Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

BiSCB-Net: A Lightweight RGB–NIR Semantic Segmentation Framework for Fruit Phenotyping

Wei Peng
West Anhui University
Maosheng Fu
fums@wxc.edu.cn
West Anhui University
Xuemei Zhu
West Anhui University
Liang Huang
Anhui Germplasm Data Technology Co., Ltd
Yiyang Hu
West Anhui University
Jiaxi Yu
West Anhui University

Research Article

Keywords: RGB–NIR imaging, semantic segmentation, multimodal fusion, tomato ripeness, grape region segmentation, visible fruit traits

Posted Date: November 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11687665/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

BiSCB-Net: A Lightweight RGB–NIR Semantic Segmentation Framework for Fruit Phenotyping

Wei Peng[1], Maosheng Fu[1*], Xuemei Zhu[1], Liang Huang[2], Yiyang Hu[1], Jiaxi Yu[1]

1 School of Electronic Information and Artificial Intelligence, West Anhui University, Lu'an 237012, China

2 Anhui Germplasm Data Technology Co., Ltd., Hefei, Anhui Province 230000, China

Corresponding author: Maosheng Fu (fums@wxc.edu.cn)

Abstract

Accurate segmentation of visible fruit regions is essential for image-based fruit phenotyping, yet illumination variation and class imbalance remain challenging. This study proposes BiSCB-Net, a lightweight RGB–NIR semantic segmentation framework integrating dual MobileNetV3-Large encoders, four-scale projected-sum fusion, depthwise-separable residual refinement, and a lightweight feature-pyramid decoder. Using class-balanced cross-entropy and foreground Dice loss, BiSCB-Net achieved foreground mIoU/mDice of 85.714 ± 0.129%/92.304 ± 0.074% on tomato and 91.837 ± 0.093%/95.745 ± 0.050% on grape. With 6.314 million parameters, its batch-1 FP32 inference latency was 9.893 ms at 1024 × 576 and 9.910 ms at 1280 × 720 on an NVIDIA L40S GPU. Evaluation across four tomato acquisition conditions characterized its performance under varying illumination. Image-level ripeness readouts achieved MAEs of 3.272, 2.670, and 1.670 percentage points for Immature, Half-ripe, and Ripe proportions, respectively, with an NVMI MAE of 0.02309. These results support lightweight RGB–NIR segmentation for visible fruit phenotyping.

Keywords: RGB–NIR imaging; semantic segmentation; multimodal fusion; tomato ripeness; grape region segmentation; visible fruit traits

1. Introduction

Quantitative assessment of fruit phenotypic traits supports cultivar evaluation, crop management, harvest planning, and precision agricultural operations. Imaging and sensing technologies are increasingly used in plant phenotyping to obtain quantitative and repeatable trait measurements, helping address limitations of conventional phenotyping, including low throughput, subjectivity, and limited comparability [1–3]. Within this workflow, reliable segmentation is an important step in extracting image-based traits, including visible-area proportions, color composition, and changes in these traits over time.

Image segmentation has become an important component of multimodal plant-image analysis. Henke et al. [4] combined image registration and classification for the segmentation of multimodal images acquired in greenhouse environments. Computer vision and image-

analysis methods have also been applied to a range of plant-phenotyping tasks, including melon germplasm phenotyping [5], time-lapse image analysis of grapevine bunches [6], RGB-D-based litchi phenotyping and weight estimation [7], and segmentation of field-acquired hyperspectral images of raspberry plants [8]. Together, these studies illustrate the range of imaging modalities and analytical approaches used in plant phenotyping.

Images acquired under protected cultivation and in field environments are subject to substantial variations in imaging conditions. Direct sunlight, shadows, artificial illumination, low-light conditions, and changes in color temperature can alter the color and contrast of RGB images. Occlusion among leaves, stems, and fruits can further obscure object boundaries. In tomato-ripeness segmentation, color is an important discriminative cue but is also particularly sensitive to illumination variations. In grape images, dense berry clusters, narrow gaps between berries, and surrounding foliage can make fruit regions difficult to delineate and increase the risk of boundary errors.

RGB and near-infrared (NIR) imaging provide complementary spectral information that may improve fruit-region discrimination under challenging imaging conditions. RGB images provide visible-spectrum color and texture information, whereas NIR images capture reflectance characteristics in the near-infrared range. The two public datasets examined in this study contain spatially aligned RGB images and 16-bit NIR images but represent different segmentation tasks. The tomato dataset includes three visible-ripeness classes (Immature, Half-ripe, and Ripe), whereas the grape dataset includes two color-based region classes (Green grape and Purple grape), rather than ripeness categories. The same network architecture was therefore trained and evaluated independently on the two datasets to examine its applicability to different fruit structures, class definitions, and illumination conditions.

Several influential architectures, including fully convolutional networks (FCN), U-Net, PSPNet, and DeepLabv3+, have advanced pixel-wise prediction, multi-scale context aggregation, and spatial detail recovery in semantic segmentation [9,10,12,13]. Feature pyramid networks (FPN), originally developed for object detection, provide a top-down multi-scale feature pyramid for feature aggregation [11]. ResNet introduced residual learning and has been widely adopted as a feature-extraction backbone, whereas BiSeNet and PIDNet further advanced efficient real-time semantic segmentation [14–16]. These methods provide important foundations for agricultural image analysis. However, when applied to single-modality inputs, they cannot fully exploit the complementary spectral information available from RGB and NIR observations.

RGB-X semantic segmentation methods commonly adopt dual-stream encoders, cross-modal calibration, or attention-based fusion strategies. MFNet and RTFNet employ two-stream architectures for RGB–thermal segmentation [17,18]; CMX introduces cross-modal rectification and Transformer-based fusion for RGB-X inputs [19]; and ECFNet adopts cross-layer interaction for efficient RGB–thermal scene parsing [20]. SegMiF and MRFS investigate the joint use of image fusion and semantic segmentation [21,22]. Attentional Feature Fusion (AFF) is a general-purpose attention-based feature-fusion mechanism for adaptively integrating features [23]. In RGB–thermal semantic segmentation, GMNet explores graded-feature multilabel learning [24], whereas MEFNet adopts a multi-expert fusion strategy [25]. CMNeXt extends cross-modal semantic segmentation to arbitrary modality combinations [26]. These studies demonstrate the value of cross-modal interaction. However, some attention-based and Transformer-based designs may increase model complexity, parameter count, and computational cost, creating challenges for lightweight agricultural applications.

In agricultural multimodal imaging, existing studies have explored RGB–NIR fusion for various segmentation tasks. YOLACTFusion employs dual ResNet50 encoders and attention-based feature fusion for tomato main-stem instance segmentation [27], while AgriFusion combines asymmetric CNN–Transformer encoders with multi-scale RGB–NIR fusion for semantic segmentation of airborne agricultural imagery [28]. More closely related to fruit phenotyping, YOLO-MSRF integrates lightweight RGB–NIR processing for tomato-fruit detection and semantic segmentation under complex illumination, with additional fruit-counting and size-estimation functions [29]. For grape analysis, YOLOv11-WFD incorporates aligned RGB, NIR, and depth inputs for instance-level grape segmentation [30]. Building on these multimodal vision studies, the present work develops a lightweight RGB–NIR semantic segmentation framework for visible fruit phenotyping, using tomato ripeness segmentation as the primary task and grape-region segmentation to assess its applicability across different fruit datasets.

Tomato-ripeness segmentation has also been investigated using various approaches. Shinoda et al. [31] compared CNN- and Transformer-based methods for tomato-ripeness instance segmentation. Rong et al. [32] combined semantic segmentation with morphological processing to identify picking points for ripe tomatoes. Pisharody et al. [33] used parameter-efficient adaptation for tomato-ripeness segmentation and yield prediction, while Chen et al. [34] examined the fusion of RGB, depth, and NIR information for lightweight tomato detection. Collectively, these studies highlight continued interest in tomato-ripeness recognition through segmentation, efficient model adaptation, and multimodal sensing. Lightweight multi-class

semantic segmentation and evaluation across different fruit datasets remain important considerations for RGB–NIR-based visible-fruit analysis.

This study develops and evaluates BiSCB-Net for lightweight RGB–NIR semantic segmentation of visible fruit regions. Tomato ripeness segmentation is the primary task, and grape-region segmentation provides evaluation on a second fruit dataset.

The main contributions of this study are as follows:

1. We develop BiSCB-Net, a lightweight RGB–NIR semantic-segmentation framework that integrates two independent MobileNetV3-Large encoders, four-scale projected-sum fusion, depthwise-separable residual refinement, and a lightweight feature-pyramid decoder.

2. We combine class-balanced cross-entropy (CBCE) with 0.5 foreground Dice loss to address pixel-frequency imbalance and directly optimize overlap across non-background classes.

3. We evaluate BiSCB-Net on two public RGB–NIR fruit datasets, using tomato ripeness segmentation as the primary task and grape segmentation to assess its applicability across different fruit datasets. The evaluation covers segmentation performance, modality contributions, illumination robustness, ablation studies, computational efficiency, and image-level ripeness assessment.

2. Materials and Methods

2.1 BiSCB-Net architecture

2.1.1 Task formulation and network overview

Given an RGB image $I^{(\mathrm{RGB})}$ and an NIR image $I^{(\mathrm{NIR})}$, the network with parameter set W predicts the class probability $p_{(i,c)} = P\left(y_i = c \mid I^{(\mathrm{RGB})}, I^{(\mathrm{NIR})}; W\right)$ at pixel i, and the class with the largest probability is assigned as the predicted label $\hat{y}_i$. Here, $\mathcal{C}$ denotes the class set and $K = |\mathcal{C}|$ its cardinality; $K = 4$ for tomato and $K = 3$ for grape.

The proposed model is referred to as BiSCB-Net throughout this study. It comprises two lightweight modality-specific encoders, four-scale projected-sum fusion, a lightweight feature-pyramid decoder, and a segmentation head. The RGB and NIR branches use topologically identical but parameter-independent MobileNetV3-Large encoders [37] initialized with ImageNet-pretrained weights [38]. The two branches do not share parameters during training, allowing them to adapt to the input statistics of their respective modalities.

Throughout the manuscript, $s \in \mathcal{S} = \{4,8,16,32\}$ denotes the feature output stride, $i \in \mathcal{V}$ denotes a valid pixel in the current minibatch, $c \in \mathcal{C}$ denotes a class, and $\mathcal{C}^+ \subset \mathcal{C}$ denotes the non-background class set. Uppercase symbols denote feature tensors or parameter collections,

whereas lowercase symbols denote indices, scalar probabilities, and weights. Calligraphic symbols denote sets. The probability that pixel i belongs to class c is $p_{(i,c)}$, and y_i is its reference class.

Figure 1 summarizes the data flow from the paired RGB and NIR inputs to the full-resolution segmentation mask. The two modality-specific encoders independently extract features at four output strides. At each scale, the features are projected to a common channel dimension, added, and refined before entering the lightweight FPN decoder. The decoder progressively combines multi-scale features, and the segmentation head produces the final pixel-wise prediction. DS denotes depthwise separable. The displayed NIR example was contrast-stretched only for visualization.

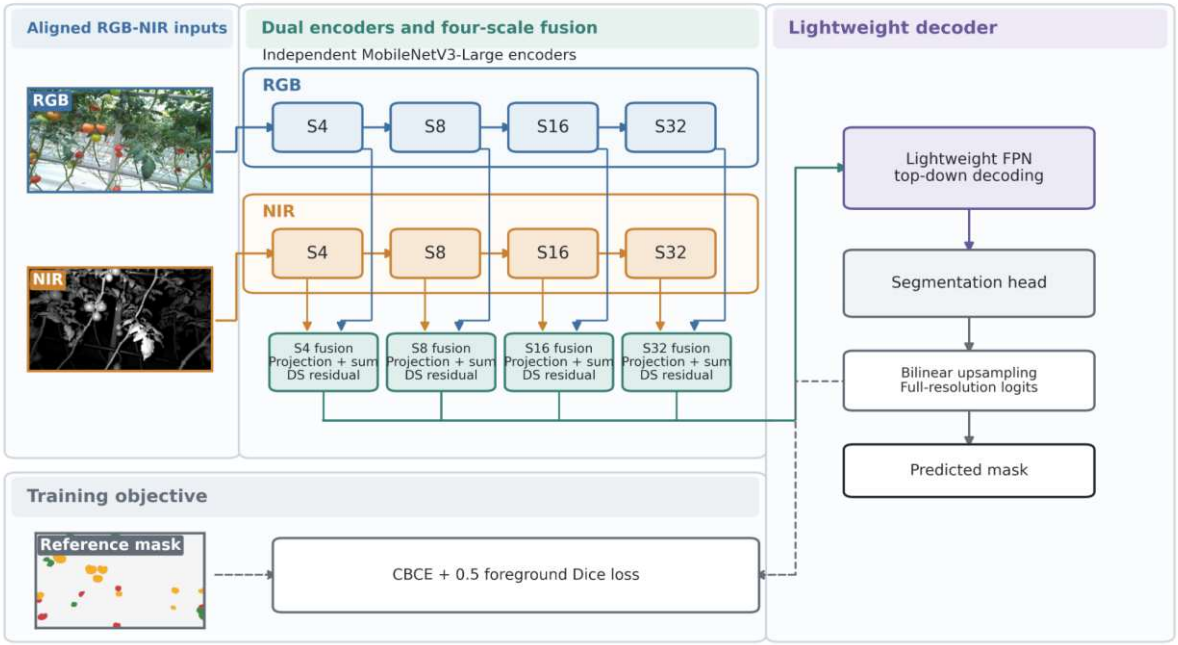

Figure 1. Overall architecture of BiSCB-Net.

The two encoders provide features at output strides S4, S8, S16, and S32, with source channel dimensions of (24, 40, 112, 960). At scale $s \in \mathcal{S}$, the RGB and NIR features are denoted by $F_s^{(\mathrm{RGB})}$ and $F_s^{(\mathrm{NIR})}$. Each modality is projected independently using a 1 × 1 convolution, batch normalization (BN), and SiLU activation, as defined in Eq. (1).

$$R_s = \mathrm{SiLU}\left(\mathrm{BN}\left(\mathrm{Conv}_{(R,s)}^{(1\times1)}\left(F_s^{(\mathrm{RGB})}\right)\right)\right), \quad N_s = \mathrm{SiLU}\left(\mathrm{BN}\left(\mathrm{Conv}_{(N,s)}^{(1\times1)}\left(F_s^{(\mathrm{NIR})}\right)\right)\right) \tag{1}$$

In Eq. (1), R_s and N_s denote the projected RGB and NIR features, respectively. The two projection operators are parameter-independent, and the target channel dimensions at S4, S8, S16, and S32 are (32, 48, 72, 128).

The fused output at scale s is $F_s = \mathrm{DSRes}(R_s + N_s), s \in \mathcal{S}$. Here, $R_s + N_s$ is elementwise addition and $\mathrm{DSRes}(\cdot)$ denotes a depthwise-separable residual refinement unit. This operation forms the low-overhead fusion path.

2.1.2 Projected-sum fusion and depthwise-separable residual unit

Depthwise-separable residual units are used for fusion refinement, FPN refinement, and the segmentation head. For input X, the main branch sequentially applies a depthwise convolution, BN, SiLU, a pointwise convolution, BN, and SiLU, as specified in Eq. (2).

$$B(X) = \mathrm{SiLU}\left(\mathrm{BN}\left(\mathrm{Conv}^{(1\times1)}\left(\mathrm{SiLU}\left(\mathrm{BN}\left(\mathrm{DWConv}^{(3\times3)}(X)\right)\right)\right)\right)\right) \quad (2)$$

In Eq. (2), $B(X)$ is the main-branch transformation, $\mathrm{DWConv}^{(3\times3)}$ performs channel-wise spatial convolution, and $\mathrm{Conv}^{(1\times1)}$ performs channel mixing. The residual output is $Y = B(X) + H(X)$, where $H(\cdot)$ is the shortcut mapping. When the input and output channel dimensions are equal, $H(X) = X$; otherwise, $H(X)$ is a 1×1 convolution-BN-SiLU projection.

The fusion and FPN refinement units preserve the channel dimension, so their shortcuts in Figure 2 are identity mappings. Only the channel-reduction unit in the segmentation head changes the feature dimension from 96 to 64 and therefore uses a projected shortcut. Separating spatial convolution from channel mixing reduces the computational cost of high-resolution feature processing. Figure 2 illustrates the projected-sum operation and the residual refinement pathway.

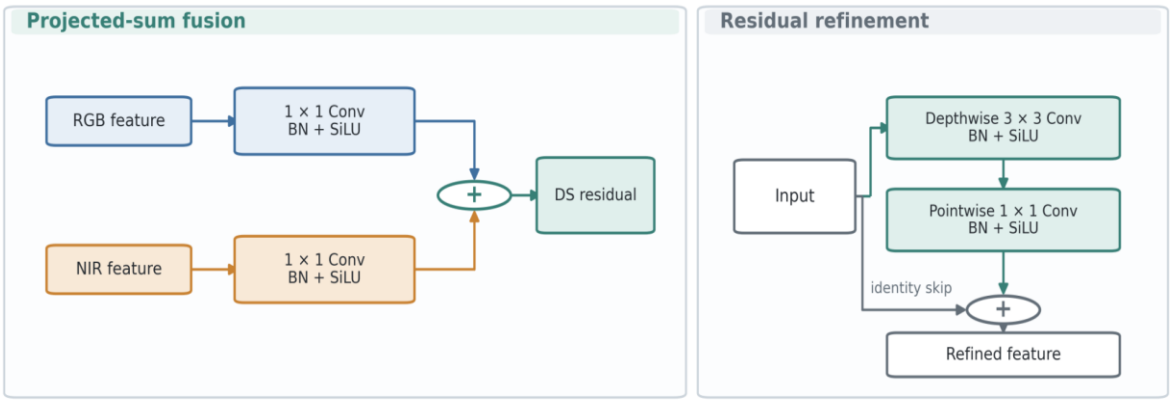

Figure 2. Projected-sum fusion and depthwise-separable residual refinement.

2.1.3 Lightweight feature pyramid decoder

The four fused features are transformed to 96 channels using lateral 1 × 1 convolution-BN-SiLU blocks and are aggregated through the top-down FPN pathway [11], as specified in Eq. (3).

$$L_s = \mathrm{SiLU}\left(\mathrm{BN}\left(\mathrm{Conv}^{(1\times1)}(F_s)\right)\right), s \in \mathcal{S} \tag{3}$$

In Eq. (3), L_s is the 96-channel lateral feature at scale s, F_s is the corresponding fused feature, and the lateral 1 × 1 convolution aligns the channel dimensions across scales.

Top-down decoding begins from the lowest-resolution feature, as defined in Eq. (4).

$$P_s = \mathrm{DSRes}(\mathrm{Up}(P_{2s}) + L_s), \quad s \in \{16,8,4\} \tag{4}$$

Decoding begins with $P_{32} = L_{32}$. In Eq. (4), P_{2s} is the deeper decoder feature at output stride $2s$, and $\mathrm{Up}(\cdot)$ bilinearly interpolates it to the spatial dimensions of L_s. The sum is refined by $\mathrm{DSRes}(\cdot)$, producing P_{16}, P_8, and P_4 in sequence, as illustrated in Figure 3.

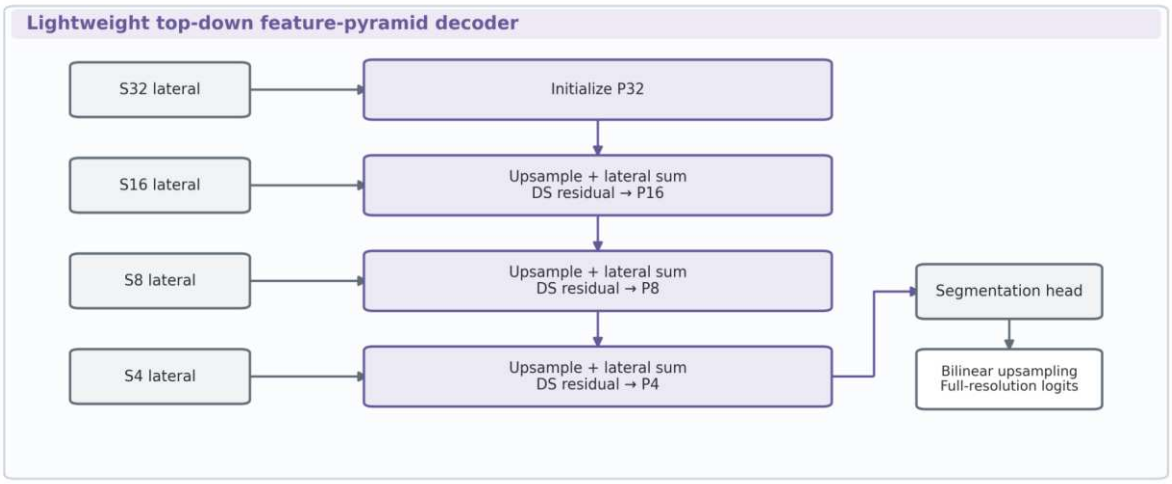

Figure 3. Lightweight feature-pyramid decoder.

2.1.4 Segmentation head

The S4 decoder feature is first transformed from 96 to 64 channels by a depthwise-separable residual unit, followed by two-dimensional dropout with probability 0.1 and a 1 × 1 classification convolution with K output channels, as defined in Eq. (5).

$$Z_4 = \mathrm{Conv}^{(1\times1)}\left(\mathrm{Dropout2D}\left(\mathrm{DSRes}_{(96\to64)}(P_4); p_{\mathrm{drop}} = 0.1\right)\right) \tag{5}$$

In Eq. (5), $\mathrm{DSRes}^{(96\to64)}$ transforms the S4 decoder feature from 96 to 64 channels, $p_{(\mathrm{drop})} = 0.1$ is the Dropout2D probability, and the final 1 × 1 convolution produces K-class S4 logits Z_4. The logits are then upsampled to the input resolution using Eq. (6).

$$p = \mathrm{softmax}\left(\mathrm{Up}_{(\mathrm{bilinear})}(Z_4)\right), \hat{y}_i = \mathrm{argmax}_{(c\in\mathcal{C})} p_{(i,c)} \tag{6}$$

In Eq. (6), $\mathrm{Up}_{(\mathrm{bilinear})}(\cdot)$ denotes bilinear upsampling to the input resolution.

2.2 Training objective

2.2.1 Class-balanced cross-entropy

Let n_c denote the pixel count of class c in the training set after add-one smoothing. The class frequency f_c is n_c divided by the sum of the smoothed counts over all classes. The class weights are calculated using Eq. (7) and normalized to a mean of 1.

$$\widetilde{w}_c = 1/\log(1.02 + f_c)\,, w_c = (K\widetilde{w}_c)/\left(\sum_{(k\in\mathcal{C})}\widetilde{w}_k\right) \tag{7}$$

In Eq. (7), $\widetilde{w}_c$ is the unnormalized weight for class c, the constant 1.02 limits the magnitude assigned to extremely infrequent classes, $K = |\mathcal{C}|$ is the number of classes, and w_c is the final mean-normalized weight. The resulting weight vectors were [0.051673, 1.104427, 1.344437, 1.499462] for tomato and [0.228274, 1.514134, 1.257591] for grape.

Under the weighted-mean reduction, CBCE weights each pixel-wise cross-entropy term by the reference class weight and normalizes the result by the sum of reference class weights in the minibatch, as defined in Eq. (8).

$$\mathcal{L}_{(\mathrm{CBCE})} = \left(\sum_{(i\in\mathcal{V})} w_{(y_i)}\left(-\log p_{(i,y_i)}\right)\right)/\left(\sum_{(i\in\mathcal{V})} w_{(y_i)}\right) \tag{8}$$

In Eq. (8), $\mathcal{V}$ is the set of valid pixels in the current minibatch, y_i is the reference class of pixel i, $p_{(i,y_i)}$ is the predicted probability for that class, and $w_{(y_i)}$ is its class weight. The denominator is the sum of the reference class weights over the current minibatch.

2.2.2 Foreground Dice loss

Soft Dice is calculated for each non-background class and then macro-averaged across the foreground classes, as defined in Eq. (9).

$$L_{\mathrm{Dice}} = \frac{1}{K-1}\sum_{c\in\mathcal{C}^+}\left(1 - \frac{2\sum_{i\in\mathcal{V}} p_{i,c}t_{i,c} + \varepsilon_s}{\sum_{i\in\mathcal{V}}(p_{i,c} + t_{i,c}) + \varepsilon_s}\right) \tag{9}$$

In Eq. (9), $\mathcal{C}^+$ is the non-background class set; because Background is the only excluded class, its cardinality is $K-1$. $p_{i,c}$ and $t_{i,c}$ denote the predicted probability and one-hot reference label for class c at pixel i, respectively, and the sums are taken over $i \in \mathcal{V}$. The smoothing constant ε_s is 10^{-6}. The outer summation macro-averages the class-wise Dice terms over the foreground classes.

The final training objective is $L = L_{\mathrm{CBCE}} + 0.5L_{\mathrm{Dice}}$. The coefficient 0.5 controls the relative contribution of the overlap term. CBCE compensates for pixel-frequency imbalance using fixed training-set class weights, whereas foreground Dice directly optimizes overlap for the non-background classes [39,40]. Figure 4 summarizes how the two loss terms are calculated and combined during training.

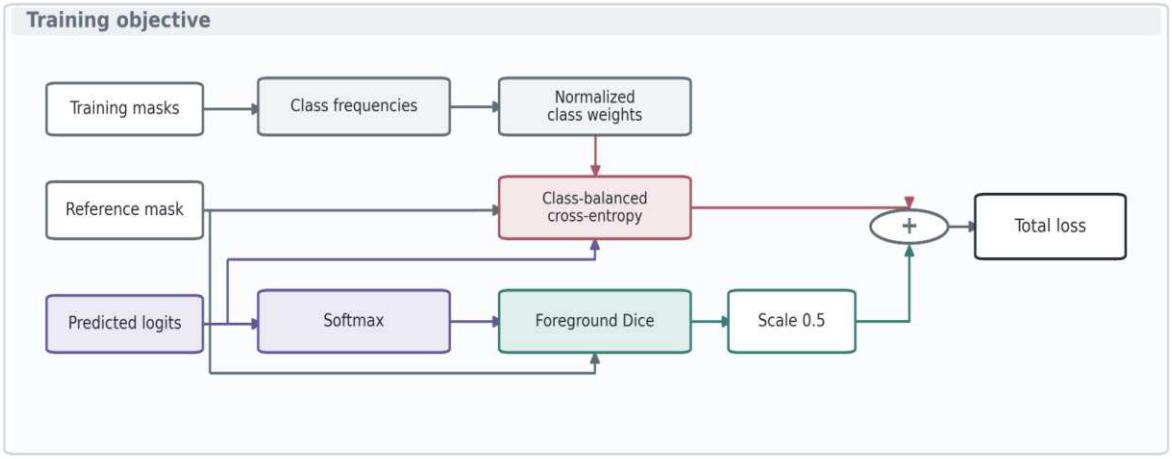

Figure 4. Training objective combining class-balanced cross-entropy and foreground Dice loss.

2.3 Image-level visible-ripeness readout calculation

For each tomato test image, the foreground class set is $\mathcal{C}^+$, comprising Immature, Half-ripe, and Ripe. Superscripts pred and ref distinguish quantities calculated from predicted and reference masks. For image j and class $c \in \mathcal{C}^+$, $A^{(\mathrm{pred})}(j,c)$ and $A^{(\mathrm{ref})}(j,c)$ are the foreground pixel counts from the two masks. The corresponding visible-area proportions, $r^{(\mathrm{pred})}(j,c)$ and $r^{(\mathrm{ref})}(j,c)$, use the total visible tomato foreground area in their respective masks as the denominator.

The normalized visible-maturity index (NVMI) was calculated separately from each predicted and reference mask as $\mathrm{NVMI}^{(\mathrm{pred})}(j) = \sum_{(c\in\mathcal{C}^+)} \alpha_c r^{(\mathrm{pred})}(j,c)$; $\mathrm{NVMI}^{(\mathrm{ref})}(j)$ used the same expression with $r^{(\mathrm{ref})}$. The class weights α_c were 0, 0.5, and 1 for Immature, Half-ripe, and Ripe, respectively. NVMI is an operational image descriptor defined from visible class composition rather than a physiologically calibrated maturity scale.

3. Experiments

3.1 Datasets and semantic targets

3.1.1 Tomato RGB–NIR dataset

The public tomato RGB–NIR dataset [35] comprises paired visible-spectrum RGB and 16-bit near-infrared (NIR) images of tomato fruits at different maturity stages. It covers four acquisition conditions—artificial light, low light, natural light, and sodium-vapor illumination—providing samples for evaluating segmentation under varied illumination.

The semantic annotations define four classes: Background, Immature, Half-ripe, and Ripe. The three foreground classes label visible tomato surfaces at different ripeness stages, whereas branches, leaves, peduncles, sepals, and other non-fruit regions are assigned to

Background. Occluded fruit surfaces are not inferred, and individual fruit instances are not identified. The task is therefore four-class semantic segmentation of visible ripeness regions.

The original images have a resolution of 1280 × 720 pixels. Each RGB image is spatially paired with its corresponding NIR image and pixel-level semantic mask.

3.1.2 Grape RGB–NIR dataset

The grape experiments used the public multimodal grape object-detection and semantic-segmentation dataset [36], retaining only spatially aligned RGB and NIR images. The source resolution was 1280 × 720 and was preserved throughout the experiments. Semantic masks were rasterized directly from the source JSON polygon annotations. The labels were Background, Green grape, and Purple grape, and each image contained only one foreground color class. The annotations represented combined visible berry or cluster regions without distinguishing individual berries or grape clusters.

3.1.3 Split definition and reproducibility

The tomato dataset was divided into 2,800 training, 800 validation, and 400 test images. The grape dataset was stratified by source illumination category and grape color using a 70%/20%/10% split, resulting in 2,768 training, 791 validation, and 395 test images. A split manifest recorded each image's source file, identifier, illumination category, and color class.

For the grape dataset, Normal-light source files retained the Normal label. Poor-light images were further divided into Dark and Supplemental groups according to their mean BT.709 luminance, calculated from ordered RGB thumbnails using a single change-point threshold. These three groups were used for illumination-stratified analysis. Both datasets were split at the image level, and the original annotations were retained without manual modification.

3.2 Image preprocessing and augmentation

Before training, tomato images were resized from 1280 × 720 to 1024 × 576, whereas grape images retained their original resolution of 1280 × 720. The paired RGB images, NIR images, and semantic masks were kept spatially aligned.

RGB images were converted to float32 and scaled to [0, 1]. For each 16-bit NIR image, zero-valued pixels were treated as invalid. The 1st and 99th percentiles of valid pixel intensities, denoted by q_1 and q_{99}, respectively, were calculated independently for each image and used for robust per-image normalization, as defined in Eq. (10).

The normalized NIR intensity at pixel i was calculated as:

$$\tilde{I}^{(\mathrm{NIR})}(i) = \begin{cases} \mathrm{clip}\left(\frac{I^{(\mathrm{NIR})}(i) - q_1}{q_{99} - q_1}, 0, 1\right), & I^{(\mathrm{NIR})}(i) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

Here, $I^{(\mathrm{NIR})}$ and $\tilde{I}^{(\mathrm{NIR})}$ denote the original and normalized NIR images, respectively, and i represents a pixel position. The values q_1 and q_{99} represent the 1st and 99th percentiles of valid pixel intensities in the current image. The clipping operation restricts normalized values to [0, 1], while invalid pixels remain zero. If q_{99} is not greater than q_1, the minimum and maximum valid pixel values are used instead; if these values are also equal, all valid pixels are assigned a value of 1.

Before encoder input, RGB images were standardized using the ImageNet channel means (0.485, 0.456, 0.406) and standard deviations (0.229, 0.224, 0.225). The normalized single-channel NIR input was standardized using a mean of 0.449 and a standard deviation of 0.226, obtained by averaging the three ImageNet channel means and standard deviations, respectively. The standardized NIR channel was replicated three times for the NIR encoder. No separate invalid-pixel mask was supplied to the network.

During training, synchronized horizontal flipping was applied to the paired RGB images, NIR images, and corresponding segmentation masks with a probability of 0.5. Other augmentations, including vertical flipping, random cropping, rotation, scaling, color jittering, Mosaic, MixUp, and Copy-Paste, were disabled. No random augmentation was applied during validation or testing. The data loader arranged the paired inputs in NIR–RGB channel order and supplied each modality to its corresponding encoder.

3.3 Experimental settings and evaluation protocol

3.3.1 Training and comparative implementation

All complete models were trained using CBCE + 0.5 foreground Dice. Tomato and grape models were trained for 200 epochs with batch size 8 and eight parallel data-loading workers. Training batches were shuffled, and incomplete final batches were omitted. Validation and test samples were evaluated in a fixed order. Pinned host memory was enabled for data transfer. AdamW was used with an initial learning rate of $1 \times 10^{(-3)}$ and weight decay of $5 \times 10^{(-4)}$. A cosine-annealing schedule was applied over $T_{(max)} = 200$ epochs and a minimum learning rate of $5 \times 10^{(-5)}$. The learning rate was updated after each epoch. Warm-up and early stopping were not used. Training used automatic mixed precision.

The complete tomato and grape models were trained independently with random seeds 0, 1, and 2. The random-number generators used by Python, NumPy, and the CPU and GPU training libraries were initialized with the specified seed, and deterministic computation was

enabled for each run. For each run, one checkpoint was selected by validation and evaluated once on the independent test set. Main complete-model results are reported as the mean and sample standard deviation of the three test results. The RGB-only and NIR-only controls were also trained independently with seeds 0, 1, and 2, and their results are reported as the mean and sample standard deviation across the three runs.

Four established RGB-X segmentation models were reimplemented based on their official repositories and adapted for aligned RGB–NIR inputs and dataset-specific output classes. In each model, NIR replaced the original thermal or auxiliary X modality. All models were trained and evaluated using the same prepared image pairs, fixed dataset splits, input resolutions, synchronized horizontal-flip augmentation, 200-epoch training schedule, total batch size of 8, CBCE + 0.5 foreground Dice loss, validation-based checkpoint selection, and test evaluation protocol. The common data loader converted RGB images to float32 and scaled them to [0, 1], while applying the robust per-image NIR normalization described in Section 3.2. MFNet and RTFNet-50 received the common preprocessed tensors directly, whereas ECFNet and CMX applied their respective model-specific internal standardization procedures, as detailed below.

MFNet reordered the four-channel tensor to RGB–NIR, used no pretrained weights, and was optimized with SGD (initial learning rate 0.01, momentum 0.9, weight decay 5×10^{-4}) with a per-epoch decay factor of 0.95. RTFNet-50 initialized both ResNet-50 encoders from ImageNet weights and used AdamW (learning rate 1×10^{-3}, weight decay 5×10^{-4}) with cosine annealing. ECFNet retained its ImageNet-pretrained ResNet-18 encoder and four-output deep supervision, applied ImageNet normalization to RGB and 0.449/0.226 standardization to NIR, and used AdamW (learning rate 1×10^{-3}, weight decay 1×10^{-3}, $\beta = (0.9, 0.99)$) with the official triangular2 cyclic schedule. CMX-MiT-B2 repeated NIR to three channels, applied ImageNet normalization to both streams, initialized both modality encoders from the same MiT-B2 pretrained checkpoint while leaving cross-modal fusion layers randomly initialized, and used AdamW (learning rate 6×10^{-5}, weight decay 5×10^{-4}) with cosine annealing.

3.3.2 Evaluation, ablation, and efficiency protocols

The primary metric was the macro-averaged intersection over union across non-background classes ($\mathrm{mIoU}_{(\mathrm{fg})}$). We also reported foreground macro-averaged Dice ($\mathrm{mDice}_{(\mathrm{fg})}$), class-wise IoU, mIoU including Background, and pixel accuracy (PA). For condition-stratified analysis, the same test set was grouped by the tomato acquisition-

condition label or the predefined grape luminance subset, and true positives (TP), false positives (FP), and false negatives (FN) were accumulated within each group before IoU was calculated. Full-test-set metrics were obtained from the confusion statistics accumulated over all test pixels.

$$\mathrm{IoU}_c = \mathrm{TP}_c/(\mathrm{TP}_c + \mathrm{FP}_c + \mathrm{FN}_c), \mathrm{Dice}_c = 2\mathrm{TP}_c/(2\mathrm{TP}_c + \mathrm{FP}_c + \mathrm{FN}_c) \quad (11)$$

In Eq. (11), TP_c, FP_c, and FN_c are the accumulated true-positive, false-positive, and false-negative pixel counts for class c in the relevant evaluation set.

The primary metrics were macro-averaged only over the non-background class set $\mathcal{C}^+$: $\mathrm{mIoU}_{(\mathrm{fg})}$ is the arithmetic mean of the foreground class IoUs, and $\mathrm{mDice}_{(\mathrm{fg})}$ is the arithmetic mean of the foreground class Dice scores.

Pixel accuracy, PA, is the number of true-positive pixels summed over all classes divided by the number of valid pixels.

Efficiency metrics included parameter count, GFLOPs, batch-1 inference latency, throughput, and peak GPU memory. All measurements used *FP*32 inference. The hardware, software environment, input resolutions, warm-up iterations, and timed iterations are specified below.

The tomato loss ablation compared CBCE alone with CBCE + 0.5 foreground Dice under the same full projected-sum architecture. Structural ablations examined the fusion operator, NIR fusion scales, and depthwise-separable (DS) residual refinement. All ablation experiments were conducted using seed 0. The loss comparison evaluated the contribution of the foreground Dice term, while the structural ablations assessed the effects of the selected fusion and refinement configurations on segmentation performance.

The fixed split manifests and checksums, training configurations, random-seed records, checkpoint-selection rules, evaluation procedures, and analysis materials are available from the corresponding author on reasonable request.

Computational efficiency was evaluated on an NVIDIA L40S GPU using FP32 precision, a batch size of 1, PyTorch 2.3.0a0+40ec155e58.nv24.03, and CUDA 12.4. All models were tested at the corresponding dataset input resolution (1024 × 576 for tomato and 1280 × 720 for grape). For BiSCB-Net, the seed-0 complete-model checkpoint was profiled separately for each dataset. After 30 warm-up iterations, inference latency was measured over 100 forward passes with GPU synchronization. The reported latency is the mean of these 100 measurements; throughput was calculated as 1000 divided by the mean latency, and peak allocated CUDA memory was recorded. Parameter count and FLOPs were measured using the

same profiling procedure for all models. The reported latency excludes image loading, data transfer, preprocessing, and mask postprocessing.

3.3.3 Visible-ripeness readout analysis

For the 400 tomato test images, predictions from three independently trained checkpoints (seeds 0–2) were evaluated against the same reference masks. Figure 6 shows the seed-0 results, with each point representing one test image.

For each seed, image-level errors were calculated as the difference between predicted and reference readouts. MAE and mean bias were computed separately across the 400 images, and Table 5 reports the mean ± sample standard deviation across the three seeds. Class-proportion errors are expressed in percentage points (pp), while NVMI errors are reported on the [0,1] scale. The 95% confidence intervals were estimated using 10,000 paired image-level bootstrap resamples, with the same resampled image indices applied across all three seeds to preserve pairing. The 2.5th and 97.5th percentiles were used as the confidence limits. Thus, the standard deviations reflect between-seed variability, whereas the confidence intervals represent image-sampling uncertainty.

4. Results

4.1 Comparative performance and modality contribution

Across three independent runs, BiSCB-Net achieved foreground mIoU and foreground mDice values of 85.714 ± 0.129% and 92.304 ± 0.074% on tomato, and 91.837 ± 0.093% and 95.745 ± 0.050% on grape, respectively. The RGB-only and NIR-only controls were also evaluated over seeds 0–2. On tomato, RGB-only achieved 85.514 ± 0.190% mIoU_fg and 92.187 ± 0.110% mDice_fg, whereas NIR-only achieved 81.446 ± 0.086% and 89.772 ± 0.053%, respectively. On grape, RGB-only achieved 86.409 ± 0.077% mIoU_fg and 92.708 ± 0.044% mDice_fg, whereas NIR-only achieved 82.701 ± 0.204% and 90.509 ± 0.127%, respectively. Relative to RGB-only, RGB–NIR BiSCB-Net increased mIoU_fg/mDice_fg by 0.200/0.117 percentage points (pp) on tomato and 5.428/3.037 pp on grape. Relative to NIR-only, the corresponding gains were 4.268/2.532 pp on tomato and 9.136/5.236 pp on grape. The smaller gain over RGB-only on tomato and the larger gains on grape indicate that the contribution of multimodal fusion differs between the two fruit datasets.

Table 1. Comparison of input modalities and representative segmentation methods.

Method / input	Tomato mIoU_fg (%)	Tomato mDice_fg (%)	Grape mIoU_fg (%)	Grape mDice_fg (%)
BiSCB-Net (RGB-only)	85.514 ± 0.190	92.187 ± 0.110	86.409 ± 0.077	92.708 ± 0.044

Method / input	Tomato mIoU_fg (%)	Tomato mDice_fg (%)	Grape mIoU_fg (%)	Grape mDice_fg (%)
BiSCB-Net (NIR-only)	81.446 ± 0.086	89.772 ± 0.053	82.701 ± 0.204	90.509 ± 0.127
BiSCB-Net	85.714 ± 0.129	92.304 ± 0.074	91.837 ± 0.093	95.745 ± 0.050
MFNet	83.381	90.932	87.843	93.524
RTFNet-50	84.433	91.557	90.792	95.173
ECFNet	85.294	92.059	90.862	95.212
CMX-MiT-B2	86.853	92.962	92.555	96.133

Under the same dataset splits, 200-epoch training budget, checkpoint-selection rule, and evaluation protocol, BiSCB-Net achieved higher foreground mIoU and mDice than ECFNet on both tomato and grape datasets. Compared with CMX-MiT-B2, BiSCB-Net delivered competitive segmentation performance while offering a substantially more compact architecture and lower computational cost, as further demonstrated in Table 4. The three BiSCB-Net input configurations were evaluated over three independent runs (seeds 0–2), with results reported as mean ± sample standard deviation. The other comparison models were evaluated using single local runs with their respective optimizer and learning-rate settings.

4.2 Ablation, condition-stratified, and efficiency analyses

Table 2 summarizes the loss and structural ablation results on tomato using seed 0. Under the same full architecture, adding 0.5 foreground Dice to CBCE improved mIoU_fg and mDice_fg from 83.894% and 91.241% to 85.853% and 92.384%, respectively, corresponding to gains of 1.959 and 1.143 percentage points (pp). Among the evaluated fusion configurations, the complete dual-projection Sum achieved the highest scores, exceeding Direct-Sum and half-channel projection + Concat by 0.078/0.047 pp and 0.178/0.102 pp, respectively. Four-scale NIR fusion also achieved the best results among the evaluated scale configurations, outperforming S32-only and S16+S32 fusion by 0.129/0.077 pp and 0.134/0.075 pp, respectively. DS residual refinement further increased mIoU_fg and mDice_fg by 0.053 and 0.030 pp, respectively.

Table 2. Seed-0 tomato ablations of the loss objective and network components.

Component	Configuration	mIoU_fg (%)	mDice_fg (%)
Loss objective	CBCE only	83.894	91.241
Fusion operator	Direct-Sum (shared projection)	85.775	92.337
	Half-channel projection + Concat	85.675	92.282
NIR fusion scales	S32 only	85.724	92.307

Component	Configuration	mIoU_fg (%)	mDice_fg (%)
	S16 + S32	85.719	92.309
DS residual	Disabled	85.800	92.354
Ours	**Dual projection + four-scale Sum + DS residual; CBCE + 0.5 foreground Dice**	**85.853**	**92.384**

Table 3 summarizes the three-run segmentation performance across different acquisition conditions on tomato and luminance groups on grape. On tomato, BiSCB-Net achieved its highest foreground mIoU and mDice under natural light, reaching 87.224% and 93.172%, respectively. Under low-light conditions, the model maintained foreground mIoU and mDice of 82.831% and 90.537%, respectively. On grape, the Supplemental group achieved the highest foreground mIoU and mDice (93.228 ± 0.158% and 96.485 ± 0.086%, respectively), followed by Dark (92.977 ± 0.135% and 96.357 ± 0.073%) and Normal (90.106 ± 0.317% and 94.794 ± 0.174%). These results characterize BiSCB-Net's segmentation performance across different acquisition conditions and image-luminance groups.

Table 3. Condition-stratified segmentation performance on the tomato and grape test sets.

Dataset	Condition/subset	mIoU_fg (%)	mDice_fg (%)	Images (n)
Tomato	Artificial light	84.846 ± 0.460	91.790 ± 0.275	100
Tomato	Low light	82.831 ± 1.438	90.537 ± 0.915	100
Tomato	Natural light	87.224 ± 0.075	93.172 ± 0.044	100
Tomato	Sodium-vapor illumination	84.917 ± 0.391	91.790 ± 0.221	100
Grape	Dark	92.977 ± 0.135	96.357 ± 0.073	100
Grape	Normal	90.106 ± 0.317	94.794 ± 0.174	195
Grape	Supplemental	93.228 ± 0.158	96.485 ± 0.086	100

BiSCB-Net contains 6.314 M parameters on both datasets. At the tomato input resolution (1024 × 576), it required 13.074 GFLOPs and achieved a mean forward-pass latency of 9.893 ms, corresponding to 101.08 images/s, with a peak allocated GPU memory of 144.75 MiB. At the grape input resolution (1280 × 720), it required 20.503 GFLOPs and achieved a mean latency of 9.910 ms (100.90 images/s), with a peak memory of 213.77 MiB. The nearly identical parameter counts reflect the shared network design, whereas the higher grape input resolution increases computational cost and memory use. The small latency difference between the two resolutions was observed on the NVIDIA L40S under the stated batch-1

FP32 protocol. The minor dataset-dependent parameter-count differences for ECFNet and CMX-MiT-B2 arise from the four-class tomato output head and the three-class grape output head. Table 4 reports input sizes in pixels, parameter counts in millions, latency in milliseconds, and peak memory in MiB. BiSCB-Net efficiency values are based on the seed-0 checkpoint; latency is averaged over 100 forward passes rather than over training seeds.

Table 4. Model complexity and inference efficiency.

Data	Model	Size	Params	GFLOPs	ms	FPS	MiB
Tomato	MFNet	1024×576	0.735	30.172	4.472	223.59	190.44
Tomato	RTFNet-50	1024×576	185.234	702.725	20.989	47.64	1073.76
Tomato	ECFNet	1024×576	21.270	153.148	11.051	90.49	400.60
Tomato	CMX-MiT-B2	1024×576	66.563	337.095	41.946	23.84	1056.69
Tomato	BiSCB-Net	1024×576	6.314	13.074	9.893	101.08	144.75
Grape	MFNet	1280×720	0.735	46.613	5.860	170.66	290.32
Grape	RTFNet-50	1280×720	185.234	1116.861	31.703	31.54	1287.30
Grape	ECFNet	1280×720	21.268	244.404	12.279	81.44	575.00
Grape	CMX-MiT-B2	1280×720	66.562	593.204	52.108	19.19	1484.86
Grape	BiSCB-Net	1280×720	6.314	20.503	9.910	100.90	213.77

4.3 Qualitative segmentation examples

Figure 5 presents representative tomato test images from the four acquisition conditions. Each row shows the RGB and NIR inputs, the ground-truth mask, and the corresponding prediction generated by the seed-0 BiSCB-Net model. The predicted masks reproduce the spatial distribution of visible tomato ripeness regions across the four acquisition conditions, showing overall agreement with the ground-truth annotations. These qualitative results complement the condition-stratified quantitative evaluation in Table 3.

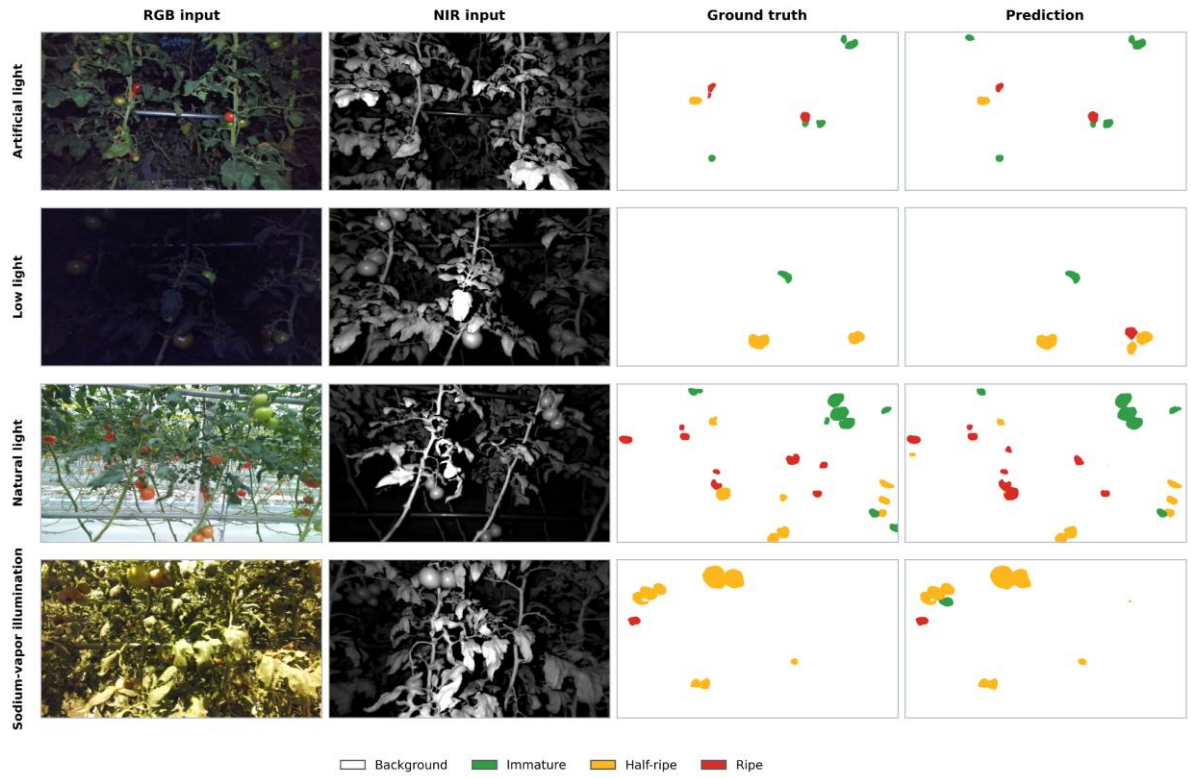

Figure 5. Representative tomato test images under four acquisition conditions. Columns show the RGB input, NIR input, ground-truth mask, and seed-0 BiSCB-Net prediction. Mask colors indicate background (white), immature (green), half-ripe (yellow), and ripe (red).

4.4 Image-level visible-ripeness readout agreement on the tomato test set

Table 5 summarizes the agreement between predicted- and reference-mask-derived image-level readouts across three independent runs (seeds 0–2). For each seed, MAE and mean bias were calculated across the same 400 test images, and the reported mean ± sample standard deviation represents the variation across the three runs. The 95% confidence intervals were estimated using 10,000 paired image-level bootstrap resamples, with the same resampled image indices applied across all three seeds to preserve image-level pairing.

The MAE values were 3.272, 2.670, and 1.670 pp for Immature, Half-ripe, and Ripe proportions, respectively, while the NVMI MAE was 0.02309. Mean biases were close to zero across all four readouts. The 95% confidence intervals included zero for Immature, Half-ripe, and NVMI, whereas Ripe showed a slight negative bias (95% CI, −0.678 to −0.001 pp). Figure 6 illustrates the agreement for seed 0, with each point representing one of the 400 test images and the dashed diagonal indicating perfect agreement between predicted- and reference-mask-derived readouts.

Across the three runs, illumination-stratified NVMI MAEs were 0.02251 ± 0.00149 under Artificial light, 0.02696 ± 0.00134 under Low light, 0.01648 ± 0.00033 under Natural light, and 0.02639 ± 0.00063 under Sodium-vapor illumination. Each condition contained 100

test images. Natural light showed the lowest NVMI readout error, while Low light and Sodium-vapor illumination showed higher errors.

Table 5. Agreement of tomato image-level visible-ripeness readouts between predicted and reference masks.

Readout	MAE (mean ± SD)	MAE 95% CI	Bias (mean ± SD)	Bias 95% CI
Immature (pp)	3.272 ± 0.071	[2.768, 3.830]	+0.373 ± 0.199	[−0.239, 0.956]
Half-ripe (pp)	2.670 ± 0.084	[2.217, 3.185]	−0.042 ± 0.092	[−0.565, 0.523]
Ripe (pp)	1.670 ± 0.029	[1.379, 1.981]	−0.331 ± 0.109	[−0.678, −0.001]
NVMI ([0,1])	0.02309 ± 0.00047	[0.01981, 0.02668]	−0.00352 ± 0.00154	[−0.00761, 0.00057]

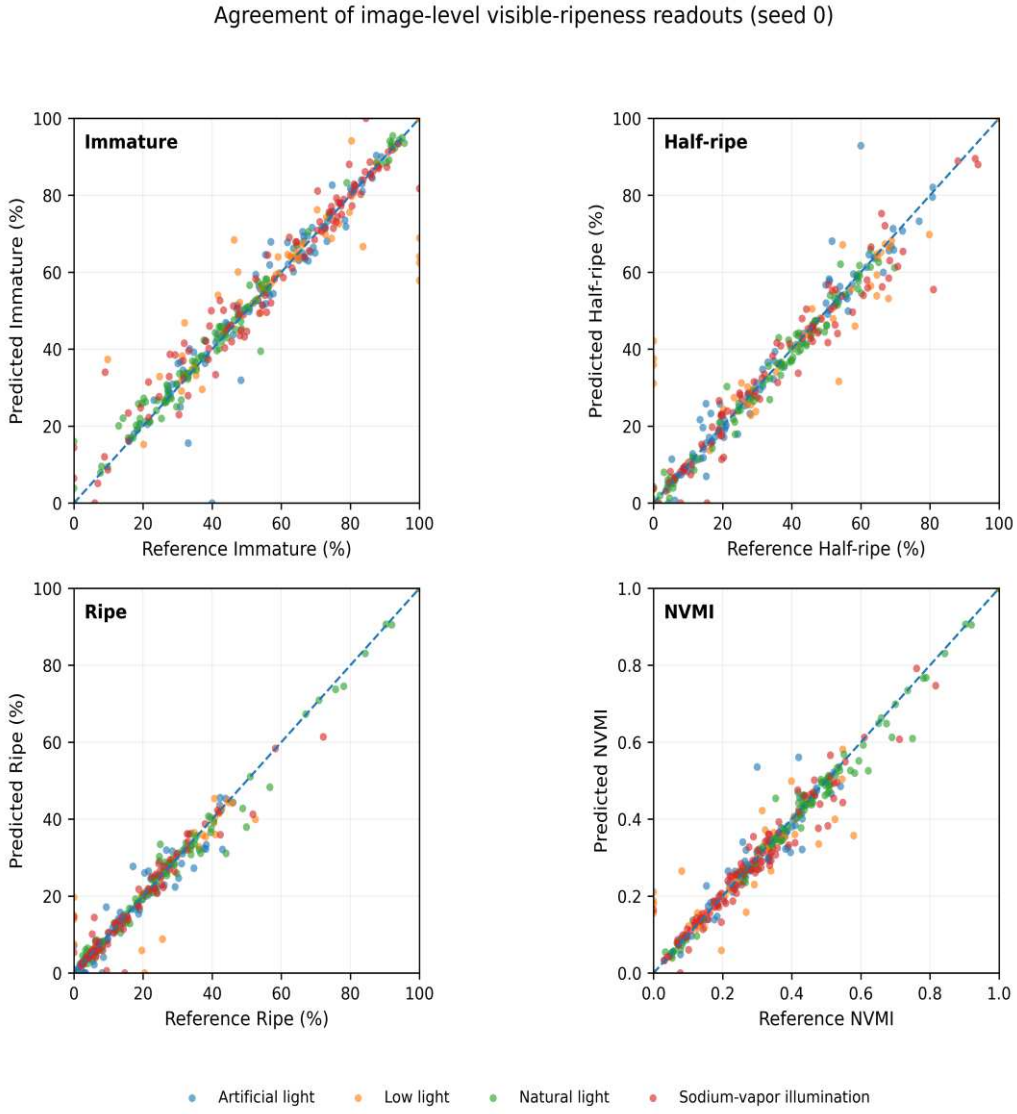

Figure 6. Seed-0 agreement plots for predicted- and reference-mask-derived visible-ripeness readouts on the 400 tomato test images.

5. Discussion and conclusions

The results demonstrate the applicability of lightweight RGB–NIR semantic segmentation to fruit phenotyping across tomato and grape datasets. The contribution of NIR varied across the two tasks: RGB–NIR fusion maintained performance comparable to RGB-only input on tomato and provided substantial improvements on grape. This cross-dataset evaluation highlights the complementary role of NIR information in fruit-region segmentation under different imaging conditions. BiSCB-Net further combines competitive segmentation performance with a compact architecture and favorable computational efficiency, providing a practical alternative for multimodal fruit phenotyping.

The ablation experiments support the selected loss formulation and network configuration. Adding foreground Dice to CBCE improved both foreground mIoU and mDice, demonstrating the value of combining class-balanced classification supervision with foreground overlap optimization. Among the evaluated structural configurations, dual-projection Sum, four-scale NIR fusion, and DS residual refinement produced the best overall results. Together, these components form a lightweight architecture that integrates multi-scale cross-modal information while maintaining computational efficiency.

Illumination-stratified evaluation further characterized the model's performance across different acquisition conditions. On tomato, natural light produced the highest segmentation scores, while low-light images remained segmentable with foreground mIoU and mDice of 82.831% and 90.537%, respectively. For grape, the three image-luminance groups showed distinct performance profiles, with the Supplemental group achieving the highest foreground mIoU and mDice. Beyond pixel-level segmentation, the predicted tomato masks supported image-level visible-ripeness readouts, with MAEs of 3.272, 2.670, and 1.670 pp for Immature, Half-ripe, and Ripe proportions, respectively, and an NVMI MAE of 0.02309. These results extend the evaluation from segmentation accuracy to the consistency of image-derived phenotypic readouts.

This study establishes a reproducible RGB–NIR segmentation workflow for visible fruit phenotyping. Future work will extend evaluation to scene-independent data splits, repeated structural ablations, and deployment-oriented profiling on edge devices. Integrating independent fruit-quality measurements will also enable the relationship between visible-ripeness readouts and physiological maturity to be investigated. Extending the grape annotations to distinguish individual berries and mixed-color clusters will further broaden the range of fruit phenotyping tasks.

Declarations

Availability of data and materials

The public tomato and grape RGB–NIR datasets analysed in this study are available from their original repositories at https://doi.org/10.11922/11-6035.csd.2023.0154.zh and https://doi.org/10.11922/11-6035.csd.2023.0132.zh, respectively. The original datasets are not redistributed by the authors. The fixed split manifests, training configurations, evaluation procedures, and analysis materials are available from the corresponding author on reasonable request.

Competing interests

The authors declare that they have no competing interests.

Funding

This work was supported by the University-Level Project of West Anhui University, "Construction of a High-Precision Perception and Intelligent Analysis Platform for the Moisture Status of *Dendrobium officinale*" (Grant No. 0041026051); the Horizontal Research Projects (Grant Nos. 0045025103 and 0045025106); the Open Research Project of Anhui Dabieshan Academy of Traditional Chinese Medicine (Grant No. TCMADM-2024-07); and the High-Level Talent Research Start-Up Projects of West Anhui University (Grant Nos. WGKQ2024011 and WGKQ2023003).

Authors' contributions

WP conceived the study, developed the methodology and software, and drafted the manuscript. MF supervised the study and administered the project. XZ contributed to software development and data curation. LH performed validation and reviewed and edited the manuscript. YH and JY performed data preprocessing and curation. All authors read and approved the final manuscript.

References

1. Furbank RT, Tester M. Phenomics—technologies to relieve the phenotyping bottleneck. Trends Plant Sci. 2011;16:635–644. https://doi.org/10.1016/j.tplants.2011.09.005
2. Tardieu F, Cabrera-Bosquet L, Pridmore T, Bennett M. Plant phenomics, from sensors to knowledge. Curr Biol. 2017;27:R770–R783. https://doi.org/10.1016/j.cub.2017.05.055
3. Shakoor N, Lee S, Mockler TC. Review: new sensors and data-driven approaches—a path to next generation phenomics. Plant Sci. 2019;282:2–10. https://doi.org/10.1016/j.plantsci.2019.01.011

4. Henke M, Junker A, Neumann K, Altmann T, Gladilin E, et al. A two-step registration-classification approach to automated segmentation of multimodal images for high-throughput greenhouse plant phenotyping. Plant Methods. 2020;16:95. https://doi.org/10.1186/s13007-020-00637-x
5. Xu S, Shen J, Wei Y, Li Y, He Y, Hu H, et al. Automatic plant phenotyping analysis of melon (Cucumis melo L.) germplasm resources using deep learning methods and computer vision. Plant Methods. 2024;20:166. https://doi.org/10.1186/s13007-024-01293-1
6. Daviet B, Fournier C, Cabrera-Bosquet L, Simonneau T, Cafier M, Romieu C, et al. Ripening dynamics revisited: an automated method to track the development of asynchronous berries on time-lapse images. Plant Methods. 2023;19:146. https://doi.org/10.1186/s13007-023-01125-8
7. Yang M, Chen R, Chen D, Wang H, Wang X, Hu F, et al. Multimodal learning on RGB-D image for precise litchi phenotyping and weight estimation. Plant Methods. 2025;21:158. https://doi.org/10.1186/s13007-025-01472-8
8. Williams D, Britten A, McCallum S, Jones H, Aitkenhead M, Karley A, et al. A method for automatic segmentation and splitting of hyperspectral images of raspberry plants collected in field conditions. Plant Methods. 2017;13:74. https://doi.org/10.1186/s13007-017-0226-y
9. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of CVPR. 2015:3431–3440. https://doi.org/10.1109/CVPR.2015.7298965
10. Ronneberger O, Fischer P, Brox T. U-Net: convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention. 2015:234–241. https://doi.org/10.1007/978-3-319-24574-4_28
11. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: Proceedings of CVPR. 2017:936–944. https://doi.org/10.1109/CVPR.2017.106
12. Zhao H, Shi J, Qi X, Wang X, Jia J. Pyramid scene parsing network. In: Proceedings of CVPR. 2017:6230–6239. https://doi.org/10.1109/CVPR.2017.660
13. Chen LC, Zhu Y, Papandreou G, Schroff F, Adam H. Encoder–decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of ECCV. 2018:833–851. https://doi.org/10.1007/978-3-030-01234-2_49
14. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of CVPR. 2016:770–778. https://doi.org/10.1109/CVPR.2016.90

15. Yu C, Wang J, Peng C, Gao C, Yu G, Sang N. BiSeNet: bilateral segmentation network for real-time semantic segmentation. In: Proceedings of ECCV. 2018:325–341. https://doi.org/10.1007/978-3-030-01261-8_20
16. Xu J, Xiong Z, Bhattacharyya SP. PIDNet: a real-time semantic segmentation network inspired by PID controllers. In: Proceedings of CVPR. 2023:19529–19539. https://doi.org/10.1109/CVPR52729.2023.01871
17. Ha Q, Watanabe K, Karasawa T, Ushiku Y, Harada T. MFNet: towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. 2017:5108–5115. https://doi.org/10.1109/IROS.2017.8206396
18. Sun Y, Zuo W, Liu M. RTFNet: RGB-thermal fusion network for semantic segmentation of urban scenes. IEEE Robot Autom Lett. 2019;4:2576–2583. https://doi.org/10.1109/LRA.2019.2904733
19. Zhang J, Liu H, Yang K, Hu X, Liu R, Stiefelhagen R. CMX: cross-modal fusion for RGB-X semantic segmentation with Transformers. IEEE Trans Intell Transp Syst. 2023;24:14679–14694. https://doi.org/10.1109/TITS.2023.3300537
20. Shen Z, Wang J, Weng Y, Pan Z, Li Y, Wang J. ECFNet: efficient cross-layer fusion network for real time RGB-thermal urban scene parsing. Digit Signal Process. 2024;151:104579. https://doi.org/10.1016/j.dsp.2024.104579
21. Liu J, Liu Z, Wu G, Ma L, Liu R, Zhong W, et al. Multi-interactive feature learning and a full-time multi-modality benchmark for image fusion and segmentation. In: Proceedings of ICCV. 2023:8115–8124. https://doi.org/10.1109/ICCV51070.2023.00745
22. Zhang H, Zuo X, Jiang J, Guo C, Ma J. MRFS: mutually reinforcing image fusion and segmentation. In: Proceedings of CVPR. 2024:26974–26983. https://doi.org/10.1109/CVPR52733.2024.02547
23. Dai Y, Gieseke F, Oehmcke S, Wu Y, Barnard K. Attentional feature fusion. In: Proceedings of WACV. 2021:3559–3568. https://doi.org/10.1109/WACV48630.2021.00360
24. Zhou W, Liu J, Lei J, Yu L, Hwang JN. GMNet: graded-feature multilabel-learning network for RGB-thermal urban scene semantic segmentation. IEEE Trans Image Process. 2021;30:7790–7802. https://doi.org/10.1109/TIP.2021.3109518
25. Lai W, Zeng F, Hu X, Li W, He S, Liu Z. MEFNet: multi-expert fusion network for RGB-thermal semantic segmentation. Eng Appl Artif Intell. 2023;125:106638. https://doi.org/10.1016/j.engappai.2023.106638

26. Zhang J, Liu R, Shi H, Yang K, Reiß S, Peng K, et al. Delivering arbitrary-modal semantic segmentation. In: Proceedings of CVPR. 2023:1136–1147. https://doi.org/10.1109/CVPR52729.2023.00116
27. Liu C, Feng Q, Sun Y, Li Y, Ru M, Xu L. YOLACTFusion: an instance segmentation method for RGB-NIR multimodal image fusion based on an attention mechanism. Comput Electron Agric. 2023;213:108186. https://doi.org/10.1016/j.compag.2023.108186
28. Li X, Qiao L, Yang C. AgriFusion: multiscale RGB–NIR fusion for semantic segmentation in airborne agricultural imagery. AgriEngineering. 2025;7:388. https://doi.org/10.3390/agriengineering7110388
29. Li A, Wang C, Wang A, Sun J, Gu F, Zhang T. YOLO-MSRF: a multimodal segmentation and refinement framework for tomato fruit detection and segmentation with count and size estimation under complex illumination. Agriculture. 2026;16:277. https://doi.org/10.3390/agriculture16020277
30. Wang P, Li C, Wei L. YOLOv11-WFD: a multimodal grape segmentation framework with wavelet convolution, FasterNeXt, and dynamic upsampling for intelligent harvesting. Agronomy. 2026;16:679. https://doi.org/10.3390/agronomy16070679
31. Shinoda R, Kataoka H, Hara K, Noguchi R. Transformer-based ripeness segmentation for tomatoes. Smart Agric Technol. 2023;4:100196. https://doi.org/10.1016/j.atech.2023.100196
32. Rong Q, Hu C, Hu X, Xu M. Picking point recognition for ripe tomatoes using semantic segmentation and morphological processing. Comput Electron Agric. 2023;210:107923. https://doi.org/10.1016/j.compag.2023.107923
33. Pisharody SN, Duraisamy P, Rangarajan AK, Whetton RL, Herrero-Langreo A. Precise tomato ripeness estimation and yield prediction using transformer-based segmentation-SegLoRA. Comput Electron Agric. 2025;233:110172. https://doi.org/10.1016/j.compag.2025.110172
34. Chen W, Rao Y, Wang F, Zhang Y, Wang T, Jin X, et al. MLP-based multimodal tomato detection in complex scenarios: insights from task-specific analysis of feature fusion architectures. Comput Electron Agric. 2024;221:108951. https://doi.org/10.1016/j.compag.2024.108951
35. Zhang Y, Rao Y, Chen W, Hou W, Yan S, Li Y, et al. A dataset of multimodal images of tomato fruits at different stages of maturity. China Scientific Data. 2025;10:1–16. https://doi.org/10.11922/11-6035.csd.2023.0154.zh

36. Chen W, Rao Y, Wang F, Zhang Y, Yang Y, Luo Q, et al. A dataset of grape multimodal object detection and semantic segmentation. China Scientific Data. 2025;10:1–16. https://doi.org/10.11922/11-6035.csd.2023.0132.zh
37. Howard A, Sandler M, Chu G, Chen LC, Chen B, Tan M, et al. Searching for MobileNetV3. In: Proceedings of ICCV. 2019:1314–1324. https://doi.org/10.1109/ICCV.2019.00140
38. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. ImageNet: a large-scale hierarchical image database. In: Proceedings of CVPR. 2009:248–255. https://doi.org/10.1109/CVPR.2009.5206848
39. Milletari F, Navab N, Ahmadi SA. V-Net: fully convolutional neural networks for volumetric medical image segmentation. In: Fourth International Conference on 3D Vision. 2016:565–571. https://doi.org/10.1109/3DV.2016.79
40. Sudre CH, Li W, Vercauteren T, Ourselin S, Cardoso MJ. Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations. In: Deep Learning in Medical Image Analysis. 2017:240–248. https://doi.org/10.1007/978-3-319-67558-9_28