

Multi-Scenario Preprocessing Analysis for Indonesian Spice Image Classification: A Comparative Study Using VGG16, MobileNetV3, and YOLOv11

Eric Ariyanto^{1✉}, Ema Utami²

^{1,2} Informatika, PJJ Pascasarjana, Universitas Amikom Yogyakarta, Indonesia

✉ **Corresponding Author** : Eric Ariyanto (e-mail: eric.ariyanto@gmail.com)

Article Information	ABSTRACT
Article History Received : May 11, 2026 Revised : May 28, 2026 Published : July 04, 2026	Indonesian spice identification is challenging due to high inter-class visual similarity among rhizome species, yet the systematic impact of image preprocessing on classification performance remains underexplored. This study evaluates three preprocessing techniques (CLAHE, Background Removal, and Data Augmentation) arranged in eight factorial schemes across three architectures (VGG16, MobileNetV3-Large, and YOLOv11-cl), comprising 72 training runs on an 856-image four-class dataset, with Macro-F1 on stratified 80/10/10 splits as the primary metric. Data Augmentation alone (scheme S4) produced the most consistent gain, with VGG16 reaching 93.80% Macro-F1. The full pipeline degraded performance across all architectures because CLAHE and Background Removal interact antagonistically, refuting the assumption that preprocessing effects are additive. Externally defined augmentation was also found to be silently bypassed by the Ultralytics framework. All five primary findings replicated on an independent seven-class dataset using 5-fold cross-validation, and Background Removal proved beneficial only for large, high-contrast objects. The study establishes an empirical preprocessing benchmark and practical deployment guidance for spice recognition.
Keywords: <i>Background Removal; CLAHE; Data Augmentation; Indonesian Spice Classification; Preprocessing Scheme Comparison.</i>	

INTRODUCTION

Indonesia is widely recognized as one of the world's most significant spice-producing nations, contributing substantially to global agricultural trade and domestic culinary heritage (Food and Agriculture Organization of the United Nations [FAO], 2022). Visual identification of spices, a task routinely performed manually by farmers, traders, and quality inspectors, is error-prone due to high inter-class morphological similarity, particularly among rhizome-based species such as galangal, ginger, kencur (*Kaempferia galanga*), and turmeric (Kementerian Pertanian, 2023).

The adoption of deep learning in agricultural image recognition has opened new possibilities for automated spice identification. Convolutional Neural Networks (CNNs) have demonstrated strong performance on fine-grained visual classification tasks (Alzubaidi et al., 2021; Purwono et al., 2022), and recent architectures such as MobileNetV3 and YOLOv11 have extended this capability to resource-constrained settings, with the YOLO family in particular evolving rapidly (Terven et al., 2023). However, classification performance in spice recognition is highly sensitive to preprocessing choices, and prior studies have evaluated these choices inconsistently.

Prior studies on Indonesian spice classification have applied different preprocessing strategies without systematic isolation of their individual effects. Early approaches relied on classical features: Riska and Farokhah (2021) classified Indonesian spices with a K-Nearest Neighbor model over color and texture features, an approach inherently constrained by hand-crafted descriptors. Subsequent deep learning studies improved on this, with data augmentation recurring as the common ingredient. Nisa and Candra (2023) reported approximately 90% accuracy with a CNN and augmentation on six classes, and Tanuwijaya and Roseanne (2021) reported approximately 92% accuracy using a modified VGG16 with augmentation on four classes; related deep learning work on Balinese seasoning classification reports comparable performance (Prasetia & Sunarya, 2024), and transfer-learning models reach 92.5% on a broader Indonesian medicinal-plant set (Musyaffa et al., 2024). Hermawan and Agustina (2023a), by contrast, reported only 60% validation accuracy for a VGG16 model on a harder seven-class set without preprocessing. Across these studies, data augmentation is the only technique applied with any consistency, while contrast enhancement and background removal are largely absent or unreported. Critically, no published study has simultaneously varied CLAHE, Background Removal, and Data Augmentation in a controlled factorial design, making it impossible to attribute observed performance differences to individual preprocessing components.

To address this gap, this study develops a systematic, component-wise preprocessing evaluation framework spanning 72 training runs ($3 \text{ architectures} \times 8 \text{ schemes} \times 3 \text{ seeds}$), with three main contributions. First, it introduces a structured eight-scheme factorial framework that isolates the individual and combined effects of CLAHE, Background Removal, and Data Augmentation, the first such framework in this domain, and confirms its generalizability on an independent seven-class dataset (Hermawan & Agustina, 2023b; 225 runs, 5-fold cross-validation). Second, it provides a cross-architecture comparison of VGG16, MobileNetV3-Large, and YOLOv11-cls under identical conditions, quantifying architecture-specific preprocessing sensitivity and revealing that externally defined augmentation pipelines are bypassed by the Ultralytics framework, an important caveat for YOLO practitioners. Third, it reports the counter-intuitive finding that the full preprocessing pipeline consistently underperforms the unprocessed baseline across all three architectures because CLAHE and Background Removal interact antagonistically, challenging the additive improvement assumption common in preprocessing design. The objective of this study is therefore to determine, through controlled factorial experimentation, which preprocessing techniques and combinations genuinely improve Indonesian spice image classification, and for which architectures.

RESEARCH METHODS

Research Design

This study adopts a quantitative experimental design structured as a factorial ablation study. Three preprocessing binary factors, CLAHE, Background Removal (BR), and Data Augmentation (DA), define eight experimental schemes ($2^3 = 8$) evaluated across three architectures and three random seeds. The unprocessed configuration (S1) serves as the anchor baseline against which all other schemes are compared. Figure 1 summarizes the end-to-end workflow, from data collection through independent external validation.

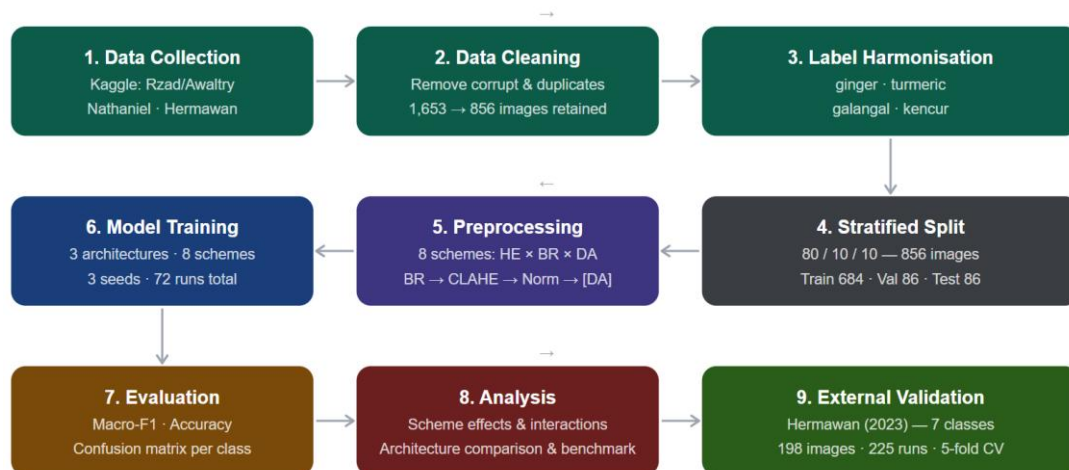


Figure 1. End-to-end research methodology, from data collection and cleaning, through per-scheme preprocessing, training of 72 runs, and evaluation, to independent external validation. (Source: this study)

Dataset Collection and Preparation

The dataset aggregates three publicly available Kaggle repositories (Awaltry/RZad, 2023; Nathaniel, 2024; Hermawan & Agustina, 2023b), the last reserved for external validation. Pooled images pass through three cleaning stages: corrupt-file rejection, cross-source deduplication by SHA-256 hash of raw bytes, and blank-image rejection (per-channel standard deviation < 0.10). Label harmonization maps colloquial and scientific folder names to four standardized English labels (for example, jahe, ginger, and zingiber_officinale all map to ginger). Of 1,653 target-class images discovered, 794 cross-source duplicates and 3 blank images were removed, and one repository overlapped 93% with another, contributing only 50 unique images.

After cleaning, the final dataset contains 856 images across four categories: galangal (207), ginger (230), kencur (210), and turmeric (209). The imbalance ratio is $1.11 \times (230/207)$, well below the 3:1 threshold requiring class-weighting interventions. A stratified 80/10/10 split yields Train: 684 / Validation: 86 / Test: 86 images. Figure 2 shows representative samples of the four spice classes, illustrating the high inter-class visual similarity among the rhizome species.

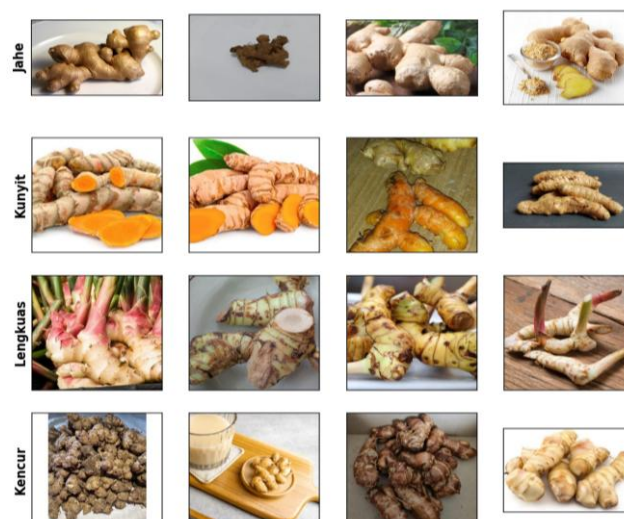


Figure 2. Representative samples of the four target classes (galangal, ginger, kencur, turmeric), illustrating the inter-class visual similarity that motivates the classification task.

(Source: Kaggle repositories listed in References)

Normalization statistics (mean and standard deviation per channel) are computed from the training split only and applied consistently across all three subsets to prevent data leakage.

Multi-source domain shift is acknowledged as a study limitation; the use of three seeds and stratified splits mitigates its impact on comparative validity.

Pre-processing Techniques

Contrast Limited Adaptive Histogram Equalization (CLAHE)

CLAHE (Zuiderveld, 1994) is applied on the L-channel of the CIE L*a*b* color space to enhance luminance contrast while preserving hue and saturation information. Tile grid size is set to 8×8 pixels with a contrast limit of 2.0. Operating on the L-channel rather than raw RGB prevents color distortion that would compromise class-discriminative hue information in spice images. CLAHE-based enhancement has been reported to improve deep-learning classification accuracy in other imaging domains (Hayati et al., 2023), motivating its inclusion as a preprocessing factor here.

Background Removal (BR)

Background removal uses the U²-Net salient object detection model via the rembg library (Qin et al., 2020), which sets all background pixels to zero (black); the output is converted to three-channel RGB for model compatibility. Imperfect segmentation inevitably introduces boundary artifacts at object edges, particularly for irregularly shaped spices, and the experiments quantify their net effect on classification.

Data Augmentation (DA)

Augmentation is applied exclusively to the training set (online, via the DataLoader) to prevent information leakage into validation and test splits. All transforms are implemented with the albumentations library: Random Horizontal Flip ($p = 0.5$), Random Vertical Flip ($p = 0.5$), Random Rotation ($\pm 30^\circ$), Random Resized Crop (scale 0.80–1.00), Color Jitter (brightness/contrast/saturation = 0.2/0.2/0.1), Gaussian Blur (3×3 kernel, $\sigma = 0.1$ –2.0), and Random Grayscale ($p = 0.10$). Parameters are selected to simulate realistic acquisition variability (illumination change, partial occlusion, orientation variation) without destroying class-discriminative surface texture cues, following standard practice for expanding limited training data (Xu et al., 2023).

Preprocessing Pipeline Order

The fixed pipeline order applies Background Removal first, followed by CLAHE, then Resize, then Normalize, and finally Data Augmentation during training only, as illustrated in Figure 3. This ordering ensures that background removal acts on raw pixels before contrast modification, that CLAHE enhances the object region rather than background noise, and that augmentation diversifies only the training distribution while inference uses a fully deterministic pipeline.

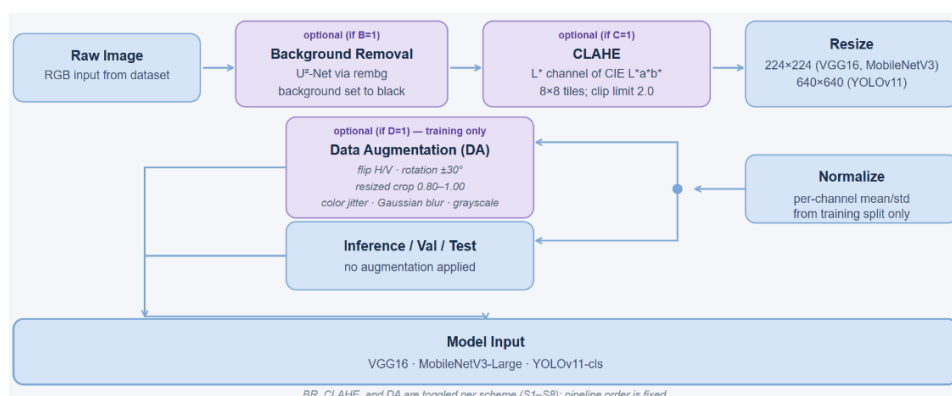


Figure 3. Preprocessing pipeline order. The deterministic chain (Background Removal, CLAHE, Resize, Normalize) is applied to all splits; Data Augmentation is applied online to the training split only. Each component is toggled per scheme (S1 to S8). (Source: this study)

Experimental Scheme Design

Eight schemes are defined by systematically varying the activation state of the three preprocessing components (Table 1). S1 serves as the unprocessed baseline. S2 to S4 isolate individual contributions. S5 to S7 evaluate pairwise interactions. S8 evaluates the full combined pipeline.

Table 1. Experimental Preprocessing Schemes (Factorial Design: CLAHE $\times$ BR $\times$ DA)

Scheme	CLAHE	BR	DA	Description
S1	X	X	X	Unprocessed baseline
S2	✓	X	X	Contrast enhancement only
S3	X	✓	X	Background removal only
S4	X	X	✓	Augmentation only
S5	✓	✓	X	CLAHE + BR (no augmentation)
S6	✓	X	✓	CLAHE + DA (no background removal)
S7	X	✓	✓	BR + DA (no contrast enhancement)
S8	✓	✓	✓	Full preprocessing pipeline

Model Architecture

Three architectures span a range of capacities and design philosophies. VGG16 (Simonyan & Zisserman, 2014), a deep CNN with 13 convolutional and 3 fully connected layers widely used as a baseline in this domain, is fully fine-tuned from ImageNet weights at 224 $\times$ 224. MobileNetV3-Large (Howard et al., 2019), a lightweight network using depthwise separable convolutions and hard-swish activation, is fine-tuned in its last two feature blocks and classifier head (earlier layers frozen) at 224 $\times$ 224. YOLOv11-cl (Ultralytics, 2024), the classification variant of the latest YOLO series with multi-scale feature extraction, is used as yolo11n-cl at 640 $\times$ 640.

Training Configuration

All experiments share standardized training conditions to ensure fair cross-model and cross-scheme comparison. The three architectures are optimized with Adam (initial learning rate 0.0001 for VGG16 and YOLOv11, 0.0003 for MobileNetV3; weight decay 1e-4 for the CNNs and 5e-4 for YOLOv11) for up to 100 epochs, with batch sizes of 16 for VGG16 and YOLOv11 and 32 for MobileNetV3, and early stopping after 15 epochs without validation-loss improvement. Each configuration (architecture $\times$ scheme $\times$ seed) produces one saved checkpoint (best_model.pth) selected at the epoch with lowest validation loss.

Total runs: 72 (3 architectures $\times$ 8 schemes $\times$ 3 seeds: 42, 123, 456). Three seeds balance reproducibility against the substantial cost of training 72 configurations for up to 100 epochs each; consistency of effect direction across seeds is the primary stability indicator, complemented by the external validation described below.

Evaluation Metrics

Performance is reported on the held-out test set (86 images) using four metrics: Accuracy, Macro-Precision, Macro-Recall, and Macro-F1 Score. Macro-averaging assigns equal weight to all four classes regardless of frequency, making it robust to class imbalance and sensitive to per-class performance balance. Macro-F1 is designated the primary evaluation metric.

Each metric is reported as mean $\pm$ standard deviation across three seeds, and pairwise Wilcoxon signed-rank tests compare each scheme against the S1 baseline. With $n = 3$ seeds the minimum achievable two-sided p-value is 0.25, so results are interpreted as consistency of effect direction rather than statistical significance, with Δ F1 (the mean difference in Macro-F1 vs. S1) as the primary effect size measure. The inferential strength of the design comes not from seed count but from independent external replication: every primary finding is re-tested on a different dataset under 5-fold cross-validation (225 runs), a more demanding and more generalizable robustness check than additional seeds on a single dataset could provide.

RESULTS AND DISCUSSION

Overall Classification Performance

Table 2 presents the complete Macro-F1 results for all 24 model-scheme combinations (3 architectures $\times$ 8 schemes), reported as mean $\pm$ standard deviation across three seeds. Values in percent; Figure 4 visualizes the same results as a heatmap.

Table 2. Macro-F1 Score (%) — All Model and Preprocessing Scheme Combinations

Scheme	VGG16	MobileNetV3	YOLOv11	Cross-arch avg
S1 – baseline	90.35 $\pm$ 1.42	88.33 $\pm$ 0.02	90.79 $\pm$ 1.94	89.82
S2 – CLAHE	87.64 $\pm$ 0.71	88.33 $\pm$ 1.11	90.39 $\pm$ 1.79	88.79
S3 – BR	85.20 $\pm$ 4.14	88.42 $\pm$ 1.19	88.45 $\pm$ 2.37	87.36
S4 – DA	93.80 $\pm$ 1.88	89.87 $\pm$ 2.45	90.79 $\pm$ 1.94*	91.49
S5 – CLAHE + BR	84.03 $\pm$ 2.95	87.70 $\pm$ 2.58	86.54 $\pm$ 0.57	86.09
S6 – CLAHE + DA	90.65 $\pm$ 2.43	89.89 $\pm$ 0.63	90.39 $\pm$ 1.79	90.31
S7 – BR + DA	87.59 $\pm$ 0.76	85.56 $\pm$ 0.74	88.45 $\pm$ 2.37	87.20
S8 – FULL	87.14 $\pm$ 3.67	86.84 $\pm$ 0.70	86.54 $\pm$ 0.57	86.84

*S4 = S1 for YOLOv11 see YOLOv11 Data Augmentation Null Effect.

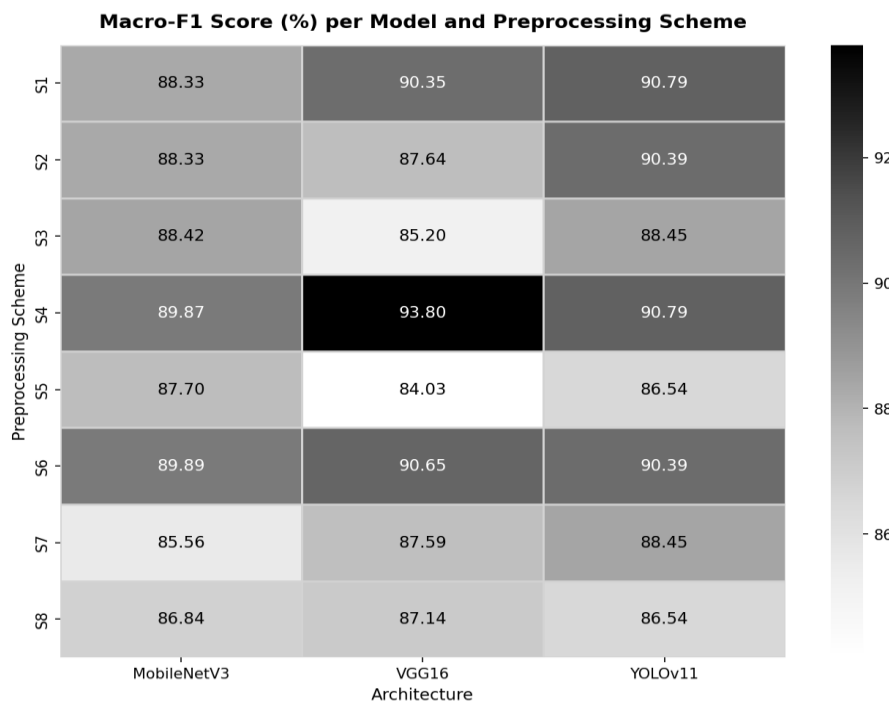


Figure 4. Macro-F1 Score (%) heatmap across 3 architectures $\times$ 8 preprocessing schemes. Cell values (means across 3 seeds) are annotated; higher values are shaded darker and lower values lighter. (Source: this study)

Across all 24 configurations, Macro-F1 ranged from 84.03% (VGG16, S5) to 93.80% (VGG16, S4), spanning 9.77 percentage points. The best single configuration is VGG16 $\times$ S4, and the best cross-architecture average is also S4 (91.49%), consistent with Data Augmentation being the most universally effective preprocessing component.

Effect of Individual Preprocessing Techniques

Relative to the S1 baseline, the single-component schemes (S2, S3, S4) show clearly different effect directions. Data Augmentation (S4) produced the most consistent individual improvement across CNN architectures: VGG16 gained +3.45 pp and MobileNetV3 gained +1.54 pp (mean +1.66 pp across architectures), suggesting that expanding the effective training distribution is the most universally beneficial preprocessing strategy for this task. The Δ is positive

across all independent seeds for both architectures (consistent effect direction). YOLOv11 registered no change under S4 ($\Delta = 0.00$ pp), for the methodological reason explained below.

CLAHE (S2) produced mixed effects (mean -1.04 pp): VGG16 showed the largest degradation (-2.71 pp), MobileNetV3 recorded no net change, and YOLOv11 showed minimal degradation (-0.40 pp). This architecture-dependent response is consistent with a normalization mechanism: VGG16 lacks batch normalization, so the amplified local contrast gradients introduced by CLAHE propagate unchecked through its feature hierarchy, whereas MobileNetV3's batch normalization and Squeeze-and-Excitation attention renormalize feature distributions and suppress artifact-dominated channels, leaving it effectively neutral to CLAHE.

Background Removal (S3) yielded a mean Δ of -2.47 pp, with the largest degradation in VGG16 (-5.15 pp): U²-Net's imperfect segmentation introduces boundary artifacts that act as spurious edge-level features. MobileNetV3 proved most resilient, likely because its depthwise separable convolutions are less sensitive to local boundary noise.

YOLOv11 Data Augmentation Null Effect: A Methodological Finding

A critical methodological observation is that YOLOv11 produces identical results for all augmentation-containing schemes: S4 = S1, S6 = S2, S7 = S3, and S8 = S5, yielding $\Delta = 0.00$ pp across all four pairs. This pattern holds consistently across all three seeds per pair (12 pairwise comparisons total), making coincidence implausible; it is a systematic pipeline-level effect.

The mechanism lies in how Ultralytics YOLO loads data. When `model.train()` is invoked, the framework builds its own internal `ClassificationDataset` and `DataLoader` from the supplied image directory paths and does not accept an externally instantiated `Dataset` object (Ultralytics, 2024). The `albumentations` transforms registered in a custom `Dataset.__getitem__` are therefore never called, as Figure 5 illustrates: only the online augmentation stage is bypassed, while offline steps already written to disk (BR, CLAHE) are carried through, which is why $S3 \neq S1$ and $S5 \neq S2$ for YOLOv11. In place of the external transforms, Ultralytics applies its own built-in augmentation for the classify task by default (random resized crop, horizontal flip, HSV color jitter, RandAugment, and random erasing); detection-only parameters such as mosaic and MixUp are recorded in `args.yaml` with default values but remain inert for classification.

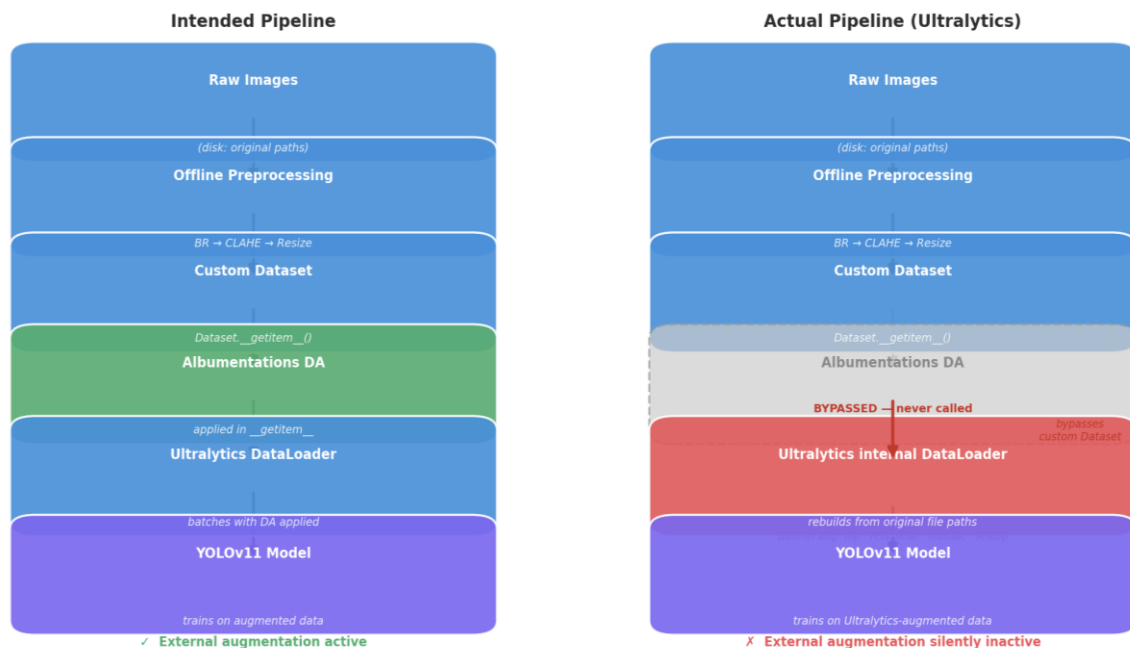


Figure 5. Pipeline collision diagram illustrating why externally defined augmentation is bypassed in YOLOv11 training: the intended data flow (left) versus the actual data flow (right) in the Ultralytics framework. (Source: this study)

This finding has an important practical implication: researchers using Ultralytics YOLO must not assume that externally configured augmentation pipelines are active during training. Studies comparing augmentation strategies with YOLO-series models must either (a) configure augmentation through Ultralytics' native training arguments appropriate to the task (for classification: `flip_lr`, `hsv_h`, `auto_augment`, `erasing`; `mosaic`/`mixup` apply to detection only), or (b) report this architectural constraint explicitly as a methodological limitation.

Effect of Combined Preprocessing Schemes

Among pairwise combinations, S6 (CLAHE + DA) achieved the highest cross-architecture average (90.31%), representing +1.49 pp over S1. For MobileNetV3, S6 is the overall best scheme ($89.89\% \pm 0.63$ pp), marginally exceeding DA-only (S4: 89.87%). The low standard deviation of MobileNetV3-S6 (0.63 pp vs. 2.45 pp for S4) suggests CLAHE provides a stabilizing effect when combined with augmentation for lightweight architectures.

S7 (BR + DA) performed consistently worse than S4 (DA only) across all architectures: BR's negative effect persists even when combined with augmentation, suggesting that boundary artifacts introduced by segmentation cannot be overcome by augmentation diversity alone.

The full pipeline S8 consistently underperformed the baseline S1 for all three architectures (avg -2.98 pp), providing evidence against the additive improvement hypothesis. The cause is an antagonistic interaction between CLAHE and BR: once BR sets the background to hard black, the sharp black-to-object boundaries fall inside CLAHE's 8×8 tiles, where histograms dominated by zeros drive extreme contrast redistribution and create spurious high-frequency edges. Because CNN filters respond strongly to edge gradients, these artifacts become salient features that crowd out genuine morphological cues. Each technique therefore amplifies the other's artifacts, which is why S5 (CLAHE+BR) and S8 fall below both S2 and S3 individually; by cross-architecture average the hierarchy $S4 > S6 > S1 > S2 > S3 > S7 > S8 > S5$ places S5 as the single worst combination.

Cross-Architecture Comparison

Each architecture reaches its best Macro-F1 under a different configuration: VGG16 at S4 ($93.80\% \pm 1.88$ pp, with accuracy 93.80%, precision 94.05%, and recall 93.86%), MobileNetV3 at S6 ($89.89\% \pm 0.63$ pp), and YOLOv11 at S1 ($90.79\% \pm 1.94$ pp). Their sensitivity to preprocessing, measured as the F1 range (the maximum minus minimum Macro-F1 across all eight schemes, where a higher value indicates greater sensitivity), differs sharply.

VGG16 attained the highest performance (93.80% F1) but also the highest preprocessing sensitivity (F1 range 9.77 pp): its large capacity exploits augmented data well, yet its deep fully-connected layers are vulnerable to background removal artifacts, and with 138M parameters trained on only 684 images its result is likely a conservative estimate of its capacity. MobileNetV3 was the most stable across schemes (range 4.33 pp), making it the safer choice when preprocessing is uncertain, and its best S6 result shows lightweight models can benefit from mild contrast enhancement paired with augmentation. YOLOv11 achieved the highest baseline (90.79%) and the most preprocessing-independent behavior, owing to its built-in augmentation.

These differing ranges reflect architectural inductive bias toward input invariance: VGG16's deep-narrow design without batch normalization is maximally expressive but maximally sensitive to perturbations, whether beneficial (DA) or harmful (BR artifacts), whereas MobileNetV3's batch normalization and SE attention, and YOLOv11's multi-scale feature pyramid, absorb distributional shifts before they compound. Practitioners deploying under variable or uncontrolled preprocessing should favor MobileNetV3 or YOLOv11; VGG16 suits only fully controlled pipelines.

A latency profile (CUDA GPU, batch size = 1) reinforces this guidance. Background Removal is the dominant bottleneck at roughly 98 ms per image, about $19\times$ slower than the fastest model inference (YOLOv11: 2.80 ms) and limiting any BR-containing pipeline to about 10 FPS even on GPU, with substantially worse latency on CPU. The three optimal configurations all avoid BR and remain viable in real time: VGG16-S4 at 5.26 ms (190 FPS), MobileNetV3-S6 at 8.76 ms

(114 FPS), and YOLOv11-S1 at 3.03 ms (330 FPS). Latency therefore gives a second, independent reason to avoid Background Removal in production.

Misclassification Analysis

Confusion matrices were generated for the best configuration per architecture, aggregated across three seeds and normalized per row. Figure 6 illustrates the 4×4 normalized confusion matrices.

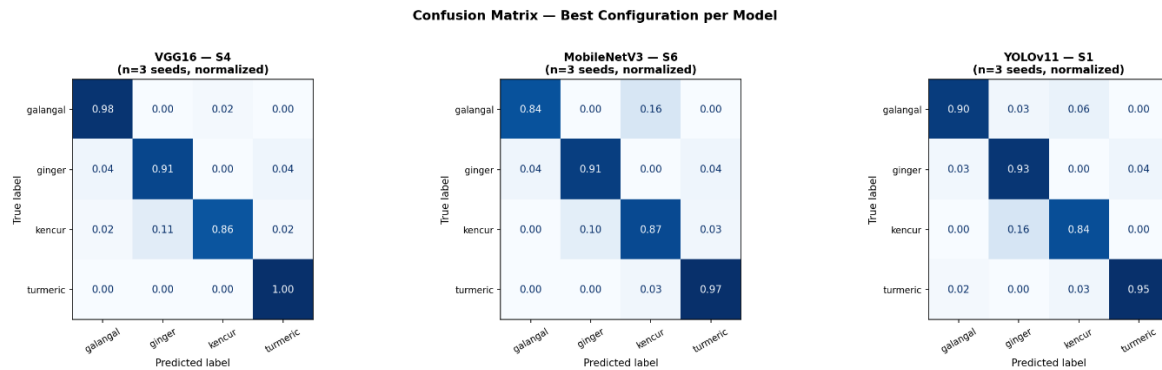


Figure 6. Normalized confusion matrices for optimal configurations: VGG16-S4 (left), MobileNetV3-S6 (center), YOLOv11-S1 (right). Values represent the proportion of true class instances (rows) classified into each predicted class (columns), aggregated across 3 seeds. (Source: this study)

The dominant misclassification across all three architectures is kencur classified as ginger (VGG16-S4 misclassifies 11.1% of kencur as ginger), reflecting their morphological similarity: both are small, irregularly shaped corms with brownish-tan surfaces. Kencur consistently records the lowest per-class F1 (VGG16-S4: 0.9150, MobileNetV3-S6: 0.8463, YOLOv11-S1: 0.8690), while turmeric records the highest, owing to its distinctive yellow-orange color, and galangal benefits from its larger size and pinker cut surface.

Benchmarking Against Prior Studies

The best configuration in this study (VGG16-S4, Macro-F1 = 93.80%) surpasses the highest directly comparable prior result (Tanuwijaya & Roseanne, 2021: approximately 92% accuracy), as well as the augmentation-based CNN result of Nisa and Candra (2023; approximately 90%), while using a reproducible, publicly documented dataset and controlled experimental conditions. Direct comparison is necessarily limited by differences in dataset composition and evaluation protocol, although accuracy and Macro-F1 are approximately equivalent for balanced four-class datasets.

External Validation – Hermawan (2023) Dataset

To assess the generalizability of findings beyond the primary dataset, the experimental framework is replicated on the independent dataset of Hermawan and Agustina (2023b): 7 spice classes (andaliman, cabe jawa, cengkeh, kapulaga, kayu manis, lada, pala), 198 unique images after SHA-256 deduplication, pre-defined splits discarded due to verified cross-split data leakage and three cross-class label errors. A 5-fold stratified cross-validation protocol is applied (225 runs: 5 schemes × 3 architectures × 3 seeds × 5 folds). Only schemes S1 to S5 are evaluated to focus on the most informative configurations.

Table 3. Macro-F1 (%) — External Validation (mean across 3 seeds × 5 folds)

Scheme	VGG16	MobileNetV3	YOLOv11
S1 – baseline	66.94	78.11	76.94
S2 – CLAHE	67.85	78.68	77.40
S3 – BR	61.04	71.40	69.66
S4 – DA	72.27	82.94	76.94
S5 – CLAHE + BR	63.95	72.98	70.90

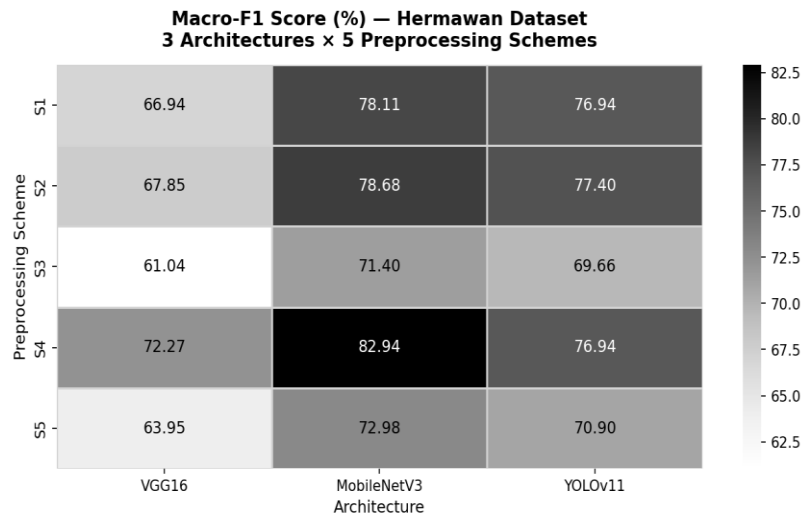


Figure 7. Macro-F1 Score (%) heatmap for external validation on the Hermawan and Agustina dataset (3 architectures × 5 schemes, mean across 3 seeds × 5 folds).
(Source: this study, Hermawan and Agustina dataset)

As the Hermawan heatmap in Figure 7 shows, all five primary findings replicated consistently: data augmentation (S4) was again the best scheme for the CNN architectures (VGG16 +5.33 pp, MobileNetV3 +4.83 pp over baseline), YOLOv11 again exhibited the augmentation null effect ($\Delta = 0.00$ pp), CLAHE-BR (S5) was again antagonistic relative to baseline (VGG16 −2.99 pp, MobileNetV3 −5.13 pp), Background Removal (S3) again degraded all three architectures, and CLAHE (S2) again had minimal net effect ($\Delta < +1$ pp across all models). The absolute F1 values are lower (approximately 20 pp below the primary study), which is expected given the substantially harder task: 7 classes, approximately 22 training images per class per fold versus 4 classes and 684 total training images in the primary study.

An additional finding on morphology-dependent background removal effectiveness emerged in the Hermawan validation. Per-class analysis of the S3 vs. S1 Δ F1, shown in Figure 8, reveals that BR effectiveness depends strongly on object morphology rather than class category alone. Background removal most damages small, clustered objects, such as andaliman (−22.6 pp) and cengkeh (−14.8 pp), while it benefits only large, compact, high-contrast objects, such as pala (+4.7 pp) and kapulaga (+2.8 pp); intermediate morphologies (cabe jawa −6.4 pp, lada −4.7 pp, kayu manis −4.1 pp) fall between these extremes.

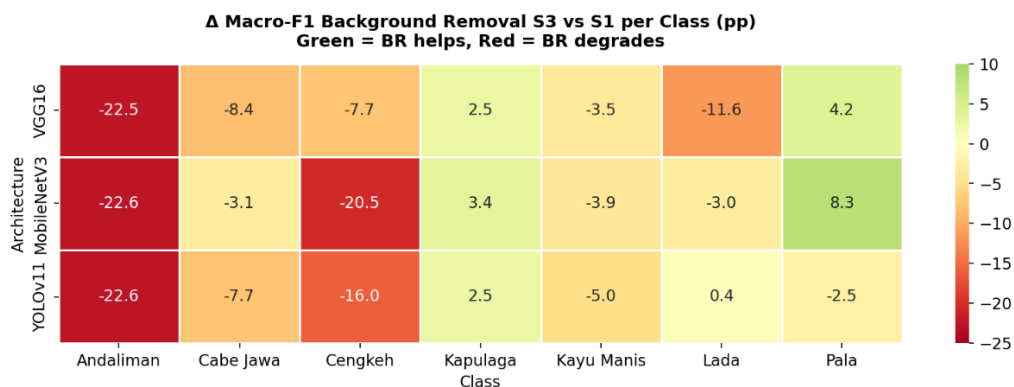


Figure 8. Per-class Δ F1 for background removal (S3 vs. S1) across three architectures. Positive annotated values indicate that BR improves the class; negative values indicate degradation.
(Source: this study, Hermawan and Agustina dataset)

Background removal is effective only for classes with compact, large-area objects and high foreground-background contrast (kapulaga, pala). For classes with small, clustered, or irregularly bounded objects (andaliman, cengkeh), BR removes discriminative features rather than isolating the object of interest. This finding refines the primary study's conclusion: the negative effect of BR is not uniform but depends on morphological characteristics of the target class. The hardest class in the Hermawan dataset is lada ($F1 = 52\text{--}59\%$ across architectures at best configurations), in contrast to kencur in the primary dataset. Lada's small, uniform grain texture creates high inter-class visual similarity with kapulaga at reduced image scales.

CONCLUSION

This study presented a systematic factorial analysis of image preprocessing for Indonesian spice classification, evaluating CLAHE, Background Removal, and Data Augmentation across eight schemes, three architectures, and three seeds (72 runs on 856 images), with independent external validation on a seven-class dataset (225 runs). Data Augmentation was the only consistently beneficial component, and the best overall configuration, VGG16 with augmentation alone, reached 93.80% Macro-F1, surpassing comparable prior studies. The full pipeline, by contrast, consistently underperformed the unprocessed baseline across all architectures because CLAHE and Background Removal interact antagonistically rather than additively. Background Removal was detrimental on average, but external validation showed its effect to be morphology-dependent, harming small or clustered objects (andaliman, -22.6 pp) while helping large, high-contrast ones (pala, $+4.7$ pp). A methodological caveat also surfaced: externally defined augmentation pipelines are inoperative inside the Ultralytics YOLOv11 framework, whose built-in augmentation instead produced the highest preprocessing-free baseline. Every one of these effect directions replicated under 5-fold cross-validation on the independent dataset, providing generalizability evidence that seed replication alone could not supply.

For practical deployment, VGG16 with augmentation is preferable when accuracy is paramount (93.80% F1, 5.26 ms per image on GPU); MobileNetV3 with CLAHE and augmentation suits resource-constrained settings (89.89% F1, with the lowest variance); and YOLOv11 on the unprocessed baseline is best when throughput dominates (90.79% F1, about 330 FPS). Background Removal should be avoided in production, as it adds roughly 98 ms per image and degrades performance across all three architectures.

Several limitations frame these conclusions and define directions for future work. The three seeds, chosen for computational feasibility, preclude formal significance testing, and the primary dataset of 856 images is modest, reflecting a field-wide scarcity of unique public Indonesian spice images, since cross-source deduplication removed 93% of one repository as duplicates. The external validation covered five of the eight schemes, and the primary dataset relied on a single stratified split rather than k-fold cross-validation. Future work should therefore apply 5-fold cross-validation to the primary dataset, expand to additional spice forms such as bark, seed, and powder, investigate class-adaptive preprocessing selection, and evaluate Background Removal with more precise segmentation models such as SAM to reduce boundary artifacts.

ACKNOWLEDGEMENTS

The authors would like to thank the Master of Informatics Engineering program at Universitas AMIKOM Yogyakarta for providing the computational resources and guidance necessary to complete this research.

REFERENCES

- Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(1), Article 53. <https://doi.org/10.1186/s40537-021-00444-8>

- Awaltry/Rzad. (2023). Indonesian spices dataset [Data set]. Kaggle. <https://www.kaggle.com/datasets/rzad24/indo-spices-new>
- Food and Agriculture Organization of the United Nations. (2022). Crops and livestock products [Data set]. FAOSTAT. <https://www.fao.org/faostat/en/#data/QCL>
- Hayati, M., Muchtar, K., Roslidar, Maulina, N., Syamsuddin, I., Elwirehardja, G. N., & Pardamean, B. (2023). Impact of CLAHE-based image enhancement for diabetic retinopathy classification through deep learning. *Procedia Computer Science*, 216, 57–66. <https://doi.org/10.1016/j.procs.2022.12.111>
- Hermawan, S., & Agustina, N. (2023a). Implementasi convolutional neural network untuk klasifikasi rempah-rempah khas Indonesia. *DoubleClick: Journal of Computer and Information Technology*, 7(1), 1–7.
- Hermawan, S., & Agustina, N. (2023b). Indonesian spices image dataset [Data set]. Kaggle. <https://www.kaggle.com/datasets/sandihermawan13/7-rempah-rempah-indonesia>
- Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., & Adam, H. (2019). Searching for MobileNetV3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV 2019)* (pp. 1314–1324). <https://doi.org/10.1109/ICCV.2019.00140>
- Kementerian Pertanian Republik Indonesia. (2023). Statistik pertanian 2023. Pusat Data dan Sistem Informasi Pertanian. <https://epublikasi.pertanian.go.id/>
- Musyaffa, M. S. I., Yudistira, N., Rahman, M. A., Basori, A. H., Firdausiah Mansur, A. B., & Batoro, J. (2024). IndoHerb: Indonesia medicinal plants recognition using transfer learning and deep learning. *Heliyon*, 10(23), Article e40606. <https://doi.org/10.1016/j.heliyon.2024.e40606>
- Nathaniel, A. (2024). Indonesian spices dataset [Data set]. Kaggle. <https://www.kaggle.com/datasets/albertnathaniel12/indonesian-spices-dataset>
- Nisa, C., & Candra, F. (2023). Klasifikasi jenis rempah-rempah menggunakan algoritma convolutional neural network. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(1), 78–84. <https://doi.org/10.57152/malcom.v4i1.1018>
- Prasetia, I. P. W., & Sunarya, I. M. G. (2024). Image classification of Balinese seasoning Base Genep based on deep learning. *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, 13(1), 79–90. <https://doi.org/10.23887/janapati.v13i1.67967>
- Purwono, P., Ma'arif, A., Rahmانيar, W., Fathurrahman, H., Frisky, A., & Haq, Q. (2022). Understanding of convolutional neural network (CNN): A review. *International Journal of Robotics and Control Systems*, 2(4), 739–748. <https://doi.org/10.31763/ijrcs.v2i4.888>
- Qin, X., Zhang, Z., Huang, C., Dehghan, M., Zaiane, O. R., & Jagersand, M. (2020). U²-Net: Going deeper with nested U-structure for salient object detection. *Pattern Recognition*, 106, Article 107404. <https://doi.org/10.1016/j.patcog.2020.107404>
- Riska, S. Y., & Farokhah, L. (2021). Classification of Indonesian spices using the K-Nearest Neighbors (K-NN) method. *Smatika Jurnal*, 11(1), 37–42.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR 2015)*. <https://arxiv.org/abs/1409.1556>
- Tanuwijaya, E., & Roseanne, A. (2021). Modifikasi arsitektur VGG16 untuk klasifikasi citra digital rempah-rempah Indonesia. *MATRIK: Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, 21(1), 189–196. <https://doi.org/10.30812/matrik.v21i1.1492>
- Terven, J., Córdova-Esparza, D.-M., & Romero-González, J.-A. (2023). A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS.

- Machine Learning and Knowledge Extraction, 5(4), 1680–1716.
<https://doi.org/10.3390/make5040083>
- Ultralytics. (2024). Ultralytics YOLO11 (Version 11.0) [Software]. GitHub.
<https://github.com/ultralytics/ultralytics>
- Xu, M., Yoon, S., Fuentes, A., & Park, D. S. (2023). A comprehensive survey of image augmentation techniques for deep learning. *Pattern Recognition*, 137, Article 109347.
<https://doi.org/10.1016/j.patcog.2022.109347>
- Zuiderveld, K. (1994). Contrast limited adaptive histogram equalization. In P. S. Heckbert (Ed.), *Graphics Gems IV* (pp. 474–485). Academic Press. <https://doi.org/10.1016/B978-0-12-336156-1.50061-6>