



Visual Classification of Spirulina Microalgae Maturity Stages Based on Color Using Convolutional Neural Networks

Dicky Hermawan¹, Hasanul Fahmi^{1,2}

¹Faculty of Computer Science, President University, Bekasi, Indonesia.

²Faculty of Artificial Intelligence & Frontier Technologies, UNITAR International University, Selangor, Malaysia

Article Info

Article history:

Received Mar 9, 2026

Revised Sept 4, 2026

Accepted Sept 21, 2026

Keywords:

Spirulina Microalgae;
Convolutional Neural Network;
Color-based classification;
Maturity stage classification;
Grad-CAM interpretability;

ABSTRACT

Manual assessment of microalgae maturity is subjective and does not scale well. This study evaluates whether a convolutional neural network can classify Spirulina microalgae into five visual maturity stages (dead, unripe, under-ripe, medium-ripe, ripe) using color alone. From 5,005 images collected over three months at a single facility, byte-exact duplicates accounted for 41.9% of the dataset; after removal, 2,909 unique images remained. To prevent near-duplicate images from the same session leaking across folds, evaluation used a group-stratified 5-fold cross-validation scheme, with groups defined via perceptual-hash clustering. An ablation across four preprocessing configurations found that combining CLAHE with augmentation, the original proposed method, was numerically weakest ($94.74\% \pm 2.15\%$ accuracy versus $97.43\% \pm 0.89\%$ with no preprocessing), differing significantly from CLAHE alone ($p = 0.021$) though not from the plain baseline at this sample size ($p = 0.085$). A separate comparison, matching fine-tuning depth (last 30 layers) and preprocessing across a custom CNN, MobileNetV2, and ResNet50, found ResNet50 highest at $99.52\% \pm 0.18\%$ accuracy, significantly ahead of the custom CNN ($97.06\% \pm 1.39\%$; $p = 0.024$), while MobileNetV2 showed severe fold-to-fold instability. The custom CNN's advantage lay in efficiency: 54.8 times fewer parameters and roughly 3.5 times faster inference than ResNet50, for only a small accuracy trade-off. A brief, unlabeled field test on six images outside the lab setting showed frequent disagreement among the three models, suggesting limited transfer from controlled to field conditions. Grad-CAM maps are presented as an indication of model attention, not evidence of underlying chlorophyll degradation biochemistry.

This is an open access article under the CC BY-NC license.



Corresponding Author:

Hasanul Fahmi,
Faculty of Artificial Intelligence & Frontier Technologies,
UNITAR International University,
Level 5, Tierra Crest, Jalan SS6/3, 47301 Kelana Jaya, Petaling Jaya, Selangor Darul Ehsan, Malaysia.
Email: fahmi.zuhry@unitar.my

1. INTRODUCTION

Color provides substantial information about the physiological state of a microalgae culture, reflecting biomass accumulation, pigment shifts associated with photosynthetic activity, and eventual cellular senescence. Spirulina growers use this directly: they sort cultures into five stages by color alone, from Dead through Unripe, Under-ripe, and Medium-ripe, to Ripe. This assessment is traditionally performed by visual inspection or laboratory-based sampling, both of which are subjective, time consuming, and difficult to standardize across observers. Image-based deep learning offers a way to

automate this visual classification task directly from cultivation images, reducing dependence on manual judgment [1], [2].

Continuous visual monitoring of ponds or photobioreactors is impractical at scale, and manual inspection is prone to inter-observer disagreement. Laboratory analysis, while more objective, adds cost and delay that limit its use for frequent monitoring. A further practical challenge is that natural and artificial lighting vary across acquisition conditions, which can distort the color information a classification model relies on. This study addresses the classification of *Spirulina* visual maturity stage from cultivation images under such lighting variability, a distinct and narrower problem than predicting a biochemically optimal harvest time [39], a distinction we revisit in Section 3.3 and the Conclusion. Several studies have applied image-based machine learning to algae related tasks other than maturity classification. Pyo et al. [3] used hyperspectral drone imagery with a 1D-CNN for pigment prediction; Shan et al. [4] and Zhang et al. [5] performed species/bloom classification using color-texture features; Guo et al. [6] and Kim et al. [7] applied remote sensing and hyperspectral imaging to bloom and species detection; and Otálora et al. [8] used an artificial neural network for microalgae species recognition. Table 1 summarizes these studies alongside the present work. None addresses the temporal, color-based classification of within-species cultivation maturity stage under variable lighting, which is the gap this study addresses.

Table 1. Comparison with prior algae-related image classification studies

Study	Task	Organism	Method	Relative to this study
Pyo et al., 2022	Pigment concentration prediction	Not species specific	Hyperspectral drone imagery + 1D-CNN	Predicts pigment level, not discrete maturity stage; requires hyperspectral sensor
Shan et al., 2020	Species/bloom recognition	Ride tide algae	Machine Learning with color microscopic imagery	Species/bloom recognition, not within-species maturity progression
Zhang et al., 2020	Species classification	Blue-green algae	Rotation invariant LBP + color features	Classifies species identity, not maturity stage
Guo, Gao, & Li, 2022	Bloom detection	Green algae	Deep learning on SAR imagery	Detects bloom presence/extent, not visual maturity stage
Kim et al., 2023	Species classification	Algae culture media	Hyperspectral imaging	Species-level classification, not temporal/maturity staging
Otálora et al., 2021	Species recognition	Multiple microalgae species	Artificial neural network	Species level recognition, not maturity classification
This study	Maturity stage classification (5 classes)	<i>Spirulina</i>	Custom CNN vs. MobileNetV2/ResNet50 (transfer learning), RGB images	Within species, color-based maturity classification with leakage-reducing validation and fair architecture comparison

Color is a common cue in computer vision applications such as medical imaging and remote sensing, though its interpretation is sensitive to illumination and acquisition conditions [9]-[11]. In digital images, color is commonly represented in the RGB space, and statistics computed over these channels can help distinguish visually distinct classes [13]. Convolutional neural networks (CNNs) are well suited to this task because they learn discriminative visual features directly from pixel data rather than relying on hand-engineered color rules [14]-[17], and have been the dominant approach for image classification since architectures such as AlexNet demonstrated the benefit of learned hierarchical features at scale [18], [19]. These methods have also been applied across diverse visual domains, including an improved ResNet50 with transfer learning for tomato pest and disease identification [34], MobileNetV2 for BISINDO sign-language recognition [35], and integrated image processing with machine learning for handwriting-shape analysis [36]. More directly relevant to the present study, CNN-based image analysis has recently been applied to *Spirulina* itself: Paranhos et al. [38] used CNNs to detect adulteration in *Spirulina* supplements from digital images, though this does not address visual

maturity-stage classification. Microalgae growth is commonly described as passing through distinct phases (lag, exponential, stationary, and death), each associated with characteristic changes in pigment and antioxidant composition as cells adapt to changing resource availability and, eventually, senescence [23], [25]. This general pattern is documented across microalgae broadly; in *Spirulina* specifically, pigment content has been shown to vary with light exposure and harvesting time [39], supporting the premise that pigment-linked color change is a meaningful signal in this organism as well. In phytoplankton more broadly, chlorophyll is known to degrade into distinct breakdown products as cell viability declines, a process studied directly in the context of cell turnover [12]. These phase dependent pigment changes motivate a color-based classification scheme, though the maturity classes used here are defined by visual labeling rather than direct biochemical measurement, a limitation discussed further in Section 2.4 and revisited in the Conclusion.

The organism studied here is *Spirulina* [20], a filamentous cyanobacterium [21] widely cultivated for its pigment-rich, protein-dense biomass and used in food, feed, and biotechnology applications [24]. Its pronounced color change across cultivation stages makes it a practical case study for color-based maturity classification. Existing algae related computer vision research emphasizes species identification, spatial detection, or bloom mapping rather than visual classification of within species cultivation stage, and the effect of lighting variability on such color-based classification has not been systematically examined. This study treats maturity staging as a five-class visual classification problem using color information alone, and evaluates model performance and robustness under this variability rather than claiming to determine a biochemically optimal moment for harvest.

To address this gap, this study evaluates a color-based CNN classification approach for *Spirulina* maturity staging using images collected across a full cultivation cycle. Beyond the classification task itself, this study contributes: a leakage-reducing evaluation protocol using deduplication and perceptual-hash-based grouping to prevent near-duplicate images from inflating cross-validated accuracy; controlled ablation isolating the individual and combined effects of CLAHE and data augmentation rather than evaluating them only in combination; a comparison across a Custom CNN, MobileNetV2, and ResNet50 under matched fine-tuning depth and architecture-appropriate preprocessing; and an interpretability analysis using Grad-CAM [33], reported as a visualization of model attention rather than evidence of the underlying biochemical mechanism.

This study has several limitations. First, maturity classification relies solely on visual/color information and does not incorporate other physiological parameters (temperature, nutrient level, light intensity, water turbidity) known to influence algae growth. Second, the current pipeline evaluates pre-cropped regions of interest and does not yet incorporate automated detection or segmentation to isolate cultivation vessels from complex backgrounds; deployment in a multi-reactor setting would currently require a fixed camera configuration or a manual cropping step. Third, interpretability analysis indicates that models are sensitive not only to color but also to spatial and textural cues from lighting and container surfaces, and, as shown by a small external field test in this study, to changes in visual context beyond the training distribution discussed further in Section 3.3. By treating maturity classification as a rigorously validated, color-based visual task rather than a proxy for biochemical harvest readiness, this study aims to provide a methodologically sound basis for automated visual monitoring of *Spirulina* cultivation, and to clarify through ablation, fair model comparison, and cautious interpretability reporting which modeling choices meaningfully affect performance and which do not.

2. MATERIALS AND METHODS

2.1. Research Design

This study follows a quantitative, image-based machine learning experimental design, not a mixed-methods design, as no qualitative component is used. Figure 1 summarizes the full pipeline. Raw images are first quality audited and deduplicated, then labeled and checked for color-space separability, before being grouped and split in a leakage-reducing manner. Model development then

proceeds as two sequential experiments rather than a single crossed design. A preprocessing ablation using the Custom CNN alone first establishes the best performing preprocessing configuration, which is then carried forward together with a matched training protocol and hyperparameters into a comparison across three architectures (the Custom CNN, MobileNetV2, and ResNet50), all evaluated under 5-fold cross-validation. Separately from this cross-validation track, one final model per architecture is retrained on a class-stratified 85/15 split of the deduplicated dataset for use in the small, unlabeled external field test and the Grad-CAM interpretability analysis reported later in this paper. This 85/15 split is not group-aware, and its own validation-set metrics (Section 2.12, Table 6) are reported for transparency only; they are not used to support any comparative claim between architectures. Cross-validated accuracy is the basis for all comparative claims between configurations and architectures, while field test predictions and Grad-CAM outputs are reported only as qualitative, exploratory observations.

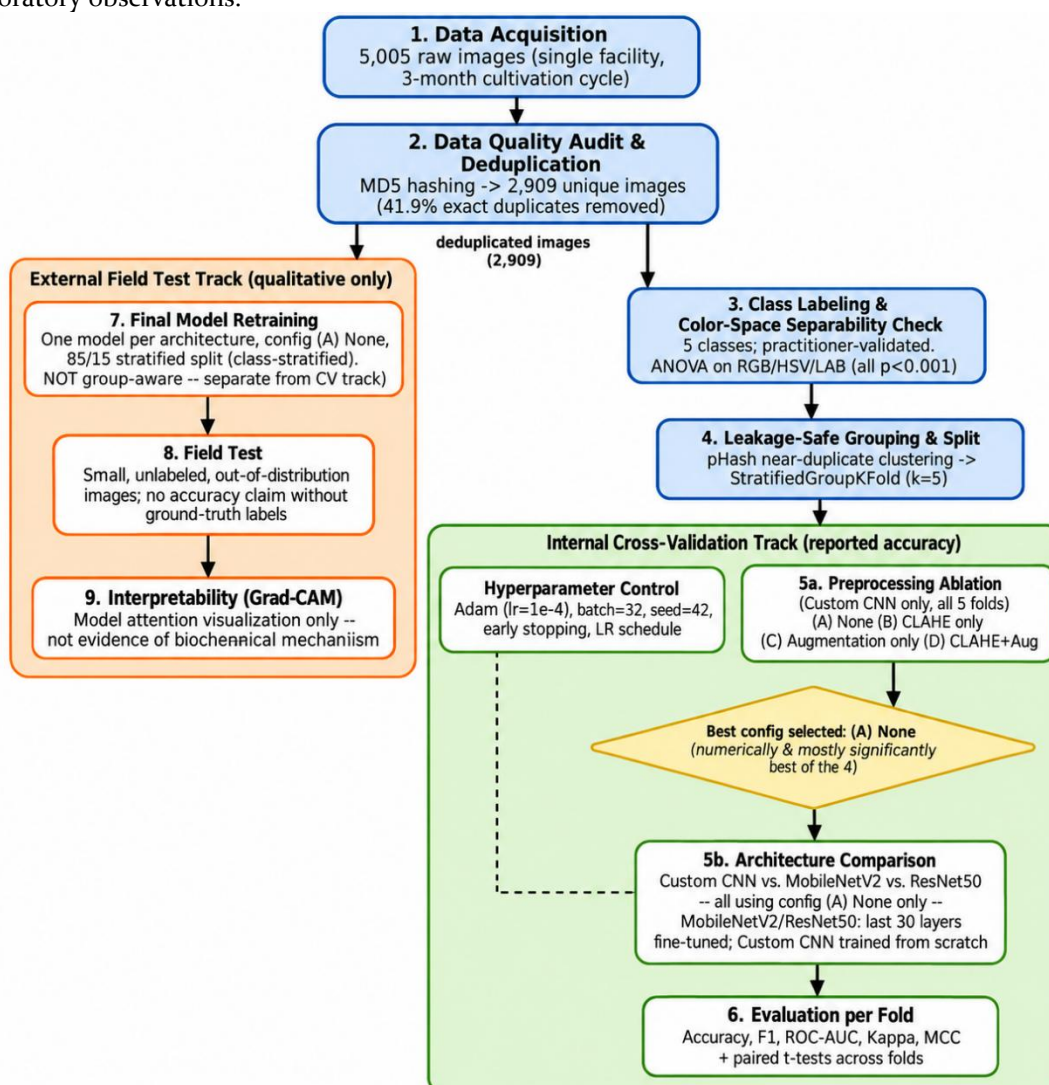


Figure 1. Research Design

2.2. Data Acquisition

Images used for model training and cross-validation were collected via direct observation of *Spirulina* cultivation at a single facility over a three-month cultivation cycle, with photography conducted each morning to reduce lighting variability. Algae were transferred from tubular photobioreactors into a

plastic container under controlled lighting prior to photography (Figure 2a). This step served two purposes: it enabled consistent, close-range image sampling, and it standardized the imaging condition across cultivation formats since real-world cultivation facilities use photobioreactors of varying physical form (e.g., glass/plastic tubular reactors, IBC [Intermediate Bulk Container]-style vessels, and open ponds) [22], so that the model would learn color-based features rather than vessel-specific visual artifacts. This protocol differs from the acquisition procedure used for the external field-test images (Section 2.12), which were instead captured in situ, directly at the photobioreactor, without sample transfer, a deliberately more demanding condition used to probe vessel-specific domain-shift risk, discussed further in Section 3.3 and the Conclusion.

Photographs were captured using a Redmi Note 11 smartphone camera with default (automatic) settings: auto exposure, auto white balance, and auto ISO, with no manual adjustment, consistent with the EXIF Model tag ("2201117TY") recorded in a sample of original site photographs during data auditing (Section 2.3). Camera resolution was not fixed or separately logged per photograph, as the device's default settings were used throughout acquisition. The dataset consists of exactly 5,005 images across five classes (dead, unripe, under-ripe, medium-ripe, ripe), stored as a mixture of JPG and PNG files, an incidental difference arising from the image editing software's export settings during cropping (Section 2.5) rather than a deliberate acquisition choice. No formal image-exclusion criteria were applied beyond the cropping step described in Section 2.5. Although image-restoration methods such as Lucy-Richardson and Wiener filtering can reduce blur in RGBA photographs [37], no deblurring was applied because it was not part of the acquisition or preprocessing protocol.

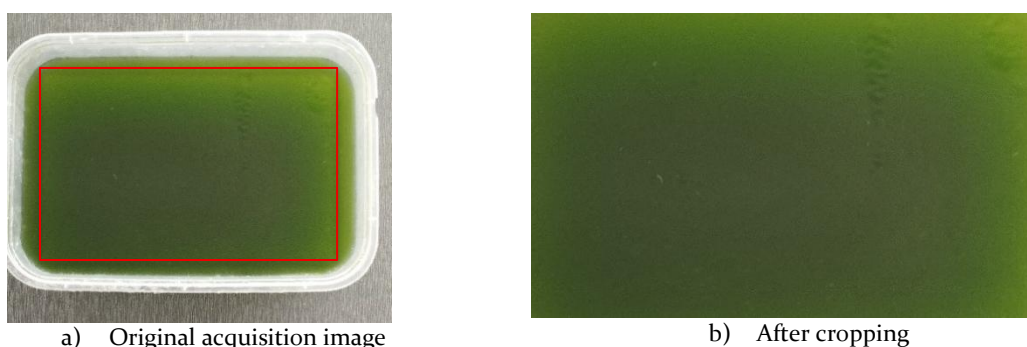


Figure 2. Algae image

2.3. Data Quality Audit and Deduplication

Before modeling, the raw dataset was audited for duplication using MD5 byte-level hashing. Of the 5,005 raw images, 2,096 (41.9%) were exact byte-level duplicates; after removing them, 2,909 unique images remained. Deduplication was not uniform across classes: the resulting class distribution (Ripe 822, Unripe 796, Dead 502, Under-ripe 419, Medium-ripe 370) is meaningfully imbalanced, in contrast to the near-uniform ~1,000-per-class distribution of the raw dataset. All subsequent preprocessing, splitting, training, and evaluation in this study use only the 2,909 deduplicated images.

2.4. Class Definition, Labeling Protocol, and Color-Space Separability

The five maturity classes are defined by visual color characteristics rather than direct biochemical measurement (e.g., chlorophyll or carotenoid concentration, cell density); this is stated explicitly as a scope limitation of the classification task (see Introduction and Discussion). Labels were reviewed by two cultivation engineers from an industry partner specializing in algae-based sustainability; the exact review protocol (independent double-labeling versus consensus discussion) was not formally documented at the time, and no inter-rater agreement statistic is available. This is noted as a limitation, and a small independent-labeling exercise to quantify inter-rater agreement is identified as a direction for future work. We note, however, that the strong color-space separability across classes

(below) and the consistently high cross-validated classification accuracy (Section 3) reflect the visual separability of the classification task itself, not the reliability of the labeling process; these two properties should not be conflated. A formal inter-rater agreement measure remains the appropriate way to assess labeling reliability, and its absence here is treated strictly as a limitation rather than something offset by model performance

To provide quantitative evidence that the five classes are visually separable in color space independent of the labeling protocol, a one-way ANOVA was computed across classes for each channel of the RGB, HSV, and LAB color spaces using the deduplicated dataset (2,909 images; Section 2.3), rather than the original, duplicate-inflated collection, so that near-duplicate images do not artificially inflate the apparent between-class separation. All nine channels showed statistically significant between-class separation (R: $F(4,2904) = 6.418$, G: $F(4,2904) = 2.658$, B: $F(4,2904) = 5.877$, H: $F(4,2904) = 2.312$, S: $F(4,2904) = 3.207$, V: $F(4,2904) = 3.067$, L: $F(4,2904) = 3.272$, A: $F(4,2904) = 3.724$, B*: $F(4,2904) = 1.512$; all $p < 0.001$), supporting the premise that color is a viable basis for distinguishing these classes, though this does not establish a biochemical mechanism.

2.5. Data Preprocessing

Each original acquisition photograph (e.g., Figure 2a) captures the sample container against a background surface; images were then cropped to isolate the algae culture region, removing container edges, lids, and background surface entirely (Figure 2b). This cropping was applied uniformly to every photograph prior to inclusion in the working dataset. The "raw dataset" of 5,005 images referenced in Section 2.2 and 2.3 therefore refers to these post-cropping images, not the uncropped originals. During revision, a code audit of the original implementation found that the CLAHE, gamma correction, and random shadow functions intended as preprocessing were defined but never connected to the training data pipeline (the `preprocessing_function` argument of the image generator was left unset), meaning the model reported as the "Proposed CNN" in the originally submitted manuscript was, in practice, trained without these enhancements. This is corrected in the present revision: CLAHE (clip limit 2.0, tile grid size 8×8, applied to the L-channel of the LAB color space) is explicitly applied via the generator's `preprocessing_function`.

Preprocessing and augmentation are applied in a fixed order, geometric/photometric augmentation, then CLAHE, then pixel rescaling (1/255), matching how Keras's `ImageDataGenerator` executes these operations. Images are resized to 224×224 pixels during loading; a code audit of the data-loading calls (`load_img` and `flow_from_dataframe`) confirmed that no interpolation argument is passed anywhere in the pipeline, so Keras's default applies: nearest-neighbor interpolation, not bilinear or bicubic resampling. Nearest-neighbor resizing does not blend pixel values across neighbors as bilinear/bicubic methods do, so it introduces no interpolated color values absent from the original photograph, at the cost of coarser, blockier edges. We report this as the pipeline's actual, verified behavior rather than an assumed default. Standard augmentation, when enabled, consists of rotation ($\pm 10^\circ$), width/height shift ($\pm 15\%$), zoom ($\pm 10\%$), horizontal flip, brightness scaling (0.9–1.1×), and shear ($\pm 15\%$); a new random parameter value within each stated range is drawn independently per training image per epoch, with horizontal flipping applied at 50% probability. Augmentation is generated on the fly at training time and applied only to the training partition of each cross-validation fold (Section 2.6), never before the leakage-reducing split and never to validation data. Gamma correction and random shadow augmentation, present in the original implementations (non-functional) preprocessing code, were not reintroduced in this revision and are outside the scope of the present ablation (Section 2.9); we note this as a deliberate scope decision rather than an oversight.

2.6. Leakage-Reducing Cross-Validation Protocol

Because images were collected across a continuous three-month cultivation cycle, near-duplicate images from the same acquisition session are likely highly correlated, and the naive random split used in the original submission (70:30, per-image, no deduplication or grouping) risks such near-duplicates spanning both training and evaluation partitions, inflating reported accuracy. Reconstructing exact acquisition sessions from EXIF metadata proved infeasible: the cropping step strips EXIF data from

this dataset's images, and the separately archived original (uncropped) photographs, which do retain EXIF timestamps, cover only a small, non-representative fraction of the acquisition period and do not map one-to-one to the full working dataset. As a substitute grouping signal, perceptual hashing (pHash) clustered near-duplicate images (Hamming distance ≤ 5) via union-find, and these clusters served as groups in a StratifiedGroupKFold (5-fold, class-stratified, group-aware) scheme, ensuring no near-duplicate cluster spans more than one fold. Perceptual-hash clustering is an approximate proxy for true acquisition sessions and may not capture every within-session correlation (e.g., same-session images with substantial visual variation could fall outside the similarity threshold), but it directly addresses the specific leakage risk identified during data auditing (Section 2.3). All accuracy figures in Section 3 are mean $\pm$ standard deviation across these 5 folds. Within each fold, early stopping and the learning-rate reduction schedule (Section 2.8) monitor validation loss on that fold's own held-out evaluation partition, the same data later used to compute that fold's reported metrics, rather than a separate split carved from the training partition. This means training-epoch count is implicitly chosen using each fold's evaluation data, a mild model-selection leakage distinct from the near-duplicate-image leakage addressed above. It applies identically to every configuration and architecture compared here, so relative comparisons remain fair, but absolute accuracy figures may be modestly optimistic as a result; we disclose this as a limitation rather than correct it retroactively, since doing so would require retraining every model under a revised protocol.

2.7. Model Architectures

The Custom CNN (Table 2) was designed as a compact, from-scratch classifier appropriate to a well-separated, low-complexity color classification task (Section 2.4 shows strong statistical separability across all nine RGB/HSV/LAB channels), rather than adapting an existing large-scale architecture whose additional representational capacity is not obviously required for this task. It totals 439,685 parameters (438,789 trainable), $5.9\times$ fewer than MobileNetV2 and $54.8\times$ fewer than ResNet50 (Section 3), following standard design conventions for compact CNNs: batch normalization [26] and dropout [27] are included as regularization following established practice, though their individual contribution was not separately ablated in this study and is not claimed as empirically demonstrated here. Max pooling is used in the first two blocks to preserve the strongest local color/edge responses at early layers, while average pooling is used in the third block to smooth the higher-level feature map before global pooling, favoring overall chromatic distribution over precise spatial localization; this design choice is a theoretical rationale rather than an empirically ablated one.

Table 2. Custom CNN architecture

Layer	Type	Kernel/Pool	Activation	Output Size	Params
Input	-	-	-	224×224×3	0
Conv2D_1	Conv2D, same padding, stride 1	3×3×64	ReLU	224×224×64	1,792
BatchNorm_1	-	-	-	224×224×64	256
MaxPool_1	MaxPooling2D, stride 2	2×2	-	112×112×64	0
Dropout_1(0.25)	-	-	-	112×112×64	0
Conv2D_2	Conv2D, same padding, stride 1	3×3×128	ReLU	112×112×128	73,856
BatchNorm_2	-	-	-	112×112×128	512
MaxPool_2	MaxPooling2D, stride 2	2×2	-	56×56×128	0
Droupour_2(0.30)	-	-	-	56×56×128	0
Conv2D_3	Conv2D, same padding, stride 1	3×3×256	ReLU	56×56×256	295,168
BatchNorm_3	-	-	-	56×56×256	1,024
AvgPool_3	AveragePooling2D, stride 2	2×2	-	28×28×256	0
Dropout_3(0.35)	-	-	-	28×28×256	0
GlobalAvgPool	-	-	-	256	0
Dense_1	Dense	-	ReLU	256	65,792

Layer	Type	Kernel/Pool	Activation	Output Size	Params
Dropout ₄ (0.5)	-	-	-	256	0
Output	Dense	-	Softmax	5	1,285
Total					439,685 (438,789 trainable)

For comparison, MobileNetV2 [28] and ResNet50 [29] were selected to span a range of model capacity and design philosophy relevant to practical deployment: MobileNetV2 represents a lightweight, mobile-oriented architecture using depthwise-separable convolutions, while ResNet50 represents a substantially deeper, higher-capacity architecture using residual connections; together they provide a meaningful capacity/efficiency contrast against the Custom CNN, rather than being presented as unqualified "state-of-the-art" baselines. Both are ImageNet-pretrained (include_top=False) with an identical fine-tuning protocol: the last 30 layers are unfrozen, all earlier layers remain frozen, and both use the same classification head (GlobalAveragePooling2D→Dense(256, ReLU)→Dropout(0.5)→Dense(5, Softmax)). Each pretrained model uses its architecture-specific preprocessing function (mobilenet_v2.preprocess_input, resnet50.preprocess_input) at both training and inference time, correcting an inconsistency identified in the original implementation, where ResNet50 inference used generic 1/255 rescaling instead of its trained preprocessing, a mismatch that plausibly contributed to its previously reported poor field performance. Total parameters: MobileNetV2 2,587,205 (1,855,621 trainable); ResNet50 24,113,541 (14,976,005 trainable).

All three architectures operate on standard RGB input; CLAHE (Section 2.5), where enabled, is applied specifically in LAB space to modify local contrast on the lightness channel without altering hue information, rather than transforming the model's input representation to a non-RGB color space. Given that all nine channels across RGB, HSV, and LAB showed statistically significant class separability (Section 2.4), no color-space transformation of the primary model input was pursued as an additional architectural variable in this study. Maximum training length (25 epochs) is governed by early stopping on validation loss (Section 2.8) rather than a fixed, arbitrarily chosen epoch count, so that each model or fold trains until convergence within a bounded budget rather than for a predetermined duration.

2.8. Training Configuration and Reproducibility

All models were trained with the Adam optimizer (learning rate 1×10^{-4}), batch size 32, for a maximum of 25 epochs, with early stopping (monitoring validation loss, patience 5, restoring best weights) and a learning-rate reduction schedule (factor 0.2, patience 3, minimum learning rate 1×10^{-6}). Early stopping in particular was configured but not actually included in the training callbacks in the original implementation, and is corrected here. A fixed random seed (42) is applied consistently across Python's hash seed, NumPy, Python's random module, and TensorFlow, for every fold, configuration, and model. Training was performed on Google Colab using an NVIDIA Tesla T4 GPU (driver 580.82.07, CUDA 13.0), with TensorFlow 2.20.0 and Keras 3.13.2. All layers use Keras/TensorFlow default weight initialization (Glorot-uniform for convolutional and dense kernels, zero-initialized biases); the model is trained with categorical cross-entropy loss against the softmax output, and no class weighting is applied.

2.9. Ablation Study

To find out how much each method CLAHE and augmentation contributes on its own and together, four different ways of preparing the data were tested. All used the same model setup, training process, and way of checking performance. The methods compared were: no preparation at all, using CLAHE alone, using augmentation alone, and using both CLAHE and augmentation together. Gamma correction and random shadow augmentation, which were part of the original (non-functional) preprocessing description, are not included in this ablation study (Section 2.5). Because this task is based on color, the levels of image changes used in the "augmentation only" and "CLAHE with augmentation" setups were kept gentle especially the brightness change ($\pm 10\%$) is smaller than usual, because using stronger color changes could make an image look like it belongs to a different color

group (for example, a lightly colored sample appearing to belong to a more mature class), which would change what the image is labeled as instead of just adding useful variety.

2.10. Statistical Validation

Model and configuration comparisons in this study are conducted within the 5-fold cross-validation protocol described in Section 2.6, not a single held-out split, which reduces the chance that one split overestimates performance or obscures variability, though some risk remains. For pairwise comparisons both between preprocessing configurations (Section 2.9) and between architectures (Section 2.7), we perform a paired t-test on the per-fold accuracy values, since each comparison is based on the same five folds under two different conditions; we did not use McNemar's test, which compares prediction outcomes from two classifiers on a shared test set, because it does not suit our fold-based design. As an additional non-parametric sensitivity check, we computed the Wilcoxon signed-rank test on the same paired per-fold values used for the paired t-test. With only five paired folds, its minimum achievable p-value is 0.0625, attained only when all five folds agree in direction, so it cannot reach $p < 0.05$ regardless of outcome. We therefore treat a Wilcoxon result at this floor value, rather than a nominal significance threshold, as the relevant non-parametric signal, reported alongside the paired t-test in Section 3.2.

The use of five folds limits our power. When a p-value is close to 0.05 but does not go below it, we describe this as a non-significant numerical trend rather than claiming the two conditions perform the same. We also report the standard deviation across the five folds throughout the analysis. This allows readers to see the variability instead of relying on a single point estimate. We apply this cautious approach in the Results section. A model does not become reliable just because it shows trends across different metrics. We reserve the term "reliable" for cases where the statistical tests or variability measures provide support. Section 2.11 explains how we calculate ROC-AUC.

2.11. Evaluation Metrics and Computational Complexity

Each model and configuration is evaluated per fold using accuracy, macro-averaged precision, recall, and F1-score, multiclass ROC-AUC (one-vs-rest formulation, macro-averaged, following the extension described by Hand and Till [32]), Cohen's Kappa [30], and Matthews correlation coefficient (MCC) [31]. We rely on macro-averaged and chance-corrected metrics rather than accuracy alone because, as reported in Section 2.3, the deduplicated dataset is not perfectly balanced across the five maturity classes (370–822 images per class); these metrics weight each class equally regardless of its support, giving a fairer picture of per-class performance than accuracy can provide on its own. Beyond the macro-averaged summary values, we report per-class precision, recall, and F1-score together with the full confusion matrix, so that misclassification patterns between visually adjacent maturity stages can be examined directly rather than inferred from an aggregate score. We note that macro-averaged, one-vs-rest ROC-AUC can read as very high even for a model with comparatively lower accuracy, since each one-vs-rest sub-problem is generally easier to separate than the full five-class task; for this reason, we report ROC-AUC alongside, and never as a substitute for, accuracy and the per-class metrics above.

Computational complexity is reported in Table 5 using total and trainable parameter counts, FLOPs, average training time per fold, and per-image inference latency, all measured on the Tesla T4 GPU environment described in Section 2.8; FLOPs are computed analytically from the convolutional and dense layers, counting each multiply-accumulate operation as two floating-point operations, and the counter recurses into nested sub-models and handles DepthwiseConv2D to avoid undercounting the pretrained backbones. For the proposed Custom CNN, this totals 439,685 parameters (438,789 trainable), approximately 3.88 GFLOPs, an average training time of approximately 282 seconds per fold over 25 epochs, and an average inference latency of 4.73ms per image. MobileNetV2 (2,587,205 parameters; 1,855,621 trainable) requires approximately 0.60 GFLOPs, trains in approximately 186 seconds per fold, and reaches an inference latency of 18.18ms per image; ResNet50 (24,113,541 parameters; 14,976,005 trainable) requires approximately 7.72 GFLOPs, trains in approximately 213 seconds per fold, and reaches an inference latency of 16.44ms per image. Despite having roughly 5.9 times more parameters than the Custom CNN, MobileNetV2's depthwise-separable convolutions give

it substantially fewer FLOPs (0.60 GFLOPs versus 3.88 GFLOPs); ResNet50, by contrast, is both larger and more computationally expensive on every axis reported.

2.12. Field Test Protocol

Beyond the 5-fold cross-validation described in Section 2.6, we test the final models on six field images, drawn from six photobioreactor units across four customer sites, rather than the single facility used for the training and validation dataset (Section 2.2). Container geometry differs across the four sites, and so does the background visible around the algae itself, such as structural framing around the vessel. Each image therefore captures a distinct, previously unseen acquisition context, not a repeat of the same condition six times over. Given the sample size, we treat this field test strictly as a qualitative interpretability probe, not a validation or generalization study, and we skip reporting an accuracy figure for it. A proportion computed from six images still carries an uncertainty interval too wide to support a claim about generalization, regardless of the outcome. We compare instead the predictions and Grad-CAM attention maps of the proposed Custom CNN and the two baseline architectures, MobileNetV2 and ResNet50, on each image.

We find something here that matters more for deployment readiness than for ranking the models: in the field images where background structure is visible, all three architectures, including our own, show Grad-CAM activation spreading into that background rather than staying confined to the algae. We treat this as a shared limitation of the current preprocessing and cropping pipeline under uncontrolled field framing, not as evidence that one architecture handles it more robustly than the others. Telling genuine shortcut learning apart from incidental attention spread requires controlled perturbation experiments, such as masking or substituting the background region; that work falls outside the present study's scope, and we flag it for future work in the Conclusion.

3. RESULTS AND DISCUSSION

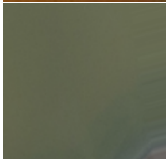
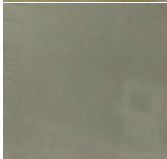
3.1. Dataset Overview

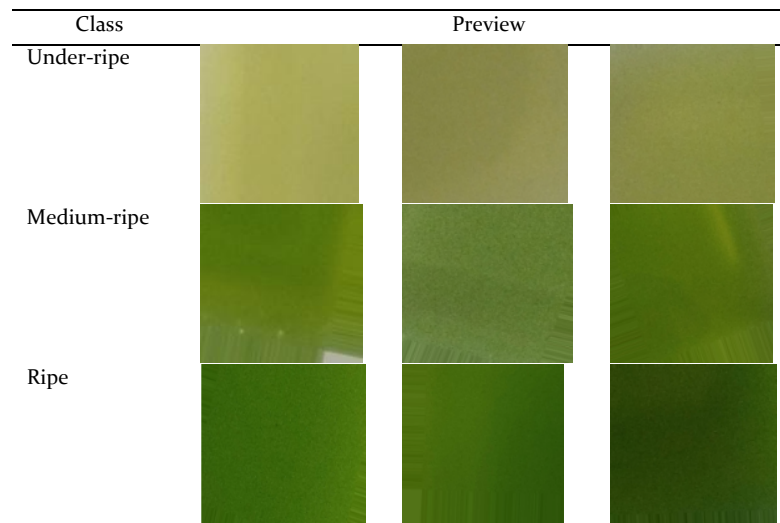
We collected the dataset by observing algae growth from initial cultivation through full maturity and eventual degradation. Image acquisition followed the protocol in Section 2.2, and preprocessing followed Section 2.5. Each image was cropped manually to isolate the algae region of interest, reducing background interference and isolating the dominant color characteristics that matter for maturity classification.

The dataset spans five maturity classes, each defined by observable color characteristics:

- Dead: red, orange or brownish-yellow coloration
- Unripe: white, transparent, or very pale green
- Under-ripe: light green
- Medium-ripe: moderate green intensity
- Ripe: strong to dark green coloration

Table 3. Representative samples from each class

Class	Preview		
Dead			
Unripe			



The results in this section come from the 2,909 deduplicated images described in Section 2.3, not the original 5,005-image collection. Deduplication left the five classes uneven: Ripe (822), Unripe (796), Dead (502), Under-ripe (419), Medium-ripe (370). That imbalance is why Section 3.2 leans on macro-averaged and chance-corrected metrics rather than accuracy alone, for the reasons given in Section 2.11. Gradual color transitions between adjacent stages also blur the boundaries between neighboring classes, a source of realistic classification difficulty that shows up in the confusion patterns discussed in Section 3.2.

3.2. Model Performance

Sections 2.6, 2.9, and 2.10 established the group-aware, leakage-reducing 5-fold cross-validation protocol as the basis for all comparative claims between preprocessing configurations and architectures. Table 4 reports the preprocessing ablation results across the four configurations described in Section 2.9. Combining CLAHE with augmentation, the configuration used in the originally submitted manuscript, was numerically the weakest of the four; this difference was statistically significant against CLAHE alone ($p = 0.021$) but not against the plain, no-preprocessing configuration at this sample size ($p = 0.085$). A Wilcoxon sensitivity check (Section 2.10) found only the CLAHE-alone versus CLAHE-plus-augmentation comparison reached the test's minimum p-value (0.0625, all five folds favoring CLAHE alone), mirroring the sole comparison significant under the paired t-test; no other pairwise comparison in Table 4 approached this floor under either test.

Table 4. Preprocessing Ablation Results (5-fold Cross-Validation)

Configuration	Accuracy	F1 (macro)	ROC-AUC	Kappa	MCC
No preprocessing	0.9743 ± 0.0089	0.9719 ± 0.0104	0.9986 ± 0.0008	0.9668 ± 0.0115	0.9672 ± 0.0112
CLAHE only	0.9734 ± 0.0110	0.9714 ± 0.0119	0.9991 ± 0.0004	0.9658 ± 0.0142	0.9662 ± 0.0137
Augmentation only	0.9647 ± 0.0209	0.9605 ± 0.0238	0.9987 ± 0.0008	0.9544 ± 0.0271	0.9557 ± 0.0257
CLAHE + Augmentation	0.9474 ± 0.0215	0.9401 ± 0.0263	0.9979 ± 0.0017	0.9319 ± 0.0282	0.9333 ± 0.0268

Table 5 reports the architecture comparison results under the same 5-fold cross-validation protocol, using the best-performing preprocessing configuration (no preprocessing) carried forward from Table 4, as described in Section 2.7. ResNet50 achieved the highest accuracy, significantly outperforming the proposed Custom CNN (paired t-test, $p = 0.024$); MobileNetV2 was both less accurate on average and far less stable across folds. A Wilcoxon sensitivity check (Section 2.10) found all three pairwise

comparisons reached the test's minimum p-value (0.0625): ResNet50 outperformed the Custom CNN, the Custom CNN outperformed MobileNetV2, and ResNet50 outperformed MobileNetV2, in every one of the five folds. None nominally cross $p < 0.05$, but this reflects the test's structural floor at this sample size rather than an inconsistent effect, and we interpret it as corroborating the paired t-test results above.

The Custom CNN's practical case rests on efficiency: it uses roughly 54.8 times fewer parameters than ResNet50 and completes inference roughly 3.5 times faster per image. This accuracy-efficiency trade-off has direct deployment implications: cloud-based monitoring can absorb ResNet50's larger footprint for its higher accuracy, while edge deployment with limited compute favors the Custom CNN's smaller footprint and faster inference at a modest accuracy cost (0.9706 versus 0.9952). FLOPs alone do not predict this well: MobileNetV2 has the fewest FLOPs (0.60 GFLOPs) yet the slowest measured latency (18.18 ms, versus ResNet50's 16.44 ms), since depthwise-separable convolutions are not always executed efficiently on GPU hardware, which is why we rely on measured latency (Section 2.11) rather than FLOPs alone. MobileNetV2's instability, collapsing below 70% accuracy in two of five folds, is a further deployment concern independent of its average accuracy; plausible contributors include its depthwise-separable layers responding less predictably to the shared fine-tuning protocol (Section 2.7) and the limited training set (2,909 images), though confirming this would require dedicated experiments beyond this study's scope. A model that occasionally collapses is unsuitable for unattended monitoring regardless of FLOPs or latency, reinforcing that average accuracy alone is an incomplete basis for architecture selection.

Table 5. Architecture Comparison Results (5-fold Cross-Validation)

Model	Accuracy	F1 (macro)	ROC- AUC	Kappa	MCC	Params	FLOPs (G)	Train Time (s/fold)	Inference (ms/img)
Custom CNN	0.9706 ± 0.0139	0.9682 ± 0.0155	0.9987 ± 0.0007	0.9622 ± 0.0180	0.9627 ± 0.0171	439,685	3.88	282.1	4.73
MobileN etV2	0.8226 ± 0.1605	0.7796 ± 0.2027	0.9691 ± 0.0370	0.7742 ± 0.2026	0.7838 ± 0.1899	2,587,205	0.60	186.1	18.18
ResNet50	0.9952 ± 0.0018	0.9946 ± 0.0021	1.0000 ± 0.0000	0.9938 ± 0.0023	0.9939 ± 0.0023	24,113,541	7.72	213.1	16.44

Section 2.12 also produced one retrained model per architecture on an 85/15 train/validation split, separate from the 5-fold protocol above. Table 6 reports accuracy, macro precision, macro recall, macro F1, and macro one-vs-rest ROC-AUC for these three models on their held-out validation set (437 images). We report this separately from Tables 4 and 5 to keep the different evaluation protocols distinct.

Table 6. Validation Performance (85/15 split, Section 2.12)

Model	Accuracy	Macro Precision	Macro Recall	Macro F1- Score	ROC-AUC
Custom CNN	0.9817	0.9975	0.9751	0.9804	0.9973
MobileNetV2	0.9611	0.9595	0.9505	0.9545	0.9970
ResNet50	0.9954	0.9934	0.9967	0.9950	1.0000

Because this is a single, non-cross-validated split rather than the 5-fold protocol, these figures are reported for transparency and to support the class-level breakdown and Grad-CAM analysis that follow, not as an independent comparative claim; on this split, ResNet50 numerically scores highest on every metric, consistent with the cross-validated comparison already reported in Table 4, with the Custom CNN and MobileNetV2 trailing at similar magnitude but with different error patterns. Table 7 breaks this down by class using precision, recall, and F1-score.

Table 7. Per-Class Precision, Recall, and F1-Score (85/15 split validation)

Class	Custom CNN			MobileNetV2			ResNet50		
	Prec.	Recall	F1-Score	Prec.	Recall	F1-Score	Prec.	Recall	F1-Score
Dead	1.000	0.987	0.993	1.000	0.920	0.958	1.000	1.000	1.000
Medium-ripe	1.000	1.000	1.000	0.963	0.929	0.945	0.982	1.000	0.991
Ripe	1.000	1.000	1.000	0.992	1.000	0.996	1.000	0.992	0.996
Under-ripe	1.000	0.889	0.941	0.906	0.921	0.913	0.984	1.000	0.992
Unripe	0.938	1.000	0.968	0.937	0.983	0.959	1.000	0.992	0.996

The aggregate numbers in Table 6 hide real differences at the class level. The Custom CNN scores a perfect F1 on Medium-ripe and Ripe but drops to 0.941 on Under-ripe: its confusion matrix shows 7 of 63 Under-ripe images misclassified as Unripe, with no other confusion of any size. MobileNetV2 spreads its errors more widely, including several Dead images misclassified across Ripe, Under-ripe, and Unripe. ResNet50 makes only two errors across all 437 validation images, one Ripe misclassified as Medium-ripe and one Unripe misclassified as Under-ripe (Figure 3). Most substantial confusions occurred between adjacent maturity classes, although MobileNetV2 also produced several non-adjacent errors involving the Dead class, consistent with the gradual color transitions noted in Section 3.1 rather than scattered or unrelated error.

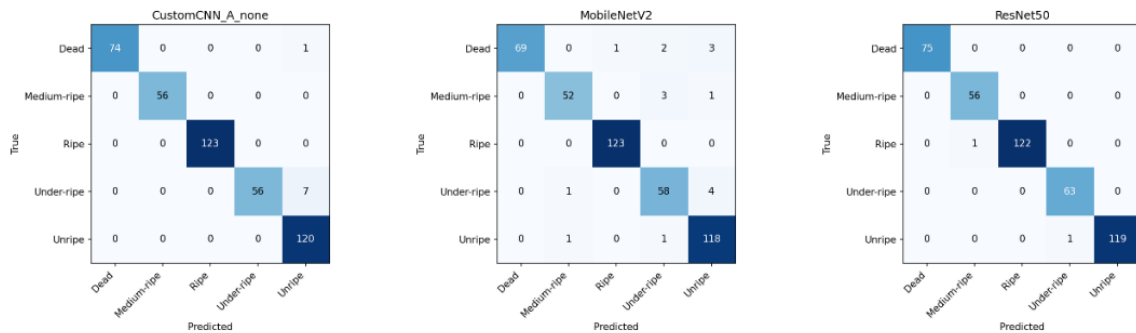


Figure 3. Confusion matrices for the three architectures on the 85/15 validation split (n=437)

Training behavior also differed across architectures (Figure 4). The Custom CNN's validation loss stayed flat near random-guess accuracy for the first five epochs before dropping sharply and tracking training loss closely from epoch six onward. MobileNetV2 reached near-perfect training accuracy within a few epochs, but its validation accuracy climbed more slowly, leaving a persistent gap between the two curves through most of the run. ResNet50's training and validation curves tracked each other closely from early on, with one small dip around epoch eight before both stabilized near 1.0.

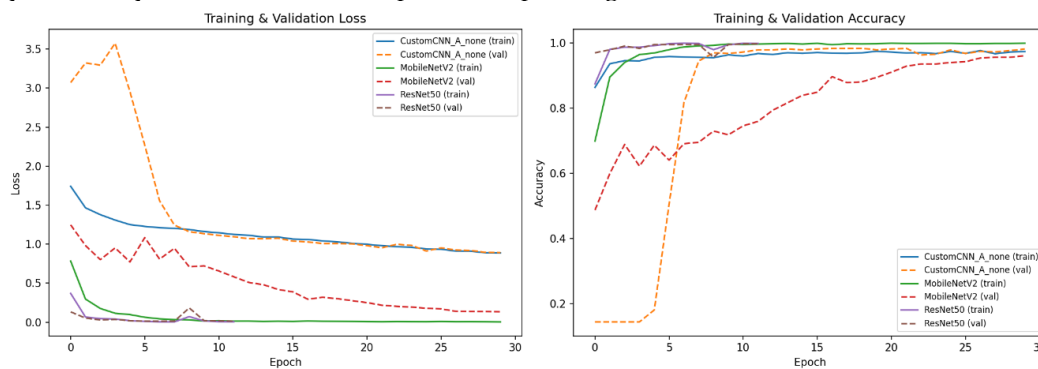


Figure 4. Training and validation loss (left) and accuracy (right) across epochs for the three architectures

3.3. Field Testing & Interpretability

Table 8 lists each model's prediction and confidence on the six field images, using the retrained models from Section 2.12. As established there, no ground-truth label exists for these images, so we treat these as predictions for qualitative comparison rather than correct or incorrect classifications.

Table 8. Field Test Predictions

Image	Custom CNN	MobileNetV2	ResNet50
1757870162110.jpg	Under-ripe (42.3%)	Unripe (73.5%)	Unripe (96.8%)
1757870162133.jpg	Dead (99.5%)	Unripe (61.5%)	Dead (47.8%)
1758381044722.jpg	Unripe (81.4%)	Dead (90.9%)	Dead (90.8%)
1758381044736.jpg	Dead (98.4%)	Dead (99.2%)	Dead (96.2%)
1758381044937.jpg	Ripe (73.3%)	Dead (61.9%)	Dead (65.2%)
1758381045020.jpg	Dead (73.6%)	Dead (80.6%)	Dead (89.9%)

We ran this evaluation twice under an identical protocol and the same random seed, once during development and once for the analysis reported here. Several predictions changed between the two runs: ResNet50 called two of the six images Ripe in the first run and Dead in the second, and the Custom CNN and MobileNetV2 shifted on individual images as well. Only one image, 1758381044736.jpg, drew the same prediction from all three models in both runs. We take this instability as a further reason to treat the field test as a qualitative probe rather than a source of generalization claims, on top of the small sample size already noted in Section 2.12.

Grad-CAM inspection across all six field images shows several specific, repeatable patterns rather than random noise, discussed below in the same order as Table 8. In image 1757870162110.jpg, which shows four containers with markedly different fill levels and colors, the Custom CNN's attention stays reasonably well confined to the liquid-filled interior of each tube, most strongly on the rightmost container; its comparatively low confidence (Under-ripe, 42.3%) plausibly reflects genuine visual ambiguity across differently filled containers rather than an attention failure. ResNet50, by contrast, places its strongest activation at the bottom-right corner, on the container's support structure rather than the liquid itself (Figure 5).

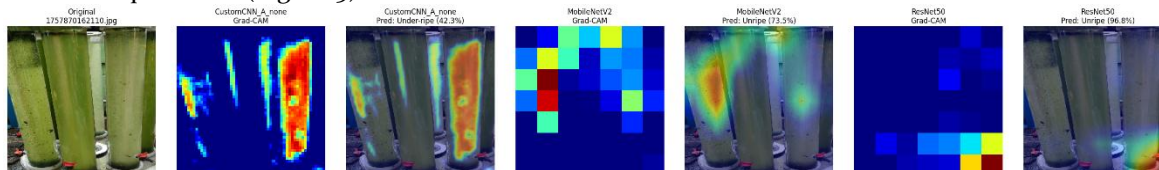


Figure 5. Grad-CAM comparison on a field image with four differently filled containers (1757870162110.jpg), showing the Custom CNN's attention staying reasonably confined to the liquid while ResNet50 attends to the support structure

In image 1757870162133.jpg, which shows small red sensor clips near the base of the photobioreactor tubes, the Custom CNN's attention concentrates precisely on the two clips in both runs, rather than on the algae-colored liquid beside them (Figure 6).

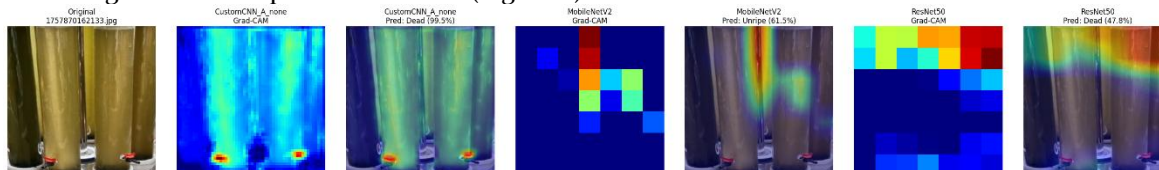


Figure 6. Grad-CAM comparison on a field image with visible sensor clips (1757870162133.jpg), showing the Custom CNN attending to the clips rather than the algae

A related pattern shows up in the two images captured through cutout wood boards. In image 1758381044722.jpg, the Custom CNN's activation traces the geometric edges of the cutouts rather than spreading evenly across the visible liquid, an edge-following response that Grad-CAM alone cannot distinguish from genuine color-based reasoning (Figure 7).

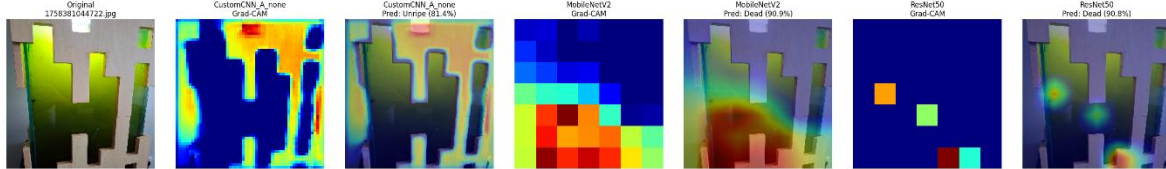


Figure 7. Grad-CAM comparison on a field image captured through a cutout wood board (1758381044722.jpg), showing the Custom CNN tracing the cutout edges

The same edge-following pattern recurs in image 1758381044736.jpg, where the Custom CNN's activation again traces the cutout boundaries rather than the liquid itself (Figure 8).



Figure 8. Grad-CAM comparison on a second field image captured through a cutout wood board (1758381044736.jpg), showing the same edge-following activation pattern

In image 1758381044937.jpg, the Custom CNN's attention forms a sharp V-shaped band tracing the wooden support notch and liquid-surface line at the top of the container, rather than spreading across the bulk of the green liquid below another instance of the same edge-following behavior; ResNet50 instead concentrates on an isolated bright spot near the right edge of the frame, away from the algae entirely (Figure 9).

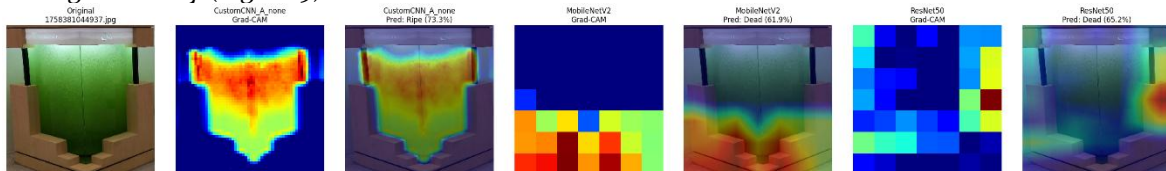


Figure 9. Grad-CAM comparison on a field image showing a V-shaped container notch (1758381044937.jpg), with the Custom CNN's attention tracing the notch and surface line rather than the liquid

In image 1758381045020.jpg, which shows a wood corner brace beside an algae panel, all three architectures direct some attention onto the brace rather than confining it to the panel, again in both runs, with ResNet50's peak activation sitting almost entirely on the wood (Figure 10).

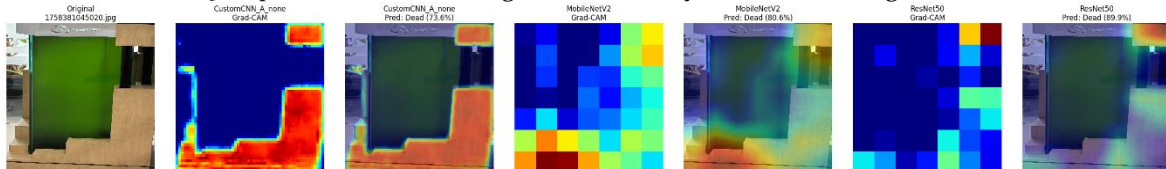


Figure 10. Grad-CAM comparison on a field image with a visible wood corner brace (1758381045020.jpg), showing all three architectures attending partly to the brace

We treat these patterns as evidence of a shared limitation in how the current preprocessing and cropping pipeline handles uncontrolled field framing, not as evidence that one architecture is more robust than another: every model tested, including the one proposed here, attends to non-algae

elements when such elements sit in the frame. Image 1757870162110.jpg is the one partial exception, where the Custom CNN's attention stayed comparatively well confined to the algae itself.

4. CONCLUSION

This study built and tested a lightweight Custom CNN for five-class algae maturity classification from color images, using a leakage-reducing 5-fold cross-validation protocol. Ablation testing across four preprocessing configurations found that neither CLAHE nor data augmentation, alone or combined, beat the plain configuration; combining both significantly underperformed it. Preprocessing choices mattered here, though the direction reversed what we expected going in. In the architecture comparison, ResNet50 reached the highest cross-validation accuracy (0.9952), beating the proposed Custom CNN (0.9706) by a margin the paired t-test confirmed as significant ($p = 0.024$). MobileNetV2 trailed on both counts: lower accuracy on average, and far less stable across folds, collapsing below 70% in two of the five. The principal advantage of the Custom CNN lies in computational efficiency rather than classification accuracy. It runs on roughly 54.8 times fewer parameters than ResNet50 and finishes inference about 3.5 times faster (4.73 ms versus 16.44 ms per image), at a real but modest cost in accuracy. Per-class analysis on the separate 85/15 validation split showed that most errors, across all three architectures, fell at boundaries between adjacent maturity classes: Under-ripe/Unripe, Medium-ripe/Under-ripe, Ripe/Medium-ripe. That pattern fits the gradual color transitions the classification scheme relies on. MobileNetV2 broke the pattern in one respect: several of its errors confused Dead samples with Ripe or Unripe, classes nowhere near Dead in the maturity order, so boundary adjacency alone doesn't explain its mistakes. We field-tested six images drawn from six photobioreactor units across four customer sites, with no ground-truth labels available, and treated this as a qualitative interpretability probe rather than a generalization test. Predictions shifted even across repeated runs of the identical training protocol on the same seed. Grad-CAM inspection showed all three architectures, including the one we propose, sometimes fixed on non-algae elements in the frame: mounting hardware, sensor clips, structural braces. We read this as a limitation of the current preprocessing and cropping pipeline under uncontrolled field framing, shared across architectures rather than specific to any single model. The models learned visual patterns tied to the five maturity classes we defined; that is not confirmation of the underlying chlorophyll or carotenoid biochemistry, which this study never measured directly. A few limitations shape how to read these results, not as afterthoughts but as part of the finding. All training and validation images came from a single cultivation facility, so reported accuracy may partly reflect that facility's conditions rather than features that generalize across sites; the six field-test images, spread across six photobioreactor units at four customer sites, give only a preliminary and statistically limited check on that question. Labels were reviewed by two cultivation engineers from an industry partner specializing in algae-based sustainability, with no formal inter-rater agreement statistic behind them. Deduplication left the training data unevenly distributed across classes. Early stopping's monitored validation loss, within each cross-validation fold, is drawn from that fold's own held-out evaluation partition rather than a separate carved-out split, a mild model-selection leakage (Section 2.6) that likely makes the reported accuracy figures modestly optimistic, though identically so across every model and configuration compared. We captured the training images after transferring algae samples into a standardized container, a deliberate choice for consistent acquisition; the field-test images, by contrast, were captured in situ, including at the one site that ran two separate units, but other vessel geometries, such as flat-panel reactors or IBC containers photographed in place, remain untested. Finally, this study measures image classification accuracy. It does not show that classification output improves harvest-timing decisions, biomass yield, or cultivation productivity, and we make no claim of an operational harvesting solution. Future work should build a larger, independently labeled external test set spanning multiple cultivation sites and photobioreactor types, sized to support a real generalization claim rather than a qualitative probe. Having independent annotators double-label a sample of the data would let us report inter-rater agreement directly instead of assuming it. Controlled perturbation experiments, masking or swapping out background regions, would separate genuine shortcut learning

from the incidental attention spread we observed here. The accuracy gap to ResNet50 is real, but so are the computational savings, which makes knowledge distillation from a larger model into a compact one worth exploring for edge deployment. Finally, we still need to connect classifier output to an actual harvest-timing decision rule and test that connection against real cultivation outcomes before anyone can claim a practical harvesting benefit.

DECLARATIONS

AI USAGE STATEMENT

AI-Assisted Language Editing: AI-assisted tools (DeepL Translator, <https://www.deepl.com/en/translator>; and QuillBot, <https://quillbot.com>) were used solely to support English translation, grammar checking, and language editing during manuscript preparation. The authors remain fully responsible for the accuracy, originality, interpretation, and final content of the manuscript.

AUTHOR CONTRIBUTION

Dicky Hermawan's contributions included conceptualization, methodology development, software development, validation, formal analysis, research resource preparation, data collection, original draft writing, and project administration.

Meanwhile, Hasanul Fahmi contributed to the development of methodology, validation of research results, review and editing of writing, project supervision, and funding procurement.

FUNDING STATEMENT

This work was funded and supported by the Faculty of Artificial Intelligence & Frontier Technologies, UNITAR International University.

CONFLICTING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

DATA AVAILABILITY STATEMENT

The image dataset supporting the findings of this study is publicly available on GitHub at <https://github.com/dhermawan215/classification-algae-neural-network/tree/main/Datasets>.

The training and evaluation code will be made available in the same repository upon publication.

REFERENCES

- [1] H. Mundhada et al., "Fruit detection and three-stage maturity grading using CNN," *Int. J. Next-Gener. Comput.*, vol. 14, no. 1, 2023, doi: 10.47164/ijngc.v14i1.1099.
- [2] L. Chuquimarca, B. Vintimilla, and S. Velastin, "Banana ripeness level classification using a simple CNN model trained with real and synthetic datasets," *arXiv:2504.08568*, 2025, doi: 10.48550/arXiv.2504.08568.
- [3] J. Pyo et al., "Drone-borne sensing of major and accessory pigments in algae using deep learning modeling," *GISci. Remote Sens.*, vol. 59, no. 1, pp. 310–332, 2022, doi: 10.1080/15481603.2022.2027120.
- [4] S. Shan et al., "Automated red tide algae recognition by the color microscopic image," in *Proc. 2020 13th Int. Congr. Image Signal Process. BioMed. Eng. Inform. (CISP-BMEI)*, 2020, pp. 852–861, doi: 10.1109/CISP-BMEI51763.2020.9263574.
- [5] Y. Zhang et al., "Blue-green algae recognition based on rotation invariant uniform LBP and color information," in *Proc. 2020 IEEE 5th Int. Conf. Signal Image Process. (ICSIP)*, 2020, pp. 46–50, doi: 10.1109/ICSIP49896.2020.9339427.
- [6] Y. Guo, L. Gao, and X. Li, "A deep learning model for green algae detection on SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–14, 2022, doi: 10.1109/TGRS.2022.3215895.
- [7] G. S. Kim et al., "Classification technique of algae using hyperspectral images of algae culture media," *Appl. Sci.*, vol. 13, no. 7, p. 4631, 2023, doi: 10.3390/app13074631.
- [8] P. Otálora et al., "Microalgae classification based on machine learning techniques," *Algal Res.*, vol. 55, p. 102256, 2021, doi: 10.1016/j.algal.2021.102256.
- [9] L. Nicolás-Sáenz et al., "Abanico: A new color space for multi-label pixel classification and color analysis," *Sensors*, vol. 23, no. 6, p. 3338, 2023, doi: 10.3390/s23063338.
- [10] A. Gijsenij, T. Gevers, and J. van de Weijer, "Computational color constancy: Survey and experiments," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2475–2489, 2011, doi: 10.1109/TIP.2011.2118224.

- [11] B. Büyükarıkan and E. Ülker, "Using convolutional neural network models illumination estimation according to light colors," *Optik*, vol. 271, Art. no. 170058, 2022, doi: 10.1016/j.jleleo.2022.170058.
- [12] D. J. Steele, "Cellular viability and the occurrence and significance of chlorophyll allomers during phytoplankton turnover," Ph.D. dissertation, Fac. Sci. Technol., Bournemouth Univ., Poole, U.K., 2014. [Online]. Available: <https://eprints.bournemouth.ac.uk/21777/>
- [13] L. Ma et al., "Prediction of banana maturity based on the sweetness and color values of different segments during ripening," *Curr. Res. Food Sci.*, vol. 5, pp. 1808–1817, 2022, doi: 10.1016/j.crfs.2022.08.024.
- [14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015, doi: 10.1038/nature14539.
- [15] D. M. Madkour, M. I. Shapiai, S. E. Mohamad, H. Aly, Z. H. Ismail, and M. Z. Ibrahim, "A systematic review of deep learning microalgae classification and detection," *IEEE Access*, vol. 11, pp. 57529–57555, 2023, doi: 10.1109/ACCESS.2023.3280410.
- [16] A. Kamilaris and F. X. Prenafeta-Boldú, "A review of the use of convolutional neural networks in agriculture," *J. Agric. Sci.*, vol. 156, no. 3, pp. 312–322, 2018, doi: 10.1017/S0021859618000436.
- [17] M. M. Taye, "Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions," *Computation*, vol. 11, no. 3, p. 52, 2023, doi: 10.3390/computation11030052.
- [18] Z. Xian, R. Huang, D. Towey, and C. Yue, "Convolutional neural network image classification based on different color space," *Tsinghua Sci. Technol.*, vol. 30, no. 1, pp. 402–417, 2025, doi: 10.26599/TST.2024.9010001.
- [19] L. Chen et al., "Review of image classification algorithms based on convolutional neural networks," *Remote Sens.*, vol. 13, no. 22, p. 4712, 2021, doi: 10.3390/rs13224712.
- [20] D. Sahoo et al., Eds., *The Algae World*. Dordrecht, The Netherlands: Springer, 2015, doi: 10.1007/978-94-017-7321-8.
- [21] S. Banayan, M. Jahadi, and K. Khosravi-Darani, "Pigment productions by *Spirulina platensis* as a renewable resource," *J. Appl. Biotechnol. Rep.*, vol. 9, no. 2, pp. 614–621, 2022, doi: 10.30491/JABR.2021.292076.1406.
- [22] K. A. Rihiha et al., "Photobioreactor design utilizing procedural three-dimensional modelling and ray tracing," *J. R. Soc. Interface.*, vol. 22, no. 222, p. 20240451, 2025, doi: 10.1098/rsif.2024.0451.
- [23] K. Price and I. H. Farag, "Resources conservation in microalgae biodiesel production," *Int. J. Eng. Tech. Res.*, vol. 1, no. 8, pp. 49–56, 2013. [Online]. Available: https://www.researchgate.net/publication/259868136_Resources_Conservation_in_Microalgae_Biodiesel_Production
- [24] T. Lafarga et al., "Spirulina for the food and functional food industries," *Food Res. Int.*, vol. 137, p. 109356, 2020, doi: 10.1016/j.foodres.2020.109356.
- [25] N. Yusuf, N. M. Athirah, and A. Suhaila, "Antioxidative responses of *Chlorella vulgaris* under different growth phases," *Squalen Bull. Mar. Fish. Postharvest Biotechnol.*, vol. 17, no. 3, pp. 111–120, 2022, doi: 10.15578/squalen.692.
- [26] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 448–456. [Online]. Available: <https://proceedings.mlr.press/v37/ioffe15.html>
- [27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014. [Online]. Available: <https://jmlr.org/papers/v15/srivastava14a.html>
- [28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.
- [29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [30] J. Cohen, "A coefficient of agreement for nominal scales," *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, 1960, doi: 10.1177/001316446002000104.
- [31] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochim. Biophys. Acta*, vol. 405, no. 2, pp. 442–451, 1975, doi: 10.1016/0005-2795(75)90109-9.
- [32] D. J. Hand and R. J. Till, "A simple generalisation of the area under the ROC curve for multiple class classification problems," *Mach. Learn.*, vol. 45, pp. 171–186, 2001, doi: 10.1023/A:1010920819831.
- [33] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 618–626, doi: 10.1109/ICCV.2017.74.

- [34] P. Linli, T. W. Sen, H. Fahmi, and R. Roestam, "Tomato pest and disease identification based on improved deep residual network and transfer learning," *Indonesian Journal of Artificial Intelligence and Data Mining*, vol. 8, no. 1, pp. 187-200, 2025, doi: 10.24014/ijaidm.v8i1.34038.
- [35] R. Setiawan, Y. Yunita, F. F. Rahman, and H. Fahmi, "BISINDO (Bahasa Isyarat Indonesia) sign language recognition using deep learning," *IT for Society*, vol. 9, no. 1, pp. 17-21, 2024, doi: 10.33021/itfs.v9i1.5076.
- [36] Rosalina, G. Sahuri, H. Fahmi, J. W. Moerland, and R. Mandala, "Personality analysis based on the handwriting shape using integration of image processing and machine learning algorithm," in *Proc. 2023 7th Int. Conf. New Media Stud. (CONMEDIA)*, 2023, pp. 27-32, doi: 10.1109/CONMEDIA60526.2023.10428228.
- [37] M. Rustam, A. Brotokuncoro, W. H. Utomo, and H. Fahmi, "Deblurring photos with Lucy-Richardson and Wiener filter algorithm in RGBA color," *Jurnal Media Infotama*, vol. 21, no. 1, pp. 21-25, 2025, doi: 10.37676/jmi.v21i1.6709.
- [38] F. O. Paranhos, M. L. C. M. dos Reis, J. dos S. Azevedo, and F. de S. Dias, "Convolutional neural networks for evaluating *Spirulina* (*Arthrospira* spp.) adulteration through digital images," *Food Anal. Methods*, vol. 18, no. 8, pp. 1789-1799, 2025, doi: 10.1007/s12161-025-02826-7.
- [39] A. Maali, R. Gheshlaghi, and M. A. Mahdavi, "Maximizing key biochemical products of *Spirulina platensis*: Optimal light quantities and best harvesting time," *OCL*, vol. 31, art. no. 21, 2024, doi: 10.1051/ocl/2024019.