor filters with kernel sizes of 31, $\sigma = 5$, $\lambda = 10$, $\gamma = 0.5$, and $\psi = 0$. Unlike standard approaches,

our orientation-specific filtering uses an optimized set of angles $[0, \pi/4, \pi/2, 3\pi/4]$, and mean responses are carefully structured into the feature vector to capture more discriminative patterns.

- **Refined GIST Descriptor Computation:** Images are converted to grayscale only if necessary, reducing unnecessary processing overhead. Our approach maintains a 4×4 grid partitioning strategy but refines Gabor filter responses for enhanced spatial feature extraction.

- **GLCM-Based Texture Analysis with Improved Masking:** We compute Gray-Level Co-occurrence Matrix (GLCM) features specifically for masked regions, improving feature relevance. The directional analysis $(0, \pi/4, \pi/2, 3\pi/4)$ and a single pixel distance are retained, but feature extraction is optimized for key texture regions.

- **SIFT Feature Extraction with Adaptive Keypoint Selection:** Unlike standard SIFT implementations, we refine keypoint selection by filtering based on contextual importance. After filtering for meaningful keypoints, their descriptors are averaged to form a 128-dimensional feature vector, ensuring a compact yet discriminative representation.

- **LBP and Color Histogram Features with Region Emphasis:** Local Binary Pattern (LBP) computation is focused on masked regions using $P = 8, R = 1$ to improve texture differentiation. Color histograms in HSV and RGB color spaces (bins=$(8, 8, 8)$) are computed, ensuring color-based discriminative power.

- **Contour-Based Shape Features with Adaptive Thresholding:** Instead of a fixed threshold, we apply adaptive thresholding and morphological operations to extract image contours more effectively. The largest contour is analyzed using shape descriptors such as area, perimeter, compactness, aspect ratio, and contour count, enhancing structural differentiation.

- **Optimized ORB Descriptor Extraction:** ORB (Oriented FAST and Rotated BRIEF) descriptors are extracted within the masked region, but unlike traditional methods, we introduce a refined selection strategy. Feature representation is obtained by averaging the ORB descriptors while ensuring key region emphasis.

These improvements over conventional methods contribute to a more robust feature representation framework, leading to better classification performance with reduced computational complexity.

4 Experiments and results

4.1 Dataset Description

In this study, we use two datasets:

- **Primary Dataset:** The **Mushrooms Classification - Common Genus's Images**, a dataset sourced from Kaggle [4]. This dataset serves as the main source for model training and evaluation.
- **Secondary Dataset:** The **Klasifikasi Jamur Dataset**, an open-source dataset from BINUSBandung, hosted on Roboflow Universe [5]. It is incorporated for comparative analysis.

4.1.1 Mushrooms Classification - Common Genus's Images

The primary dataset consists of **6,714 images** of the most common Northern European mushroom genera, categorized into 9 distinct classes. The dataset includes between 300 to 1,500 images per class, capturing diverse species and environmental conditions.

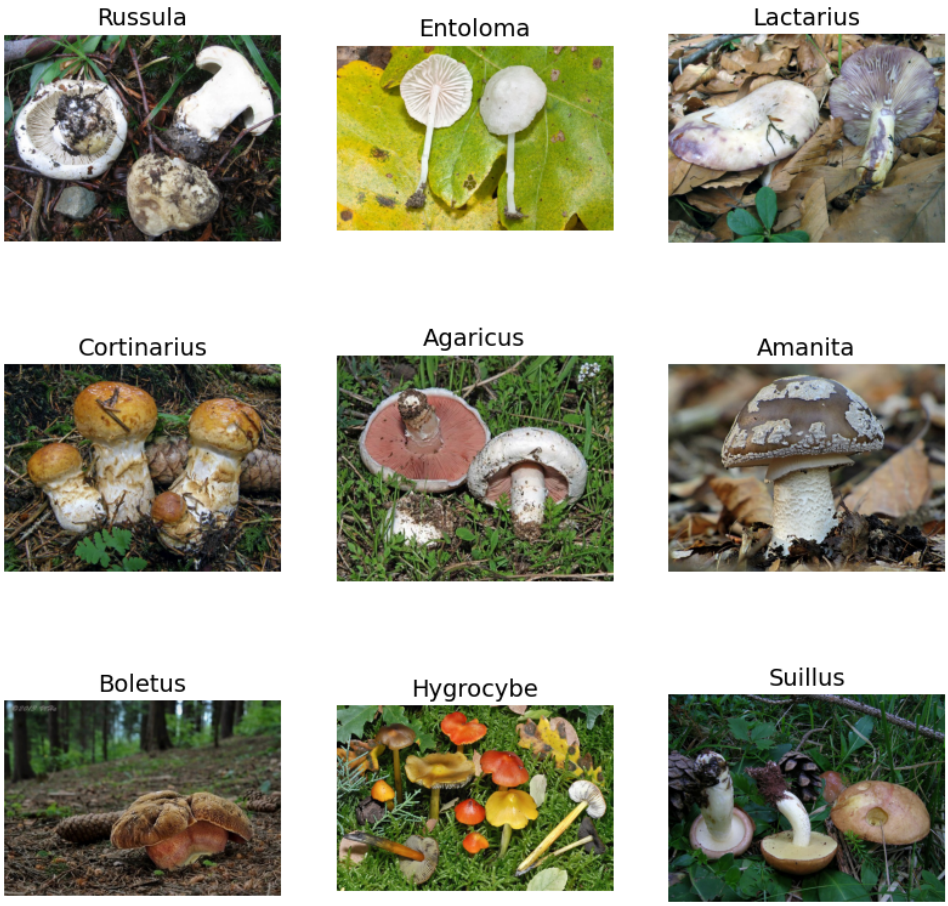

Figure 2: Sample images from the primary dataset

4.1.2 Klasifikasi Jamur Dataset

The secondary dataset consists of **4,764** images spanning 15 mushroom types. All images have been preprocessed by the original dataset authors, including background removal, resizing to 640x640 pixels, and augmentation techniques such as rotation and flipping. These modifications were applied before incorporation into our study. A visual overview of the dataset can be seen in Figure 3.

Figure 3: Sample images from the secondary dataset

4.2 Experiments and results

4.2.1 Data Processing and Feature Selection

a) Preprocessing and Augmentation

To ensure consistency across datasets, preprocessing steps were applied.We also removed some noisy images, as shown in Figure 4, before performing further processing. The primary dataset underwent background removal using GrabCut before augmentation. Figure 5 illustrates an example of this process, where the background is successfully extracted, isolating the mushroom.

After background removal, the dataset was split into training and testing sets with an 80-20 ratio. Each training image, along with its original counterpart (for GIST processing), was augmented once using horizontal flipping and slight rotation to increase variability. Before training, the data underwent label encoding, standard scaling, and

mean imputation to ensure consistency and stability.

The secondary dataset, however, had already undergone preprocessing, including background removal and resizing to 640x640 pixels, as seen in Figure 3. Unlike the primary dataset, the secondary dataset had augmentation applied beforehand, requiring careful splitting to prevent data leakage. Specifically, augmented versions of the same image were ensured to remain within the same set, avoiding overlap between training and testing sets.

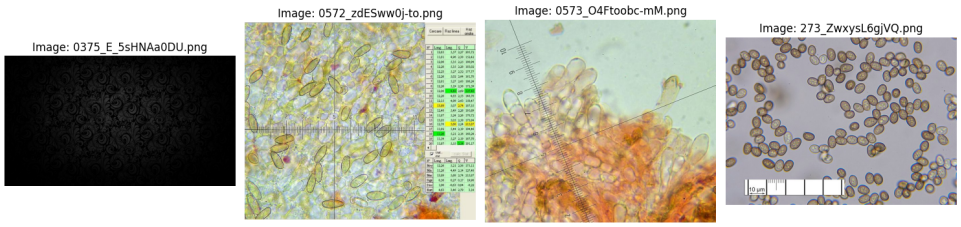

Figure 4: Example of noisy images before removal.

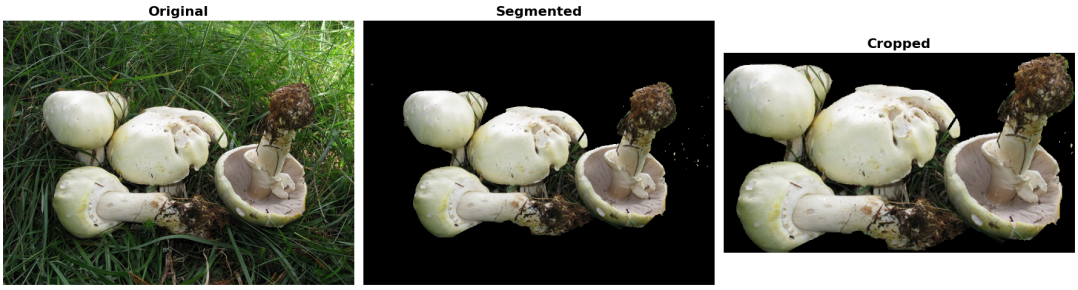

Figure 5: Comparison of different image processing techniques:
(Left) Original image, **(Middle)** Segmented image with background removal,
(Right) Cropped image focusing on mushrooms.

b) Feature Selection

To improve model performance, different feature combinations were tested using LightGBM as the evaluation model. The selected features included color histograms, texture descriptors, and edge-based features. Various feature subsets were examined to assess their impact on classification accuracy.

For the primary dataset, the optimal feature combination consisted of **Color histograms, SIFT, LBP, GIST, GLCM, ORB, Hu moments, and Color basic**, totaling 1,300 features:

$$1024 + 128 + 10 + 64 + 20 + 32 + 7 + 15 = 1300$$

However, not all features contributed equally to classification performance. LightGBM effectively utilized **873** features, filtering out less informative ones to improve efficiency. Other models employed different numbers of features, as summarized in Table 1.

Model	Number of Features Used
Random Forest	1060
LightGBM	873
XGBoost	900
CatBoost	590
Decision Tree	**1300 (All features retained)**
Extra Trees	**1300 (All features retained)**

Table 1: Number of features used by each model for the primary dataset

For the secondary dataset, the optimal feature combination consisted of **Color histograms, SIFT, LBP, GLCM, Hu moments, and ORB**, totaling 1,221 features:

$$1024 + 128 + 10 + 20 + 7 + 32 = 1221$$

Similarly, LightGBM effectively utilized only **407** features. The number of features used by other models is detailed in Table 2.

Model	Number of Features Used
Random Forest	630
LightGBM	407
XGBoost	416
CatBoost	365
Decision Tree	**1221 (All features retained)**
Extra Trees	**1221 (All features retained)**

Table 2: Number of features used by each model for the secondary dataset

Notably, **Decision Tree and Extra Trees retained all features** in both datasets, as these models do not perform intrinsic feature selection before training.

These results highlight the impact of dataset-specific feature engineering and the varying feature selection strategies across different models.

4.2.2 Results

This section presents the classification performance of four ML models: SVM, RF, LGBM, and XGB in distinguishing the purity of nine mushroom varieties. The experiments were

conducted on both the primary and secondary datasets, with results reported separately for clarity. To ensure a fair comparison, we used a consistent evaluation methodology with fixed training and test datasets.

a) The primary dataset

Each ML model was evaluated using the optimal feature set to leverage diverse color, texture, and edge-based features. Table 3 presents their classification performance.

Model	Accuracy	Precision	Recall	F1-score
LGBM	**0.6333**	**0.6359**	**0.6333**	**0.6215**
Random Forest	0.5477	0.5864	0.5477	0.5269
XGBoost	0.5932	0.5993	0.5932	0.5767
Decision Tree	0.3121	0.3152	0.3121	0.3133
Extra Trees	0.5311	0.5876	0.5311	0.5077
CatBoost	0.5530	0.5382	0.5530	0.5278

Table 3: Performance comparison of different models on the primary dataset

LGBM achieved the highest accuracy (63.33%), followed by XGBoost and CatBoost. Boosting models consistently outperformed others, while Decision Tree had the lowest accuracy.

Figure 6 presents the confusion matrix of LGBM, highlighting misclassification patterns in mushroom variety prediction.

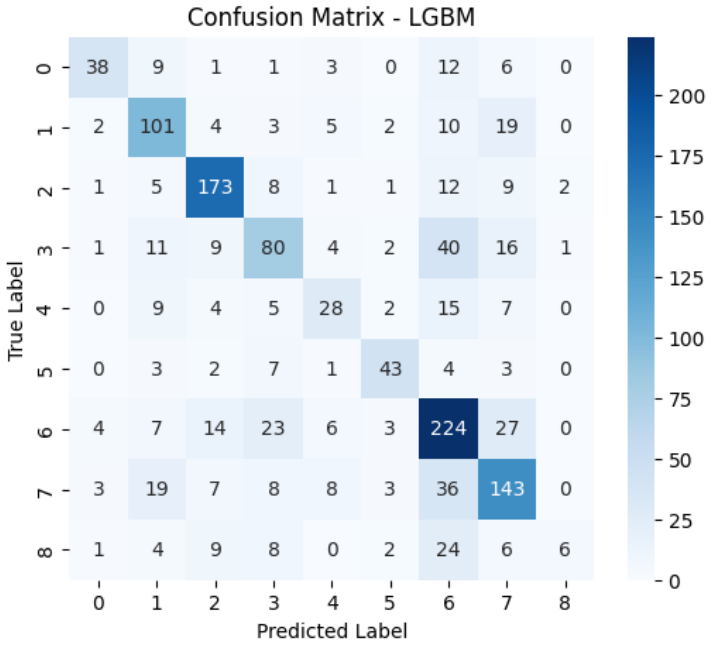

Figure 6: Confusion Matrix of LightGBM on the primary dataset: (0) *Agaricus*, (1) *Amanita*, (2) *Boletus*, (3) *Cortinarius*, (4) *Entoloma*, (5) *Hygrocybe*, (6) *Lactarius*, (7) *Russula*, and (8) *Suillus*.

b) The secondary dataset

Table 4 presents the model performances on the **Klasifikasi Jamur Dataset**.

Model	Accuracy	Precision	Recall	F1-score
LGBM	0.9088	0.9098	0.9088	0.9071
Random Forest	0.9256	0.9318	0.9256	0.9249
XGBoost	0.8857	0.8896	0.8857	0.8850
Decision Tree	0.7254	0.7697	0.7254	0.7249
Extra Trees	**0.9308**	**0.9345**	**0.9308**	**0.9303**
CatBoost	0.9256	0.9298	0.9256	0.9246

Table 4: Performance comparison of different models on the secondary dataset

Extra Trees achieved the highest accuracy (93.08%), slightly outperforming Random Forest and CatBoost. Decision Tree had the lowest performance (72.54%), reinforcing the effectiveness of ensemble methods.

Figure 7 shows the confusion matrix of Extra Trees, highlighting classification strengths and misclassification trends.

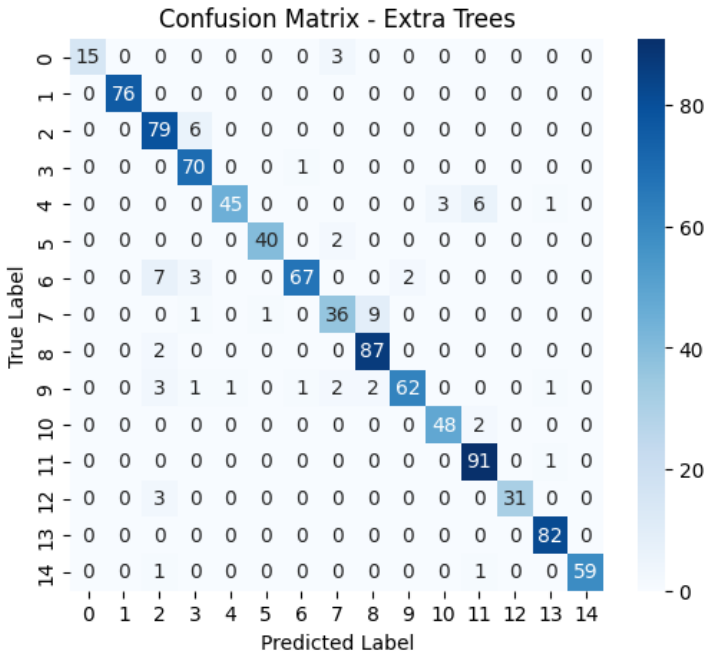

Figure 7: Confusion Matrix of Extra Trees on the secondary dataset: (0) *Chanterelle*, (1) *Dried Shiitake*, (2) *Dried Wood Ear*, (3) *Enoki*, (4) *King Oyster*, (5) *Lingzhi*, (6) *Oyster*, (7) *Red Pine*, (8) *Shiitake*, (9) *Shimeji*, (10) *Snow*, (11) *Straw*, (12) *Truffle*, (13) *White Button*, (14) *Wood Ear*.

4.2.3 Feature Selection Analysis

Although feature selection was explored using feature importance rankings from Random Forest and LightGBM, the performance gains were negligible (less than 1%). Therefore, the initial feature set was retained to ensure stability in subsequent evaluations.

However, analyzing feature importance still provides valuable insights into how models make decisions. To visualize this, we present the top 20 most important features ranked by Random Forest and LightGBM. This highlights which features contribute the most to classification, even though the full feature set was preserved.The detailed ranking can be seen in Figure 8.

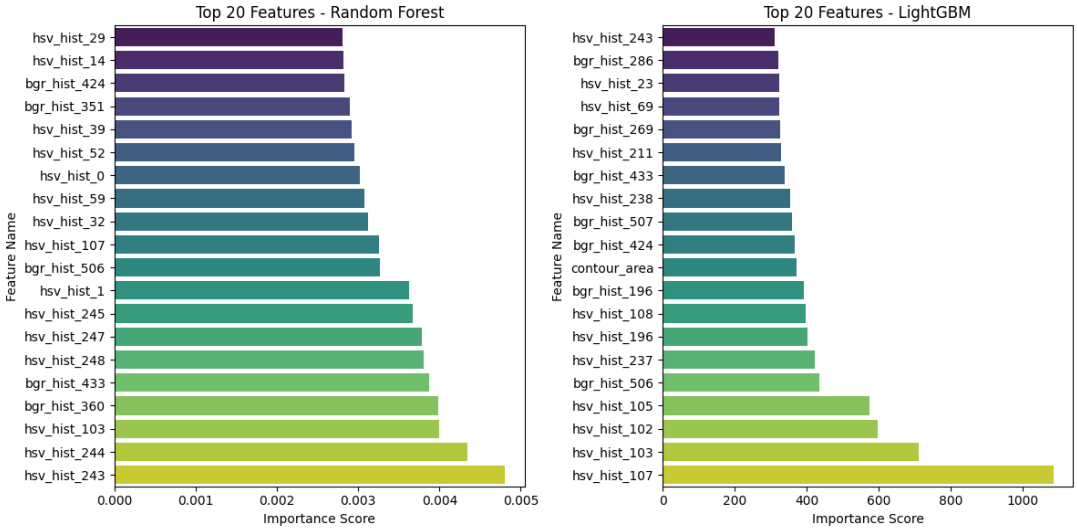

Figure 8: Top 20 most important features ranked by Random Forest and LightGBM for the primary dataset.

Random Forest tends to distribute importance more evenly across features, whereas LightGBM assigns significantly higher importance to a few key features, such as `hsv_hist_107`. This suggests that LightGBM may rely on a smaller subset of critical features for decision-making, while Random Forest leverages a more diverse set. Despite these differences, the overall feature importance patterns align, reinforcing the robustness of the selected feature set.

5 Errors and Adjustments

Data Leakage

- **Errors Encountered:** Data leakage occurred due to applying augmentation on both training and test sets. Additionally, labels from the test set were unintentionally included in training.
- **Adjustments Made:** Augmentation was restricted to only the training set to prevent information from leaking into the test set. The dataset pipeline was revised to ensure no label contamination.
- **Effect on Results:** The original inflated accuracy was corrected after fixing the leakage, leading to more realistic and generalizable model performance.

6 Comparison and Discussion

This section is a comparative discussion of different classification models for both datasets, along with major observations, strengths, and weaknesses.

6.1 Model Performance Comparison

LightGBM led in the primary dataset with an accuracy of 63.33%, while Extra Trees and Random Forest trailed closely behind in performance. Decision Tree registered the lowest accuracy (31.21%), justifying the point that single tree-based models lack the ability to handle complex image classification tasks.

In the secondary dataset, Extra Trees performed best with an accuracy of 93.08%, indicating that this model benefits significantly from well-processed images with cleaner backgrounds. Random Forest and CatBoost performed similarly well, while XGBoost and LightGBM had slightly lower accuracy. Decision Tree remained the worst-performing model, reinforcing the necessity of ensemble learning for robust mushroom classification.

The differences in model performance across the two datasets can be attributed to their distinct preprocessing characteristics. The key factors explaining these variations include:

- **Robustness to Background Variability:** LightGBM achieved the highest accuracy on the **primary dataset**, where background removal was performed by our team but with less refinement compared to the secondary dataset. As a boosting

algorithm, LightGBM grows trees leaf-wise, allowing it to focus on the most informative splits while handling background inconsistencies and lighting variations effectively.

- **Feature Selection Efficiency:** LightGBM's built-in feature selection helps mitigate the effects of residual background artifacts, which is crucial in datasets where background removal is not perfectly clean.

- **Impact of Preprocessing on Tree-Based Models:** In contrast, Extra Trees outperformed other models on the **secondary dataset**, where images had undergone extensive preprocessing by the original dataset authors. The structured nature of this dataset reduced background variability, allowing tree-based models to extract more stable patterns.

- **Randomization and Feature Interaction:** *Extra Trees, which constructs fully randomized trees, captures feature interactions effectively without requiring extensive hyperparameter tuning.* The reduced background complexity in the secondary dataset allowed it to leverage discriminative patterns more efficiently, resulting in higher classification accuracy.

6.2 Influence of Dataset Properties

One of the most notable differences between the two datasets is the level of background processing. The primary dataset consists of images where background removal was applied by our team but not as extensively refined as in the secondary dataset, which had undergone structured preprocessing by the original dataset authors. While cleaner images in the secondary dataset improved accuracy for tree-based models like Extra Trees, LightGBM demonstrated superior performance on the primary dataset, suggesting that certain boosting methods are more resilient to background artifacts.

Additionally, class imbalance affected performance across both datasets. In the primary dataset, classes such as *Lactarius* (class 6) and *Russula* (class 7) had significantly more training images due to augmentation, leading to classification biases where models were more likely to predict these dominant classes. Similarly, in the secondary dataset, classes such as *Dried Wood Ear* (class 2) and *Straw* (class 11) were overrepresented, potentially influencing model predictions.

While oversampling techniques such as SMOTE were considered, they did not significantly improve classification performance. A possible explanation is that synthetic samples generated by SMOTE lacked meaningful variation, as certain mushroom genera exhibit high visual similarity. This limitation suggests that alternative approaches, such as cost-sensitive learning or targeted data collection strategies, may be more effective in addressing class imbalance.

6.3 Limitations and Challenges

Despite strong classification performance, several challenges remain. First, mushroom classification using machine learning is still a developing field with limited benchmark studies for comparison, making it difficult to assess whether the achieved results represent a significant advancement over existing methods.

Second, while ensemble models performed well, their complexity and computational cost could limit real-time applications. Additionally, models trained on preprocessed images may struggle with unprocessed real-world data, highlighting the trade-off between preprocessing quality and model generalizability.

Finally, the high visual similarity between certain mushroom genera and class imbalances contributed to misclassifications, as observed in the confusion matrices. Future work should explore advanced augmentation strategies, cost-sensitive learning, or the integration of deep learning techniques to enhance classification robustness.

6.4 Summary of Findings

The results emphasize that the LightGBM and Extra Trees ensemble algorithms provided the most precise classification across both datasets. Image preprocessing in the secondary dataset improved precision for tree-based models but may not generalize well outside controlled experiments. Meanwhile, boosting models like LightGBM exhibited strong resilience to background artifacts, making them better suited for less refined datasets. Addressing class imbalance and enhancing feature extraction techniques remain crucial challenges in improving mushroom classification accuracy.

7 Conclusions

This project contrasted some mushroom identification classification models on two image datasets with varying preprocessing features. It was noted that the ensemble learning algorithms, Extra Trees and LightGBM, had a tendency to perform better than their individual tree versions. Preprocessing of the dataset was observed to have some effect, with the models trained on the extra dataset with larger numbers of preprocessed images providing greater accuracy but questionable real-world usability.

Although with much promise, there are some challenges. The class imbalance interfered with the model's predictions and introduced overrepresented class biases. Similarity in appearance between some mushroom genera also led to misclassifications. The future work should explore more sophisticated methods of augmentation, cost-sensitive learning, and feature extraction augmentation for enhancing robustness in classification.

For actual application, the blending of raw image and processed information in a hybrid environment can well compromise accuracy at the expense of applicability and vice versa. Deep learning operations can be incorporated into other researches for more accurate feature representation and recognition.

References

[1] Ottom, M. Classification of mushroom fungi using machine learning techniques. Int. J. Adv. Trends Comput. Sci. Eng. 2019,8,2378–2385.

Available:https://www.researchgate.net/publication/337024220_Classification_of_Mushroom_Fungi_Using_Machine_Learning_Techniques

[2] J. Preechasuk, O. Chaowalit, F. Pensiri, and P. Visutsak, "Image analysisof mushroom types classification by convolution neural networks," inProceedings of the 2019 2nd Artificial Intelligence and Cloud ComputingConference, 2019.

Available:https://www.researchgate.net/publication/339306595_Image_Analysis_of_Mushroom_Types_Classification_by_Convolution_Neural_Networks

[3] Rother, C., Kolmogorov, V., and Blake, A. (2004). "grabcut":interactive foreground extraction using iterated graph cuts. ACMTransactions on Graphics, 23(3):309–314.

Available:https://www.researchgate.net/publication/220184077_GrabCut_Interactive_Foreground_Extraction_Using_Iterated_Graph_Cuts

[4] [Online]Available:https://www.kaggle.com/datasets/maysee/mushrooms-classification-common-genuss-images

[5] [Online]Available:https://universe.roboflow.com/binusbandung/klasifikasi-jamur

[6] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. International Journal of Computer Vision, 42:145–175, May,2001.

Available:https://www.researchgate.net/figure/GIST-descriptor-Oliva-Torralba-2001-a fig1_334651721

[7] D G Lowe, "Object recognition from local scale-invariant features[C]", Proceedings of the seventh IEEE international conference on computer vision, vol. 2, pp. 1150-1157, 1999.

Available:https://ieeexplore.ieee.org/document/790410/authors#authors

[8] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "Orb: An efficient alternative to sift or surf", 2011 International conference on computer vision, pp. 2564-2571, 2011.

Available:https://ieeexplore.ieee.org/document/6126544

[9] M.-K. Hu, "Visual pattern recognition by moment invariants", IEEE Trans. Inf. Theory, vol. IT-8, no. 2, pp. 179-187, Feb. 1962.

Available:https://ieeexplore.ieee.org/document/1057692

[10] X. Y. Gong, H. Su, D. Xu et al.,"An overview of contour detection approaches,"Int. J. Autom. Comput. 15, 656–672 (2018)

Available:https://www.researchgate.net/publication/326063134_An_Overview_of_Contour_Detection_Approaches

[11] Thi-Thu-Hong Phan, Tran Thi Thanh Hai, Le Thi Lan, Vo Ta Hoang, Vu Hai, and Thuy Thi Nguyen. Comparative study on vision based rice seed varieties identification. In 2015 Seventh International Conference on Knowledge and Systems Engineering (KSE), pages 377–382, 2015.

Available:https://ieeexplore.ieee.org/document/7371816

[12] M.P. Ojala and D. Harwood. Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In Proceedings of the 12th IAPR International Conference on Pattern Recognition, volume 1, pages 582–585,1994.

Available:https://ieeexplore.ieee.org/document/576366

[13] P. Liu, J.-M. Guo, K. Chamnongthai and H. Prasetyo, "Fusion of Color Histogram and LBP-based Features for Texture Image Retrieval and Classification," Information Sciences, vol. 390, pp. 95-111, 2017.

Available:https://www.researchgate.net/publication/312408472_Fusion_of_Color_Histogram_and_LBPbased_Features_for_Texture_Image_Retrieval_and_Classification

[14] R. M. Haralick, K. Shanmugam, and I. Dinstein. Texture analysis segmentation using grey level co-occurrence matrix. IEEE Transactions on Systems, Man, and Cybernetics, 3(6):610–621, 1973

Available:https://www.researchgate.net/publication/3115759_Haralick_RM_Shanmuga_K_Dinstein_ITextural_features_for_image_classification_IEEE_Trans_Syst_Man_Cybern_3_610-621

[15] Bahzad Charbuty and Adnan Abdulazeez. Classification based on decision tree algorithm for machine learning. Journal of Applied Science and Technology Trends, pages 20 – 28, Mar. 2021. Available:https://www.researchgate.net/publication/350386944_Classification_Based_on_Decision_Tree_Algorithm_for_Machine_Learning

[16] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. Available:https://www.researchgate.net/publication/350386944_Classification_Based_on_Decision_Tree_Algorithm_for_Machine_Learning

[17] D.E. Geurts and L. Wehenkel. Extremely randomized trees. Machine Learning, 63(1):3–42, 2006. Available:https://www.researchgate.net/publication/220343368_Extremely_Randomized_Trees

[18] M. R. Machado, S. Karray and I. T. de Sousa, "LightGBM: an Effective Decision Tree Gradient Boosting Method to Predict Customer Loyalty in the Finance Industry", 2019 14th International Conference on Computer Science and Education (ICCSE), pp. 1111-1116, 2019. Available:https://ieeexplore.ieee.org/document/8845529

[19] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 785–794, 2016. Available:https://www.researchgate.net/publication/310824798_XGBoost_A_Scalable_Tree_Boosting_System

[20] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin. Catboost: Unbiased boosting with categorical features. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, pages 6639–6649, 2018. Available:https://papers.nips.cc/paper_files/paper/2018/hash/14491b756b3a51daac41c24863285549-Abstract.html