ORIGINAL RESEARCH **Open access**

Reducing Annotation Effort for Field Vegetable Instance Segmentation via Unsupervised Image-to-Image Translation

Nguyen Thanh Huy[1*], Pham Quang Minh[1], Le Thi Bich[2], Tran Van Nam[3]

Abstract

Because of the steadily worsening labor shortage in agriculture, the push toward automation continues to grow. This pattern stands out clearly in field vegetable farming, where most costs are labor-intensive. Smart camera setups are essential for handling the most demanding jobs. Standard advanced learning models need massive volumes of annotated data to perform dependable detection and segmentation. The work of collecting and annotating data is further complicated by the many different field situations that datasets must cover. This study proposes an unsupervised technique that integrates diverse field environments and examines its application to new lighting conditions, various plant development phases, and different cultivars. A Cycle-Consistent Adversarial Network (CycleGAN) serves this goal by matching different lighting levels and backgrounds to the existing training data diversity, all without any extra annotation work. Using CycleGAN for exposure adjustment reduced the labelling workload by up to 5 times compared to a varied dataset, since fewer differences are needed to achieve solid segmentation. Extending the method to additional growth phases and to a separate cabbage variety highlights the opportunity for broader data generalisation.

Keywords Cabbage, Instance segmentation, CycleGAN, Annotation effort

*Correspondence:
Nguyen Thanh Huy
huy.nguyen@vnua.edu.vn
[1] Department of AI for Dragon Fruit and Longan Flowering Induction Prediction, Faculty of Agriculture, Vietnam National University of Agriculture, Hanoi, Vietnam
[2] Department of Machine Learning for Tropical Storm and Flood Impact on Rice, Faculty of Agriculture, Can Tho University, Can Tho, Vietnam
[3] Department of Smart Nutrient Management, Faculty of Agriculture, Hue University, Hue, Vietnam

Introduction

Growing interest in sustainable food systems that limit synthetic pesticides and mineral fertilisers is driving the expansion of eco-friendly, organic farming. This development causes major shifts in production, as tasks such as weed control and harvesting are highly labour-intensive and must be automated to be cost-effective. The majority of such automation depends on advanced sensor inputs, especially camera systems. AI-driven image processing techniques can deliver the critical details farmers need to enhance their operations.

As we advance, crop management will likely place greater emphasis on individual plants throughout the growing season. Farmers might apply plant-specific practices to improve growth by optimising fertiliser use, targeted spraying, mechanical weed removal, or harvesting [1, 2]. As farmers increasingly rely on camera-supported crop oversight for financial support, the underlying algorithms need to be adjusted for real agricultural settings.

This is an open access article distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0), which permits non-commercial use, distribution, reproduction, and adaptation in any medium or format, provided that appropriate credit is given to the original author(s) and source, a link to the license is provided, and any changes made are indicated. Adapted material must be distributed under the same license.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

Current leading plant detection algorithms rely heavily on neural networks, given their growing potential. Still, these networks demand very large quantities of labelled examples for each specific situation. The wide range of difficult external factors, along with the requirement for systems to work across changing growth stages, pushes this data demand even higher [1].

Providing artificial shade or light directly in the field proves difficult and often unrealistic. It also restricts how freely the sensor platform can move over the ground. Systems using drones face similar restrictions and cannot use controlled lighting or shading. Based on the precision level a neural network requires, a dependable camera solution might require a collection of millions of labelled images.

A promising approach to reduce labelling work, improve reliability, and shrink the image count is to use image-to-image translation networks. These rely on Generative Adversarial Networks (GANs). GANs were first introduced by Goodfellow *et al.* [3] and involve two opposing networks. One generator learns typical dataset patterns to create new images, while the discriminator judges how realistic those images are. Since their introduction, GANs have advanced rapidly, creating new opportunities for generating and altering images. Image-to-image translation represents one key evolution. Rather than generating images solely from a dataset, as in traditional GANs, this method modifies a given input image using knowledge from another dataset. It works by connecting two GANs in a loop. The initial GAN shifts an input from the source set toward the style of the target set. The resulting picture at this stage looks completely different from the starting image. True refinement happens in the second GAN, which converts the target-style image back toward the source domain while staying faithful to the original input. This back-and-forth loop lets one visual domain adapt to another. The approach is referred to as CycleGAN (CG) [4].

CG can adjust fluctuating environmental or plant-related factors to align with the training conditions of an instance segmentation network [5]. As a result, the network would train on just one environmental setup, cutting both the training data volume and the annotation workload needed for the instance segmentation network to master all important features. **Figure 1a** displays a sample camera image affected by strong exposure. **Figure 1b** shows the same image after standardisation via CycleGAN-based image-to-image translation.

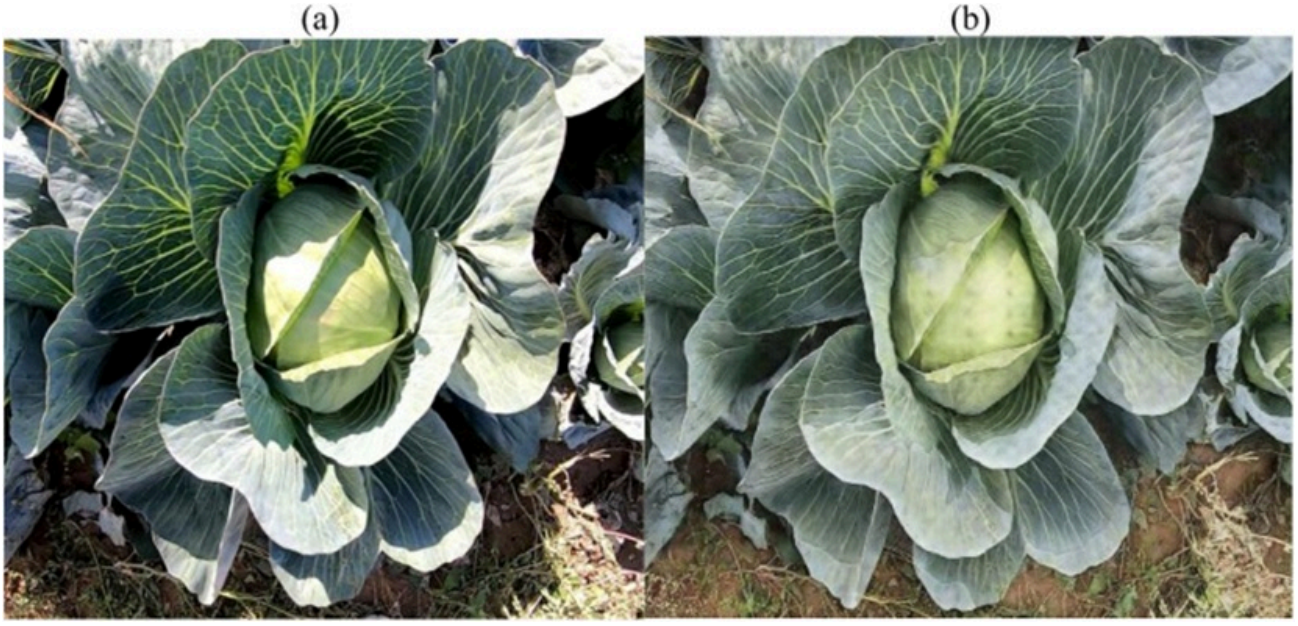


Figure 1. Raw camera image (a) with an intense exposure influence and the corresponding CG image (b) with adapted exposure.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

This translation process can be applied to every camera image, allowing the neural network always to see consistent conditions and variety. Such a strategy could greatly reduce the workload of creating reliable camera-based neural network tools and make it easier to handle multiple varieties and field conditions.

Neural network approaches for object detection and semantic segmentation underpin many research efforts [1, 2, 6, 7]. RGB images remain the main input for tracking plant growth or spotting weeds. However, bringing these techniques into actual use requires substantial investment to achieve reliability, as agricultural environments are subject to numerous changing influences. Several strategies exist to lessen the harmful impact of these conditions on neural network performance.

One direction is to reduce external disturbances with additional equipment, for example, by adding artificial shade or light around the camera's field of view [6]. This stabilises conditions and allows a smaller dataset to deliver good segmentation results. That said, physical shading limits movement on uneven ground, and drones cannot apply such controls. Stronger sensors or depth data offer another path, since depth remains mostly unaffected by lighting. Methods such as structure-from-motion and stereo vision work well for estimating depth outdoors [8]. Multispectral imaging is also studied for reliable weed identification [9]. While extra sensors can significantly enhance neural detection systems, their high cost and added complexity often make them impractical. Standard photo editing techniques for fixing exposure, such as gradient adjustments, can help with bright or dark spots but can lose the natural colour and contrast found in even lighting [10].

Expanding and diversifying training datasets to include every possible environment has received extensive testing. The simplest tactic is recording the target object under many different real-world conditions to create a richly varied collection. This helps the network learn recognition features that are invariant to changes in angle, form, and other variables. Because building and labelling such a collection manually takes significant time, researchers have explored ways to ease this burden. GANs can simulate lighting effects inside the training set [11], producing diversity without new fieldwork. Image augmentation also reduces problems from viewpoint shifts and lowers some collection costs [12]. Yet exposure changes, colour shifts, shadows, and backgrounds still resist effective simulation through basic augmentation. Traditional transfer learning delivers useful outcomes by borrowing features from other domains, thereby somewhat reducing labelling and collection requirements, although limitations persist [13]. Labelling aids, such as Meta's Segment-Anything-Model (SAM), help save time [14]. Even so, SAM focuses on contours, while cabbage head masks cross those boundaries, limiting its usefulness here. Unsupervised methods offer attractive avenues for adapting and improving images, as they require no labels. CG techniques for enhancement and lighting correction already exist and show strong promise [15].

This paper proposes using an image-to-image translation network to reduce the additional labelling and collection work caused by varying exposure levels. A CG pre-processes images before they reach an instance segmentation network (Mask R-CNN) [5]. Matching incoming image exposure to the segmentation model's training conditions simplifies the features required for accurate cabbage head separation. In addition to cutting labelling demands through exposure alignment, the study also investigates how well this method transfers, helping reduce the need to record separate exposure cases for other growth stages and one more variety.

Materials and Methods

Image pre-processing

The complete workflow for identifying and segmenting a cabbage head **(Figure 2)** using CG-based pre-processing consists of three primary stages.

1. The incoming RGB image is fed into the CG, where aspects such as exposure and background differences are modified.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

2. The modified image is subsequently supplied to the Mask R-CNN, which carries out instance segmentation on the cabbage head.

3. The generated binary mask from the instance segmentation is then overlaid onto the original RGB image. This enables the CG's benefits to support segmentation while preserving the colour details from the untouched RGB image for further examination.

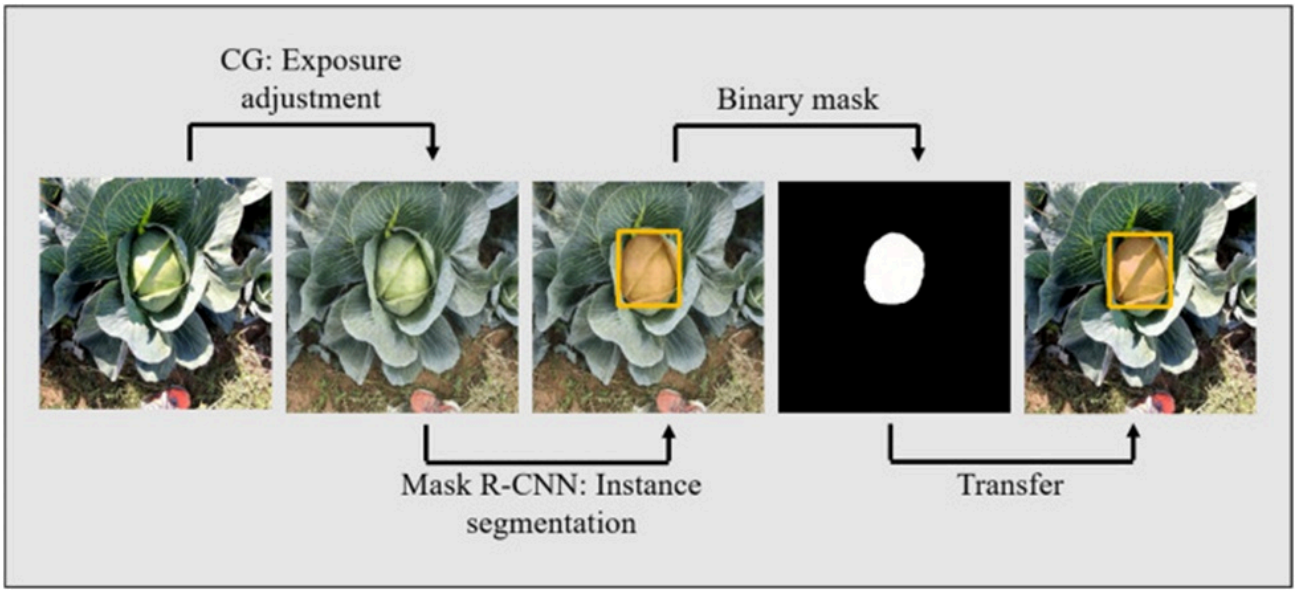


Figure 2. Process of detecting and segmenting a cabbage head with the pre-processing by a CG.

Experimental setup

To examine the various exposure conditions, the training and testing recordings were conducted close to Stuttgart, Germany (48.659693°N, 9.255520°E). No extra mechanical stabilisation, shading, or artificial lighting was applied during image capture. Images were acquired using a GoPro camera (Hero 7, GoPro Inc., San Mateo, CA, USA) at 1920×1440 pixels (1920×1440p@60fps 4:3) at 60 frames per second, with the camera travelling along the crop row at roughly 1 m/s and maintaining a fixed height of 90 cm. A vertical camera orientation toward the plant was chosen for cabbage analysis. At advanced growth stages, even minor camera tilting severely limits the view of the cabbage because of its broad upright leaves [1]. Three primary datasets were collected.

1. Exposure dataset: In the initial dataset, five separate recording sessions at the same growth stage were conducted to establish a foundation for studying exposure effects **(Table 1)**. Capturing the same cabbage plants under varying sun positions helped isolate the impact of exposure. Two distinct spatial zones—one dedicated to training images and one to test images—were established to prevent any overlap between training and test sets [1].

Table 1. Recording parameters of the exposure dataset (L1–L5).

Label	Time	Sun height above horizon (°)	Weather condition
L1	09 am	21.30	Light cloud cover
L2	10 am	30.55	Direct sunlight
L3	1:30 pm	48.21	Direct sunlight
L4	3:30 pm	40.22	Light cloud cover

© 2024 The Author(s). | Published by Agro Intelligence Research Press

L5	6:30 pm	13.66	Direct sunlight

Owing to the sun's low angle, L5 produced the strongest shadows. **Figure 3** illustrates the same cabbage plant captured at different times of day.



Figure 3. Exposure variations L1–L5 of the same cabbage plant to analyse the influence of exposure.

2. Growth stage dataset: To assess how CycleGAN exposure adjustment affects younger plants, cabbage specimens were documented at roughly one-week intervals using the same camera configuration. Six distinct growth stages were captured, spanning from BBCH stage 42 (W1) to BBCH stage 48 (W6), when 80% of the anticipated cabbage volume is attained **(Figure 4)** [16]. Images representing the different exposure levels were obtained at BBCH stage 48 (W6).

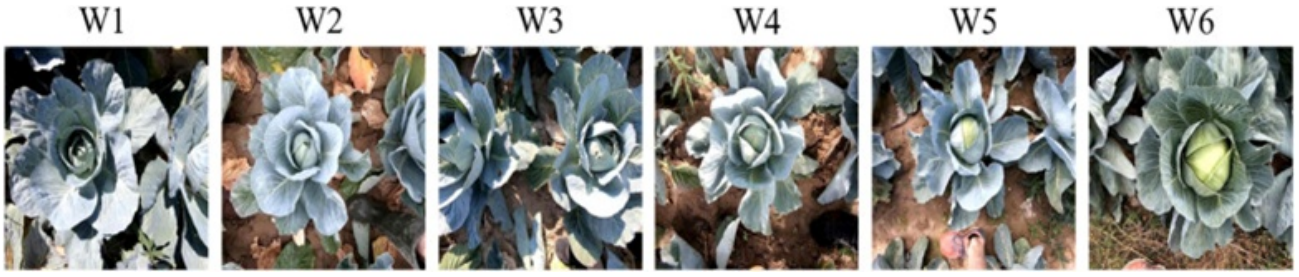


Figure 4. Growth stage variations (W1–W6) of the same cabbage variety varying from BBCH stage 42 to BBCH stage 48.

Table 2 presents the recording settings used for the growth stage dataset.

Table 2. Recording parameters of the growth stage dataset (W1–W6).

Stage	Date	Time	Sun height above horizon (°)	Weather condition
W1	26.07.2020	2:30 pm	58.17	Direct sunlight
W2	04.08.2020	2:00 pm	57.75	Light cloud cover
W3	09.08.2020	16:30 am	40.44	Direct sunlight
W4	18.08.2020	10:30 am	39.00	Direct sunlight
W5	25.08.2020	10 am	32.95	Light cloud cover
W6	04.09.2020	09 am	21.30	Shallow cloud cover

3. Variety Transfer Dataset: To examine the adaptability of the CycleGAN method to a different cultivar, red cabbage and white cabbage plants at matching growth stages were recorded in neighbouring fields (48.68185°N, 9.24825°E) across three exposure conditions, Lt1, Lt2, and Lt3 **(Figure 5)**. **Table 3** details the recording settings for the various exposure levels and cultivars.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

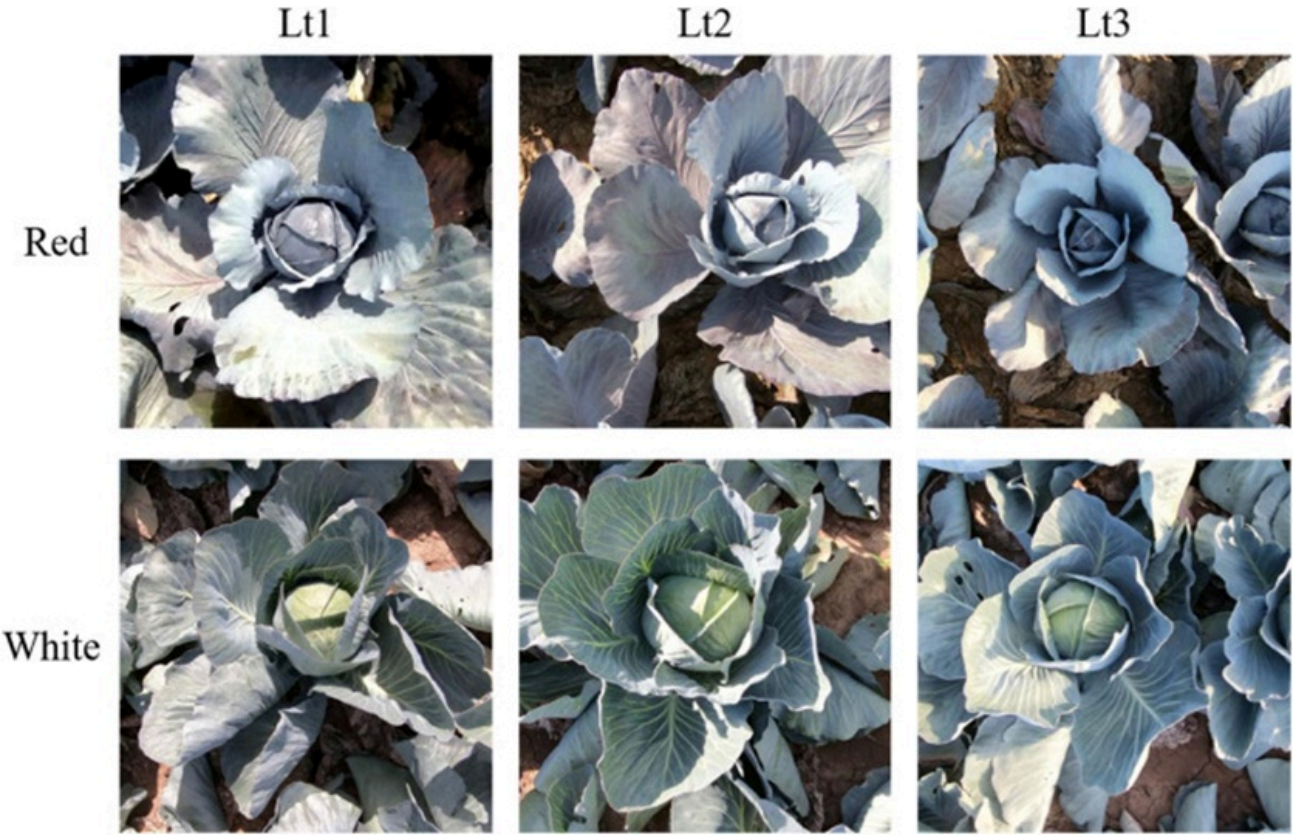


Figure 5. White and red cabbage plants at three different exposure variations (Lt1–Lt3).

Table 3. Recording parameters of the variety transfer dataset (Lt1–Lt3).

Exposure scenario	Date: 24.08.2022 time:	Height of the sun above the horizon (°)	Weather condition
Lt1	09:40 am	30.21	direct sunlight
Lt2	12:30 pm	50.61	direct sunlight
Lt3	6:40 pm	15.66	direct sunlight

Detection and segmentation performance was measured using the mean average precision (mAP) for each class. The mAP computation involves determining precision **(Eq. 1)**, recall **(Eq. 2)**, and the intersection over union (IoU).

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

Precision indicates the proportion of correctly identified pixels among all detected pixels, calculated as the ratio of true positives (TP) to false positives (FP). Recall reflects the proportion of all correctly segmented pixels (TP) relative to the overlooked pixels (False Negative—FN). IoU is the overlap between the ground-truth and predicted masks, divided by the sum of their areas. Once precision and recall are obtained, recall is plotted on the x-axis and precision on the y-axis. The

© 2024 The Author(s). | Published by Agro Intelligence Research Press

mAP is calculated from the area under the precision-recall curve. For the ultimate mAP value, the IoU threshold defining a true positive is incrementally increased from 0.5 to 0.95 in 0.05 increments, and the resulting scores are averaged [1, 17].

Model prediction accuracy on unseen data can vary depending on how the annotated data is split between the training and validation sets. Training only a single model makes the outcome vulnerable to this variability and thus less reliable. To mitigate this, K-fold cross-validation is employed. In this method, the training and validation data are pooled and randomly split into K equally sized groups. K separate model versions are then trained, each using a different validation group [18]. A modified five-fold cross-validation was applied to assess the instance segmentation outcomes. Five distinct splits of training and validation images were used for every network configuration. Test images were excluded from cross-validation because they belong to a different image domain than the training data. Training for both the Mask R-CNN and CycleGAN took place on a workstation equipped with an AMD Ryzen Threadripper 2920X 12-core processor (64 GB RAM) and a 24 GB graphics card (GeForce RTX 3090).

Mask R-CNN

Mask R-CNN is a framework for object instance segmentation. It builds on the Faster R-CNN framework for bounding-box detection by adding an extra branch for mask generation. The faster R-CNN includes two core elements. The first is the Region Proposal Network (RPN), which suggests multiple Regions of Interest (RoIs) containing recognised features. The second is the Region-of-Interest-Pooling (RoIPool), which extracts compact feature maps from each RoI for classification and bounding-box adjustment. Beyond these, the Mask R-CNN produces a binary mask for every RoI. Consequently, the overall loss for Mask R-CNN combines losses from class prediction, bounding box regression, and object mask creation [1, 5]. Instance segmentation provides a vital foundation for linking measured values to specific individual objects, enabling plant-specific data to be assigned even to overlapping specimens [1].

The implemented Mask R-CNN utilises a ResNet50 backbone [19]. It operates on TensorFlow (GPU 2.4.0) and the Keras library (2.0.8) within Python (3.6). Networks were trained at a resolution of 512×512 pixels. The entire network was initialised with random weights and a learning rate of 0.001, with early stopping activated if the validation loss did not improve for at least 5 epochs. To prevent overfitting due to viewpoint changes, various image augmentation methods (shifts, rotations, scaling, and mirroring) were applied [12].

CycleGAN

The structure of a GAN (Generative Adversarial Network) for data simulation from a given dataset involves two fundamental parts. The generator component creates random data through a decoder architecture. The discriminator employs an encoder design and is trained on real images from the datasets. Its objective is to differentiate between genuine dataset images and the generator's simulated outputs. The generator uses the discriminator's feedback to refine its parameters and generate results that closely match the dataset, making it difficult for the discriminator to tell them apart [20].

Employing a CG enables unpaired image-to-image translation, allowing adjustment of colours and textures in an input image according to a target dataset. The CG expands the standard GAN design by using several generators and discriminators. Generators follow an autoencoder layout that compresses incoming images to key features before reconstructing them in a different visual domain. The input to the first generator consists of cabbage images with varied exposure **(Figure 6a)**. The first discriminator evaluates its output against the target dataset of uniformly exposed images. As in a typical GAN, the standard adversarial loss is computed, and the varied-exposure images are aligned with the uniform-exposure ones. In a subsequent phase, the same procedure repeats, but with the second generator receiving the first generator's outputs. The second generator's results are then assessed by the second discriminator using the source dataset. This leads homogeneous images to be converted back toward heterogeneous exposure. Examining each GAN structure separately does not yet constitute full translation. Both GANs are additionally guided by Cycle Consistency Loss, which measures the discrepancy between input and output images of the generators. The Forward Cycle

© 2024 The Author(s). | Published by Agro Intelligence Research Press

Consistency Loss compares the first generator's input from the source dataset with the second generator's output **(Figure 6a)**. The Backward Cycle Consistency Loss compares the second generator's input from the target dataset with the first generator's output **(Figure 6b)**. To maintain the original colour balance, identity loss is also computed by feeding a target-domain image to the generator and measuring the difference between its output and the target image [4]. These mechanisms enable unsupervised training on unpaired datasets by mapping the target domain onto input images.

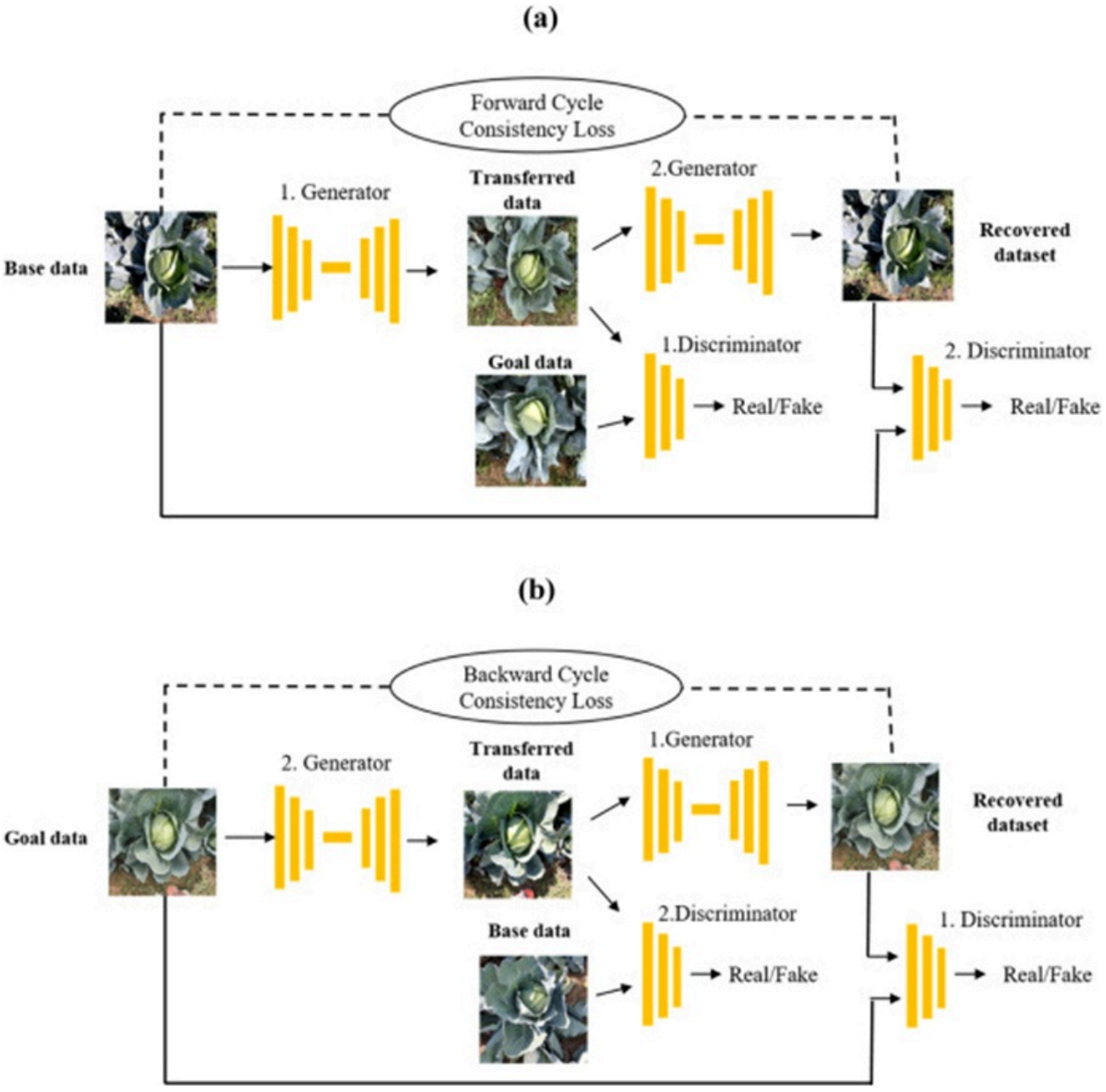


Figure 6. CycleGAN architecture for an unsupervised image translation process: (a) Forward Cycle Consistency Loss, (b) Backward Cycle Consistency Loss.

To standardise exposure conditions, the CG's training dataset (Base data) included the exposure scenarios L2–L5, with 1500 images allocated to each domain. The target dataset for exposure matching was L1. No additional labelling was required to train the CG. The CG implementation mainly relies on the TensorFlow library (GPU 2.4.0). Its generators adopt a U-Net architecture [21].

© 2024 The Author(s). | Published by Agro Intelligence Research Press

The CG was trained on 512×512-resolution images. Hyperparameters included a buffer size of 100, a batch size of 1, and a lambda value of 10 for the TensorFlow CG [22]. As unsupervised CG training lacks definitive evaluation metrics, an alternative stopping criterion was necessary. One hundred epochs were trained, and for each epoch, a test set was used to measure the resulting mAP. The epoch yielding the peak mAP was selected for all further assessments.

Network instances based on dataset variations

To examine how the CG-driven image transformation affects outcomes, two key evaluations applied separate strategies. The underlying network architectures for instance segmentation (Mask R-CNN) and image translation (CycleGAN) remained identical across all evaluations. Differences appeared only in how the training datasets were assembled and how the test datasets were pre-processed. The initial evaluation compares mAP scores across all exposure stages **(Figure 3)** to gauge potential cuts in labelling work. The follow-up evaluation checks whether the findings from the first evaluation can reduce both labelling and recording demands. In each evaluation, a core split exists between two kinds of training datasets for the instance segmentation network and two kinds of test datasets for assessment. This led to three unique network-and-test-data setups for deeper study.

1. SDN: The Single Domain Network (SDN) uses training images from a single domain. Limiting training to a single exposure level, variety, or growth stage lets the resulting segmentation performance of the instance segmentation network highlight how mismatches in the test dataset's domains affect results compared to the training set. As an illustration, **Figure 7** depicts the allocation of training and test datasets for the Mask R-CNN in the SDN method during the first evaluation (Chapter 3.1). The training dataset contains only one image domain (L1), whereas the test dataset covers all image domains (L1–L5).

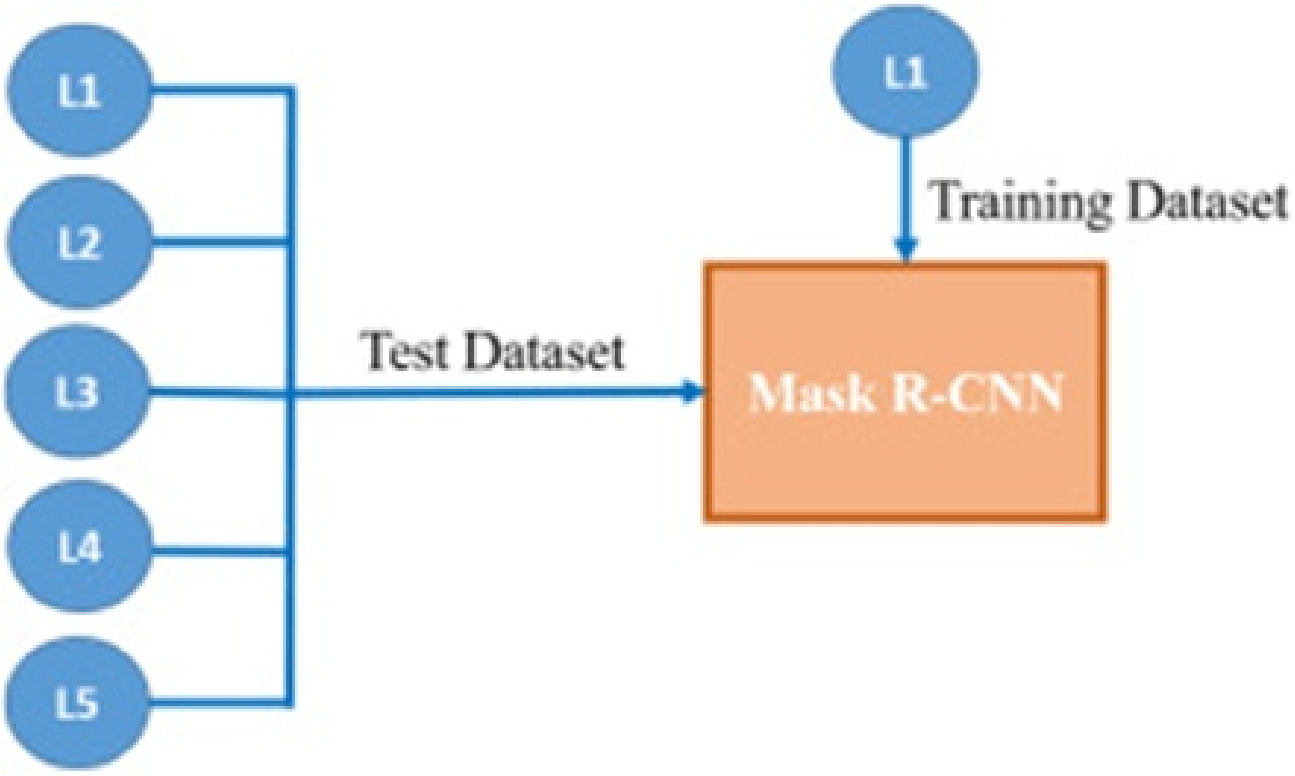


Figure 7. Schematic representation of the division of different image domains of the training and test data sets of the Single Domain Network (SDN).

Such an arrangement allows linking the standard deviation of exposure values (LAB(L)) directly to segmentation performance. A greater deviation of exposure values from the training domain L1 should produce sharper drops in

© 2024 The Author(s). | Published by Agro Intelligence Research Press

segmentation performance. Assessment of this method relied on raw test images drawn from differing exposure, growth stage, or variety settings.

2. MDN: The Multi Domain Network (MDN) receives training on images from several exposure levels, growth stages (MDN_G), or varieties (MDN_RW) in balanced shares suited to the evaluation. As shown in **Figure 8**, the Mask R-CNN training and test dataset allocation for the MDN method in the first evaluation includes all image domains (L1-L5) for both training and testing.

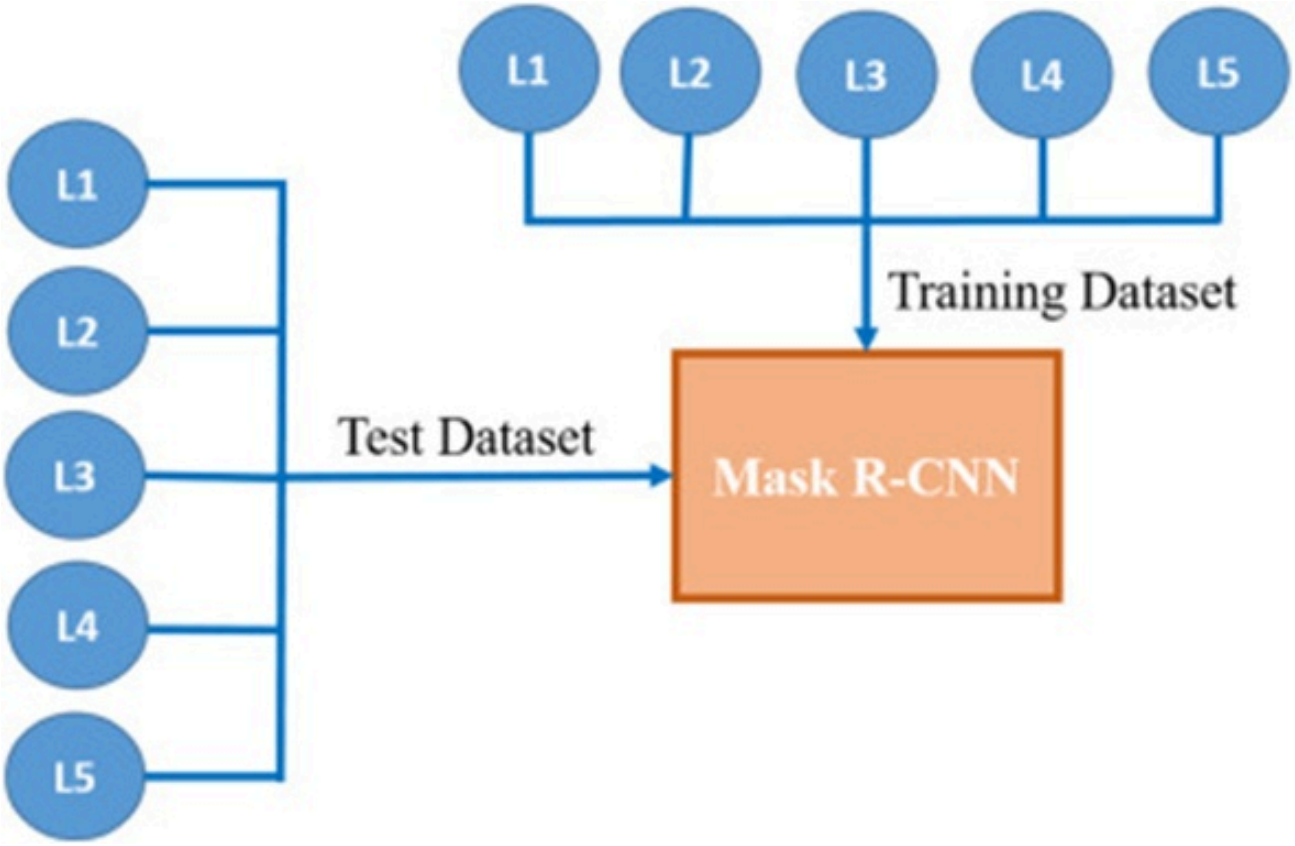


Figure 8. Schematic representation of the division of different image domains of the training and test data sets of the Multi Domain Network (MDN).

This represents a typical method for creating a varied dataset that promotes learning of features unaffected by specific domains. Evaluation used raw test images from varied exposure, growth stage, or variety conditions.

3. SDN CG: The Single Domain Network – CycleGAN (SDN CG) employs the identical network as the SDN. To study domain adaptation via the CG, test images for the SDN CG underwent CG pre-processing. **Figure 9** illustrates the allocation of the training and test datasets for the Mask R-CNN combined with the CG in the SDN CG approach. The Mask R-CNN training dataset uses only one image domain (L1), and the test dataset uses only one transformed domain (L1). L1 comes from the CG image translation, which sought to map test image domains L1-L5 to the L1 domain. The CG had access to all domains (L1-L5) during its training.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

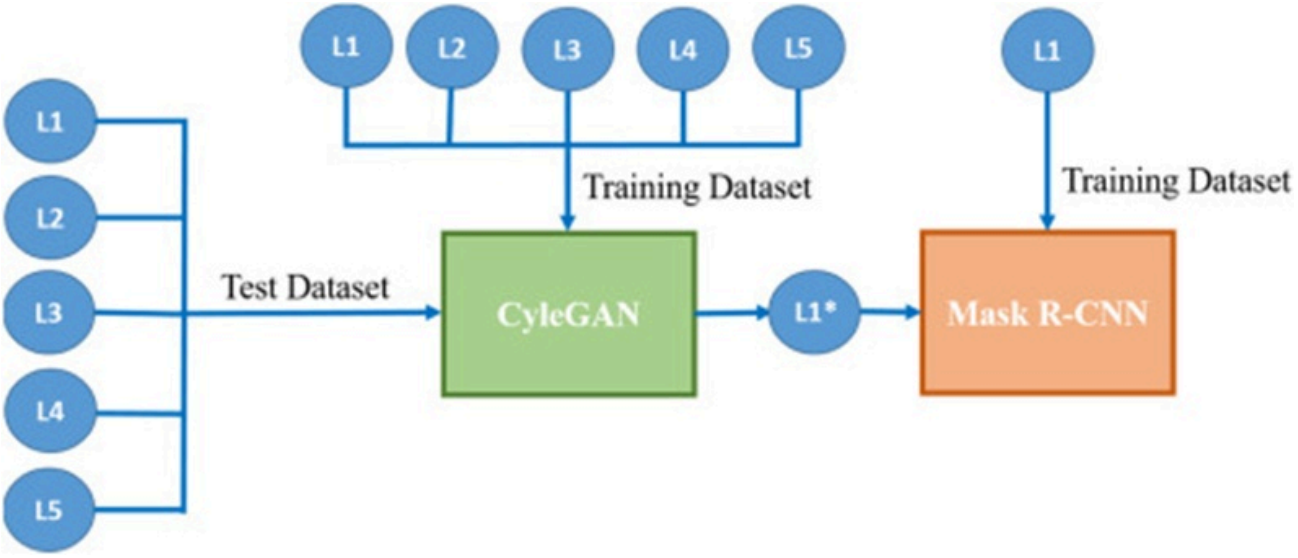


Figure 9. Schematic representation of the division of different image domains of the training and test data sets of the Single Domain Network - CycleGAN (SDN CG).

Table 4 summarises all network-test data pairs used in later evaluations, along with the corresponding training data for Mask R-CNN and CG. For each training variation of Mask R-CNN or CG, a separate network is trained, using five separate splits of training and validation data (five-fold cross-validation).

Table 4. Listing of the various network and test data in combination with the respective training data of the Mask R-CNN and the CycleGAN.

Network/Model	Test data combination	Training data	Data processing
Mask R-CNN	Training data	Training data	–
CG	Test data	–	Unprocessed
SDN	L1	–	Unprocessed
MDN	L1–L5	–	Unprocessed
SDN CG	L1	L1–L5	Pre-processed
MDN_Small	L1, L4, L5	–	Unprocessed
SDN CG_Small	L1	L1, L4, L5	Pre-processed
MDN_G	W1–W6	–	Unprocessed
SDN_R	Red_Lt1	–	Unprocessed
SDN_W	White_Lt1	–	Unprocessed
MDN_R	Red_Lt1–Red_Lt3	–	Unprocessed
MDN_RW	Red_Lt1, White_Lt1	–	Unprocessed
SDN CG_W	White_Lt1	White_Lt1–White_Lt3, Red_Lt1	Pre-processed

For insights into labelling effort in the first evaluation, each setup was trained on dataset sizes ranging from 100 to 900 images. All reported mAP figures are averages across five-fold cross-validation for each network setup.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

Results and Discussion

Analysis of the labelling effort

Figure 10 presents the three network strategies—SDN, MDN, and SDN CG—assessed against all exposure conditions. The test dataset held 250 images, including 50 from each of the five exposure conditions (**Figure 3**, L1–L5).

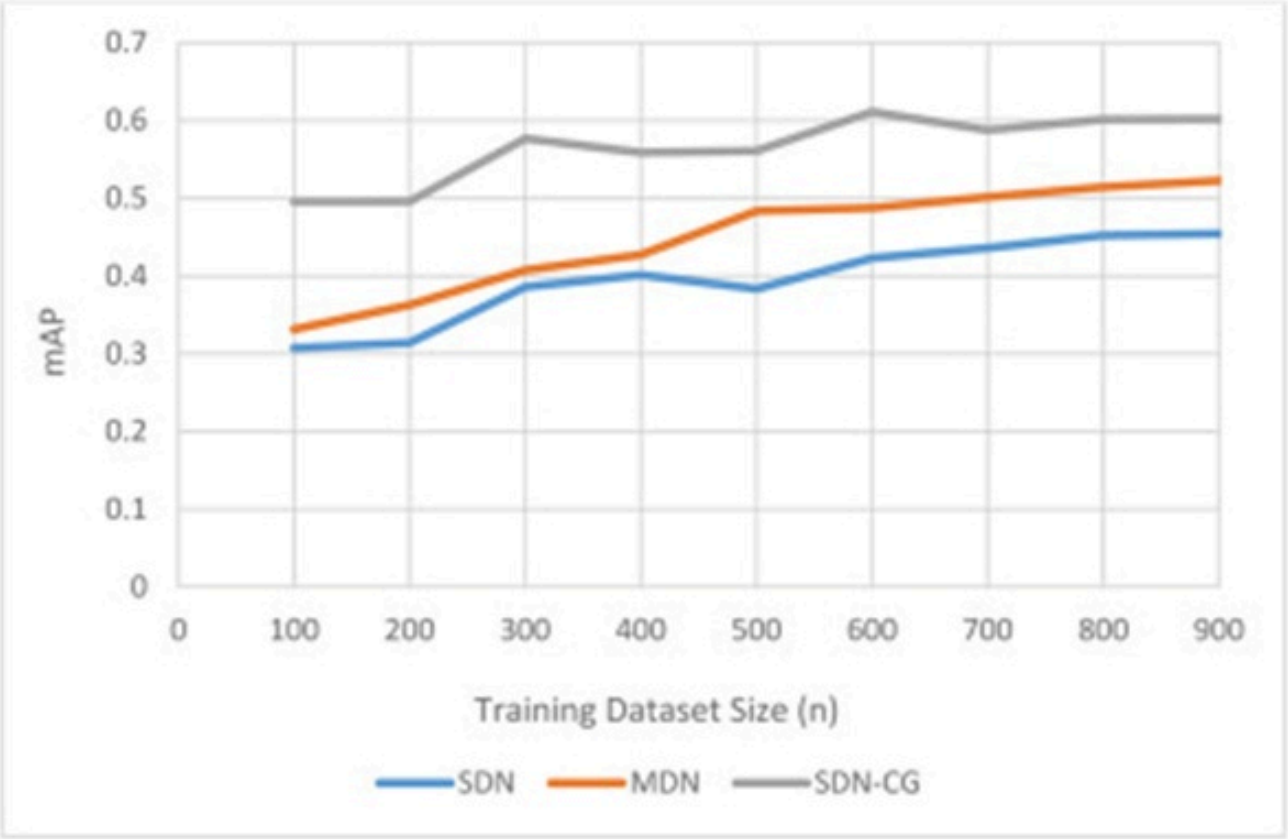


Figure 10. Mean Value of the five-fold cross-validation mAP Values of the three different Networks (SDN, MDN, SDN CG) over the different training dataset sizes.

Comparison of the SDN and MDN curves in **Figure 10** reveals that SDN networks delivered 11.8 % lower segmentation performance than MDN networks on average across all dataset sizes. At 900 training images, SDN reached only 0.454 mAP. A saturation pattern in mAP relative to training size is generally expected, shaped by object complexity, dataset makeup, and network design. Past a threshold, extra data from the same domain yields little further gain. This pattern is faintly evident in the SDN results, where adding the same-domain data yields only a minimal improvement in mAP. **Table 5** lists example mAP scores for the five exposure conditions at 600 images.

Table 5. SDN mAP values of different exposure scenarios (L1 – L5) of cabbages at BBCH stage 48.

Dataset: 600	L1 (mAP)	L2 (mAP)	L3 (mAP)	L4 (mAP)	L5 (mAP)
SDN	0.702	0.279	0.223	0.598	0.321

According to **Table 5**, SDN posted its best accuracy of 0.702 on L1, matching the training domain. L4 also scored a solid 0.598 mAP, despite minor exposure differences compared to L1. Training restricted to one exposure caused large drops

© 2024 The Author(s). | Published by Agro Intelligence Research Press

in other conditions, notably L2, L3, and L5, with performance averaging 60.9% lower than L1. These findings underline the importance of broader training data for usable segmentation masks in practice.

The MDN curve in **Figure 10** lacks clear saturation, which would require even bigger datasets. Heterogeneous data provides a richer feature set, requiring more samples before mAP plateaus. This diversity supports better handling of L2-L5 exposures. Across all sizes, MDN networks improved accuracy by 13.5% on average compared to SDN. **Table 6** presents mAP results per exposure in the test set, showing an average gain of 44.3% over L2, L3, and L5 compared to SDN. Matching L1 or L4 accuracy remains unlikely even with more data, given the heightened complexity introduced by extreme over- and underexposure.

Table 6. MDN mAP values of different exposure scenarios (L1–L5) of cabbages at BBCH stage 48.

Dataset: 600	L1 (mAP)	L2 (mAP)	L3 (mAP)	L4 (mAP)	L5 (mAP)
MDN	0.637	0.354	0.314	0.611	0.52

Examining the SDN CG curve data, an average improvement of 26% in segmentation accuracy is observed across all dataset sizes when compared with the MDN results. Concurrently, the saturation point for segmentation performance is reached considerably earlier than in either the SDN or MDN networks. This apparent plateau occurs already at a dataset size of 600 images, suggesting that additional annotation effort beyond this point is unlikely to yield meaningful gains in segmentation accuracy for this image domain. In contrast, it is anticipated that MDN networks will continue to improve and eventually reach accuracy levels similar to those of the training dataset as the training dataset size increases.

To illustrate the potential reduction in labelling effort, a comparison can be made between the mAP results of the MDN trained with 500 images and those of the SDN CG trained with 100 images. The MDN configuration, using 500 training images, comprises 100 images per exposure condition (L1–L5). Consequently, the network has access to 100 samples per exposure scenario during training, enabling it to learn exposure-specific characteristics. When evaluated across test data from all five exposure conditions, this MDN achieves an mAP of 0.48.

If the CG-based exposure adjustment applied to the test data is effective, the five distinct exposure variants in the test set are unified into a single exposure representation. Under this condition, a training set of only 100 images should theoretically be sufficient to match the performance of an MDN trained on 500 images, since that MDN likewise only utilises 100 images per exposure category. This relationship is reflected in the results, where the SDN CG model trained with 100 images attains a mAP of 0.495, closely matching the 0.48 achieved by the MDN. This demonstrates a clear relationship between dataset size and domain variance, indicating a potential reduction in labelling effort of up to fivefold. **Table 7** presents the mAP values of the SDN CG network at a dataset size of 600.

Table 7. SDN CG mAP values of different exposure scenarios (L1–L5) of cabbages at BBCH stage 48.

Dataset: 600	L1 (mAP)	L2 (mAP)	L3 (mAP)	L4 (mAP)	L5 (mAP)
SDNCG	0.733	0.593	0.4165	0.667	0.643

Improved accuracies were observed throughout, with L2 rising from 0.354 to 0.593 (a 67.5% gain). CG processing aligned exposure and colour differences, shrinking feature diversity. **Figure 11** contrasts exposure values before and after CG adaptation. RGB images were converted to the LAB colour space so that exposure data concentrate in the L channel (LAB(L)), with low values for underexposure and high values for overexposure. Pixel counts per value were plotted. **Table 8** reports standard deviations (std) of pixel counts across LAB(L) ranges.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

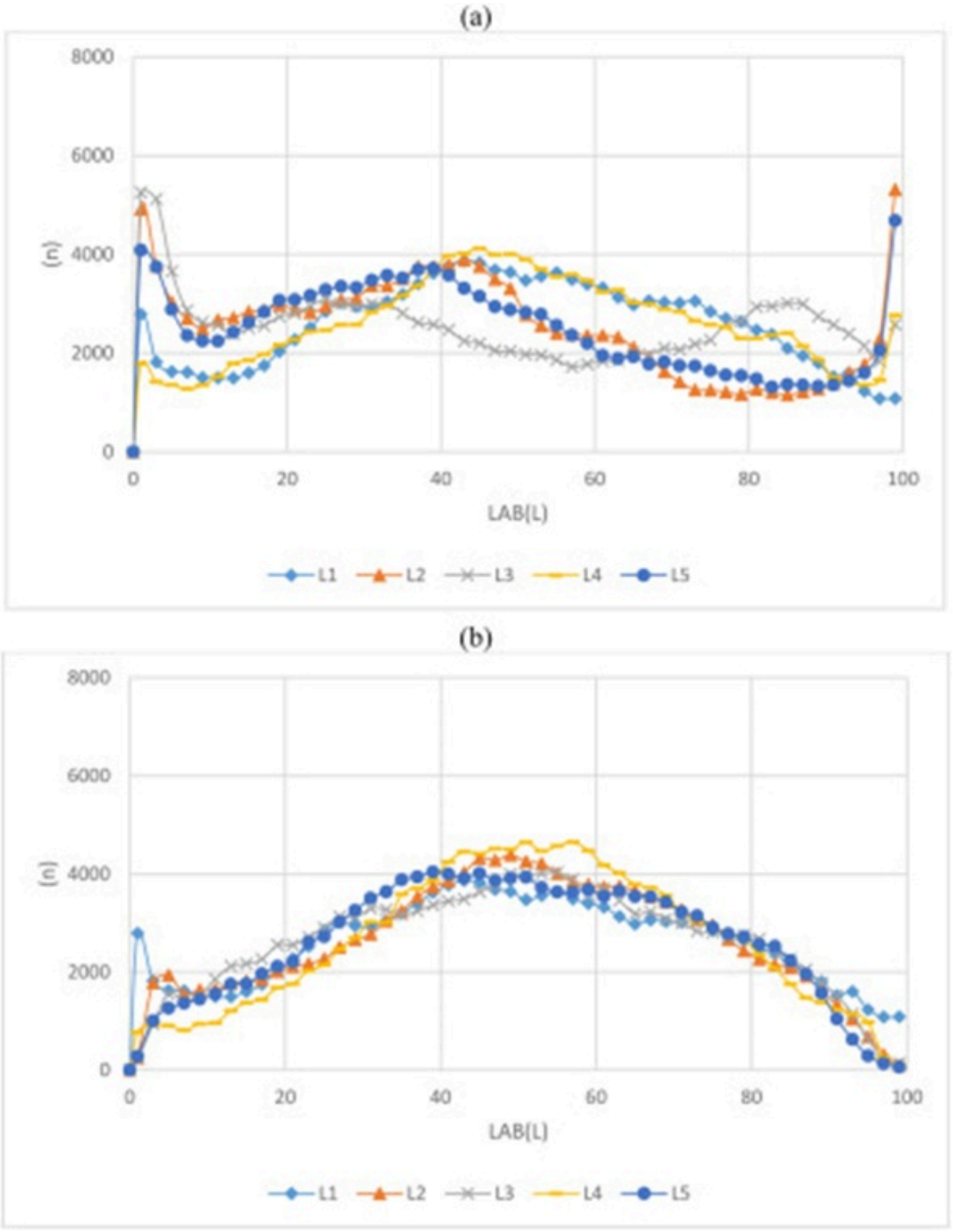


Figure 11. Exposure channel comparison (LAB(L)); (a) unprocessed images; (b) adapted images by the CycleGAN.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

Table 8. Standard deviation (std) values of the pixel count of LAB(L) ranges of the unprocessed and adapted images from datasets L1 to L5.

Pixel count	LAB(L) 0–10	LAB(L) 90–100	LAB(L) 0–100
Unprocessed images (std)	896	659	572
Adapted images (std)	402	281	262

Figure 11a reveals widespread areas in the 0-10 and 90-100 ranges, marking over- and underexposed zones. **Figure 11b** shows curves realigned toward L1. Adaptation cut standard deviation in 0-10 from 896 to 402 and in 90-100 from 659 to 281 **(Table 8)**. Overall deviation fell from 572 to 262. **Figure 12** repeats the adaptation example across all five exposures (L1–L5).



Figure 12. Images of cabbages of BBCH stage 48 adapted by the CycleGAN to exposure scenario L1.

Extra computational load from CG pre-processing remains low. Though training is compute-heavy, only one encoder-decoder part runs per image at inference, adding just 0.027 s per image on the given hardware. By contrast, Mask R-CNN takes 0.15 s per image for segmentation.

Analysis of the transferability

The previous section 3.1 illustrated that employing a CG to correct exposure levels yields major enhancements in segmentation performance, even when datasets are relatively small. Beyond lowering labelling demands, the current section investigates whether this technique can also decrease the workload of capturing data for additional varieties. Only outcomes based on the 600-image dataset are reviewed, as this volume corresponds to the anticipated saturation level of the mAP curve, where additional images yield negligible further improvement under the SDN CG configuration. To clarify the interpretation of mAP figures throughout the evaluations, scores tied to domains included in the training are highlighted in blue, whereas those from domains missing from the Mask R-CNN or CG training sets are shown in orange and bold. Orange entries particularly illustrate the degree of transferability and opportunities to cut recording requirements. Three distinct transferability assessments are considered.

1. Adaptivity to unknown exposure scenarios: All prior tests included exposure stages L1–L5 within the training sets for both CG and MDN networks. Here, the CG and MDN networks are intentionally missing exposure stages L2 and L3 to test adaptation to unfamiliar lighting situations. **Table 9** presents the mAP results for L1-L5.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

Table 9. mAP values of cabbages of known and unknown exposure scenarios analysed by a Mask R-CNN based on a heterogeneous and adapted homogeneous training dataset.

Dataset: 600	L1 (mAP)	L2 (mAP)	L3 (mAP)	L4 (mAP)	L5 (mAP)
MDN_Small	0.651	**0.35**	**0.268**	0.618	0.527
SDNCG_Small	0.732	**0.532**	**0.408**	0.66	0.662

Findings from Table 9 reveal that shrinking the MDN training set from 5 to 3 domains results in only trivial improvements in accuracy on the retained domains. Excluding L2 and L3 causes a tiny 0.04-point decline in L2 accuracy, yet a substantial 14.6% drop in L3 accuracy. The SDN CG_Small network performs almost unchanged on L1, L4, and L5, consistent with these having already attained mAP saturation at 600 images. L3 segmentation accuracy remains largely stable compared to the complete SDN CG network. L2, however, declines by 10.2%, from 0.593 to 0.532 mAP. Although this drop is meaningful, it still exceeds the MDN_Small result by 52% and stays below the 14.6% loss. Overall, transferability appears somewhat constrained when training covers just three exposure stages. Training with four or five exposure stages would likely support stronger generalisation to novel exposures.

2. Adaptivity to earlier growth stages: Younger developmental phases are examined to eliminate the need for separate recordings with varying exposures for each phase **(Figure 4)**. In the MDN_G benchmark network, images from earlier phases were evenly distributed across the training set. MDN_G seeks to cultivate diverse plant features spanning several phases. With a total of 600 training images, MDN_G provides 100 images for each of the growth stages, ranging from BBCH 42 to 48. MDN_G acts as the reference network, comparable to the MDN discussed in Chapter 3.1. **Table 10** contrasts mAP values among MDN_G, SDN, and SDN CG.

Table 10. mAP values of cabbages of different growth stages (BBCH 42 -48) analysed by a Mask R-CNN based on a heterogeneous and homogeneous training dataset and an adapted test dataset.

Dataset: 600	W1 (mAP)	W2 (mAP)	W3 (mAP)	W4 (mAP)	W5 (mAP)	W6 (mAP)
MDN_G	0.08	0.253	0.196	0.221	0.431	0.632
SDN	**0**	**0.044**	**0.012**	**0.11**	**0.379**	0.711
SDNCG	**0.06**	**0.177**	**0.207**	**0.205**	**0.519**	0.711

Table 10 demonstrates that shrinking cabbage heads become progressively harder to segment precisely, owing to their reduced presence in the frame and increased leaf overlap. With 100 images per stage, MDN_G still manages adequate segmentation at later stages, W5 and W6. Beginning at W4 and moving earlier, task difficulty escalates, reducing mAP as accurate separation becomes challenging. Larger collections would be essential to address this trend. SDN outcomes, based solely on W6 data, emphasise the requirement for multiple stages in training. Detection accuracy falls sharply from W4 onward. SDN CG delivers a 15.7 % boost at the final two stages (W5 and W6) over MDN_G. From W4 earlier, however, variations in size and visibility exceed the network's capacity to generalise from the training data, yielding no mAP benefit from exposure correction alone compared to MDN_G. On average, accuracy on the unknown domains W1-W5 declined by 1.1% relative to the MDN_G set.

Nevertheless, results indicate that exposure correction can produce outcomes approaching those of heterogeneous training without the cost of multi-stage recordings, albeit with restrictions. **Figure 13** presents the six growth stages together with their SDN CG counterparts.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

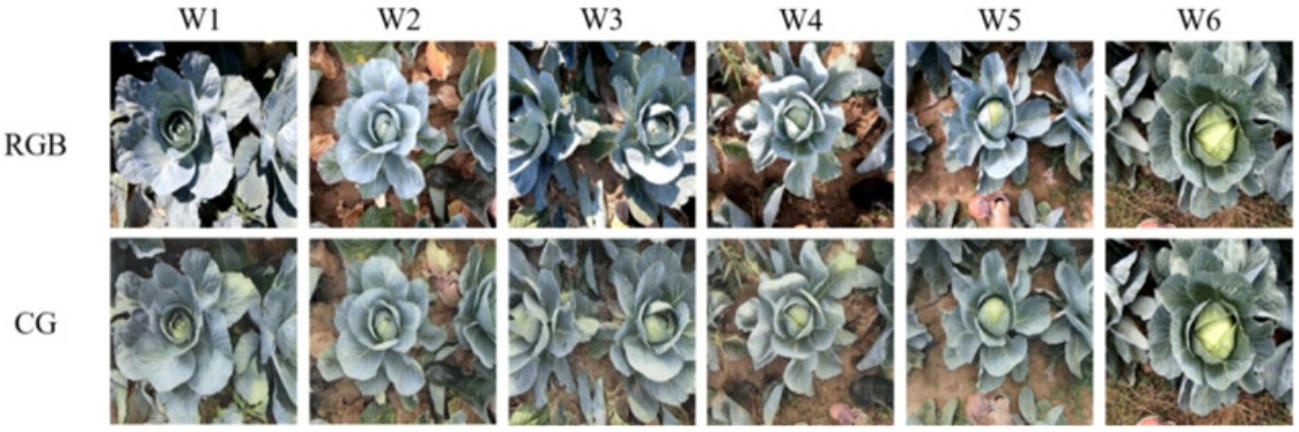


Figure 13. Exposure adjustment by the CG for different growth stages of cabbage from BBCH stage 42 (W1) to BBCH stage 48 (W6).

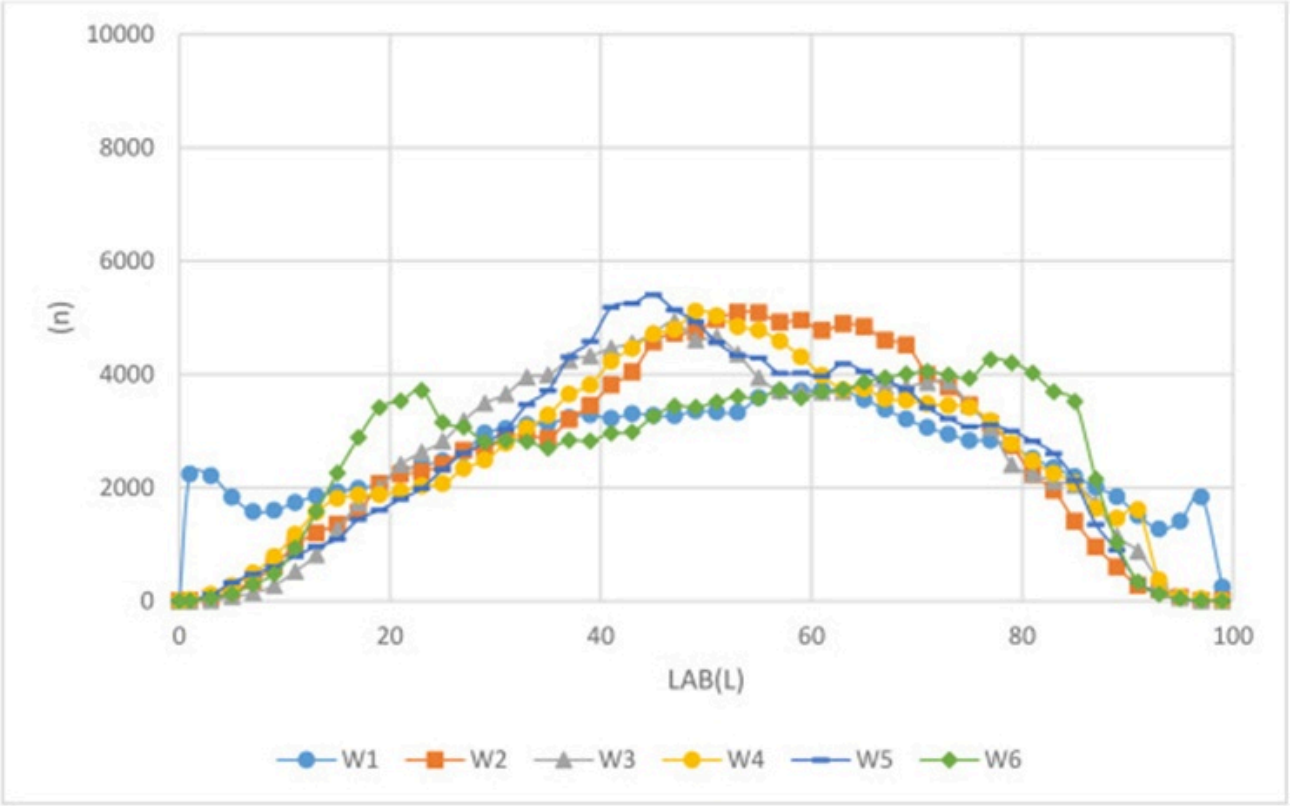
The CG achieved good visual exposure matching across stages despite being trained exclusively on W6 material. Segmentation hurdles derive not solely from lighting but primarily from reduced cabbage dimensions and visibility, which restrict precision at early stages. **Figure 14** depicts the pixel distributions in the L channel of the LAB colour space, which represent exposure. **Table 11** provides standard deviations of pixel counts in various LAB(L) intervals.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

(a)



(b)



© 2024 The Author(s). | Published by Agro Intelligence Research Press

Figure 14. LAB(L) colour channel comparison of the different growth stages with unprocessed (a) and adapted (b) images.

Table 11. Standard deviation (std) values of the pixel count of LAB(L) ranges of the unprocessed and adapted images from datasets W1-W6.

Pixel count	LAB(L) 0–10	LAB(L) 90–100	LAB(L) 0–100
Unprocessed images (std)	917	1135	743
Adapted images (std)	603	436	498

Figure 14b shows exposure realignment akin to **Figure 11**. Standard deviations decreased in the 0–10 range from 917 to 603 and in the 90–100 range from 1135 to 436 **(Table 11)**. Overall, CG exposure correction reduced LAB(L) standard deviation from 743 to 498. This verifies the CG's effective harmonisation of exposure across growth stages. Primary limitations on segmentation stem more from cabbage size and visibility than lighting differences, constraining early-stage results. Investigations using expanded datasets and dividing the SDN across two or three stages could offer a deeper understanding of achievable savings in labelling and recording across multiple phases.

3. Adaptivity to other varieties: Beyond exposure, the CG can alter the general colouration of cabbage plants. This permits shifting input images into the domain of an already-trained variety that shares a similar structure and shape. Two examinations were performed. First, red cabbage images are mapped to white cabbage through CG and processed by a white cabbage network. The procedure follows **Figure 2**, with the key change being the adjustment of variety by CG in the opening step rather than exposure. Second, exposure adaptation learned from white cabbage is extended to red cabbage, enabling segmentation without fresh recordings under differing exposures **(Figure 15)**.

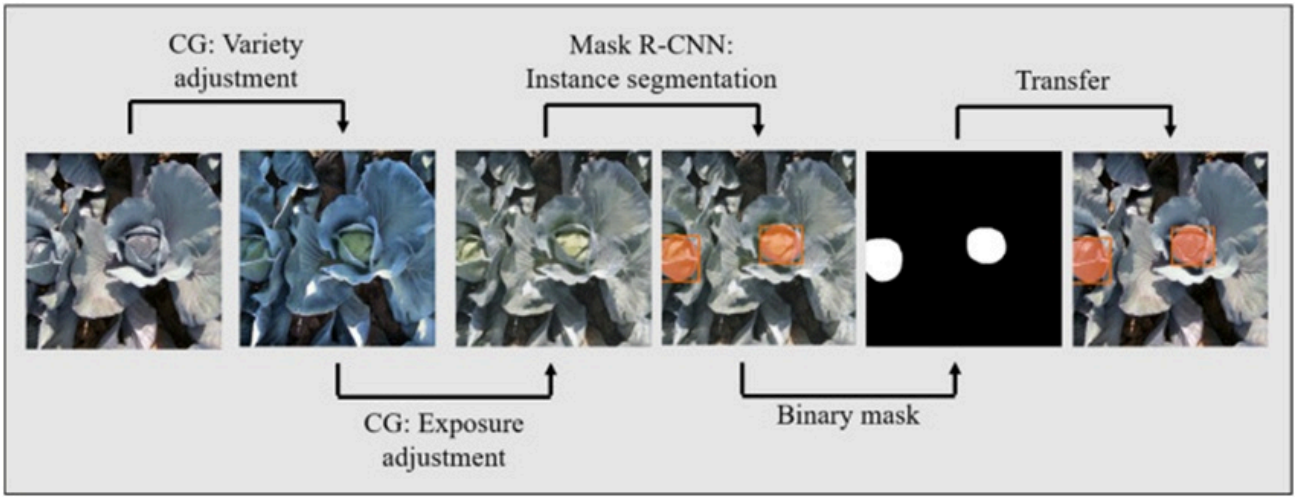


Figure 15. Workflow of the variety and exposure adjustment to reduce the capturing effort of the training dataset for instance segmentation.

Four setups were tested for the initial variety transfer study. SDN_R training used exclusively red cabbage images, SDN_W used exclusively white cabbage images, and MDN_RW combined equal shares of both. SDN CG_W matches SDN_W except that CG (CG_R2W) converted red test images into white cabbage images. **Table 12** summarises transfer results from red to white cabbage under a single exposure condition.

Table 12. mAP values of the variety transfer of red to white cabbage in a single exposure scenario.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

Dataset: 600	Red cabbage (mAP)	White cabbage (mAP)
SDN_R	0.492	**0**
SDN_W	**0**	0.639
MDN_RW	0.375	0.626
SDNCG_W	0.51	0.639

Table 12 indicates greater difficulty for Mask R-CNN in handling red cabbages, with white cabbages achieving 29.8% higher accuracy at equal training volumes. Additionally, neither SDN_R nor SDN_W could detect the alternate variety. Halving the data to 300 images per domain in MDN_RW lowered red cabbage accuracy from 0.492 to 0.375 (23.7% reduction)—substantially more than white cabbage, which declined from 0.639 to 0.626 (2%). Converting red images to white via CG_R2W produced marginally superior accuracy on the white-trained SDN CG_W than the red-only SDN_R. Stronger head-leaf contrast in white cabbage likely contributes to this. Such conversion enables 50% savings in labelling, as SDN CG_W achieves comparable accuracy to SDN_R and SDN_W with half the labelled volume. **Figure 16** demonstrates the transfer of variety from red to white cabbage.

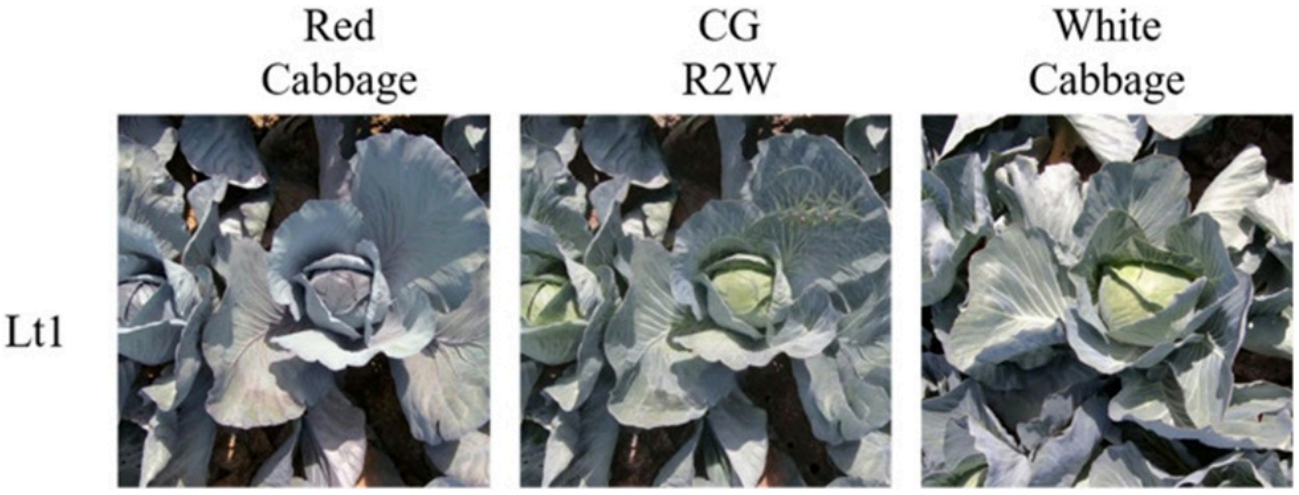


Figure 16. Variety transfer from red cabbage to white cabbage; CG_R2W is the adapted red cabbage image.

To minimise both labelling and recording burdens, exposure adaptation advantages from white cabbage are applied to red cabbage next. Six dataset versions were prepared: three red cabbage sets under Lt1, Lt2, and Lt3 exposures and three matching white cabbage sets **(Figure 5)**.

The aim is to leverage the pre-trained exposure CG (CG_L) for white cabbage and variety CG (CG_R2W) on the unfamiliar red cabbage Lt2 and Lt3 combinations. Initially, CG_R2W shifts red images (Lt1-Lt3) to white cabbage images, irrespective of exposure, trained on source red Lt1 and target white Lt1-Lt3. Only a variety shift is anticipated. Subsequently, CG_L, trained on white Lt1 handles, performs exposure correction. **Figure 17** outlines the steps: first, producing heterogeneously exposed white images via CG_R2W (CG White_b); then, exposure correction via CG_L.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

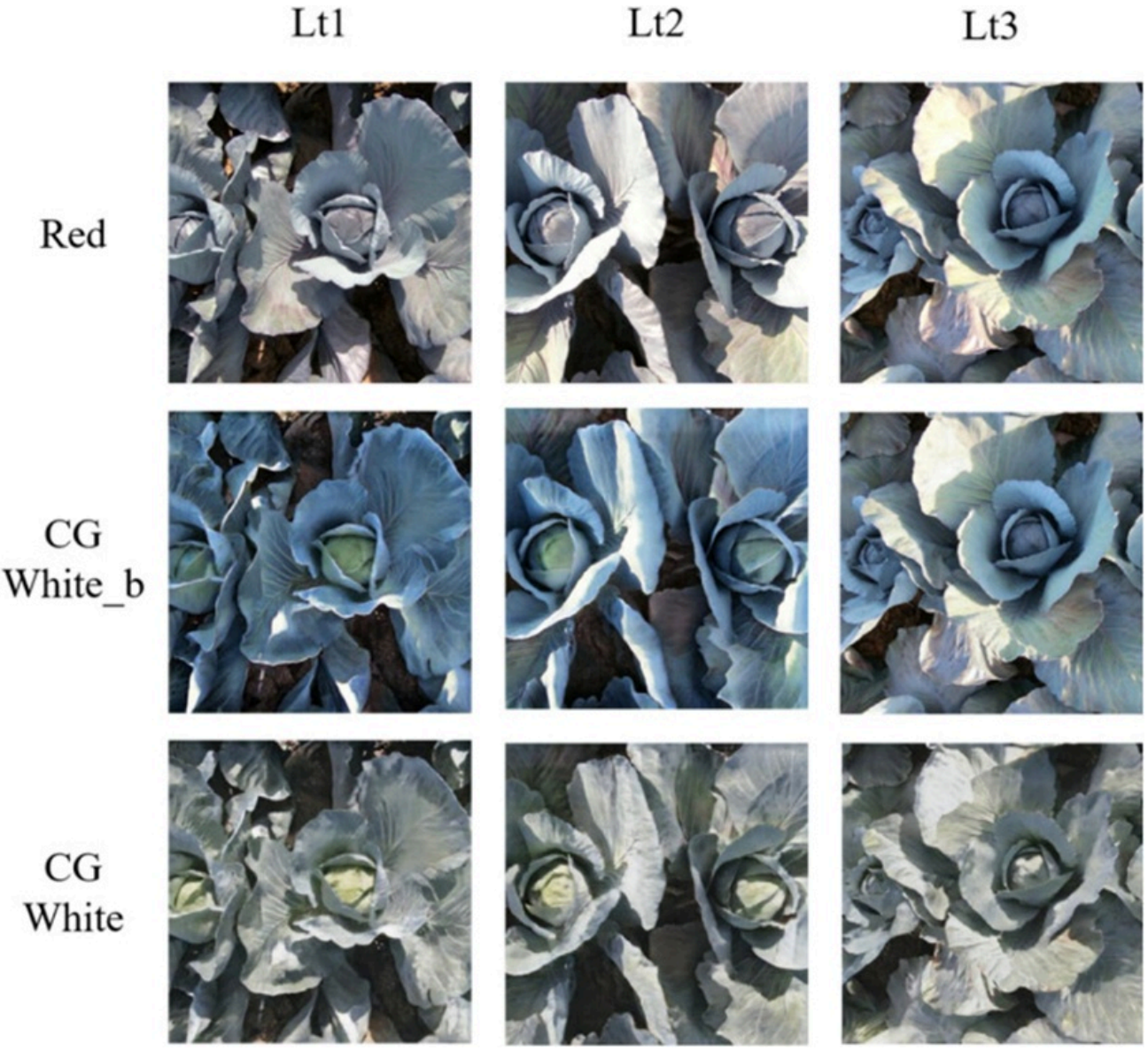


Figure 17. Exposure and variety transfer of three different exposure variations of red cabbage to one exposure variation of white cabbage.

Table 13 presents transfer outcomes. MDN_R functions as a reference across Lt1-Lt3 with mixed red cabbage. SDN CG_W trains solely on white Lt1 data. Variety plus exposure adjustment on Lt1 improves segmentation from 0.499 to 0.548 mAP (9.8% increase), as predicted given the Lt1 base. Lt2 and Lt3 variants show an average 44.8 % mAP decrease versus MDN_R. Transferring unknown exposures Lt2 and Lt3 evidently generated unfamiliar exposure-variety pairings, impairing the quality of the secondary transformation.

Table 13. mAP values for variety and exposure adaptation by the CG.

Dataset:600	**Red_Lt1 (mAP)**	**Red_Lt2 (mAP)**	**Red_Lt3 (mAP)**
MDN_R	0.499	0.544	0.38
SDNCG_W	0.548	**0.383**	**0.127**

© 2024 The Author(s). | Published by Agro Intelligence Research Press

Figure 18 displays LAB(L) distributions for raw red cabbage images (a) and transformed white cabbage images (b), with standard deviations detailed in **Table 14**.

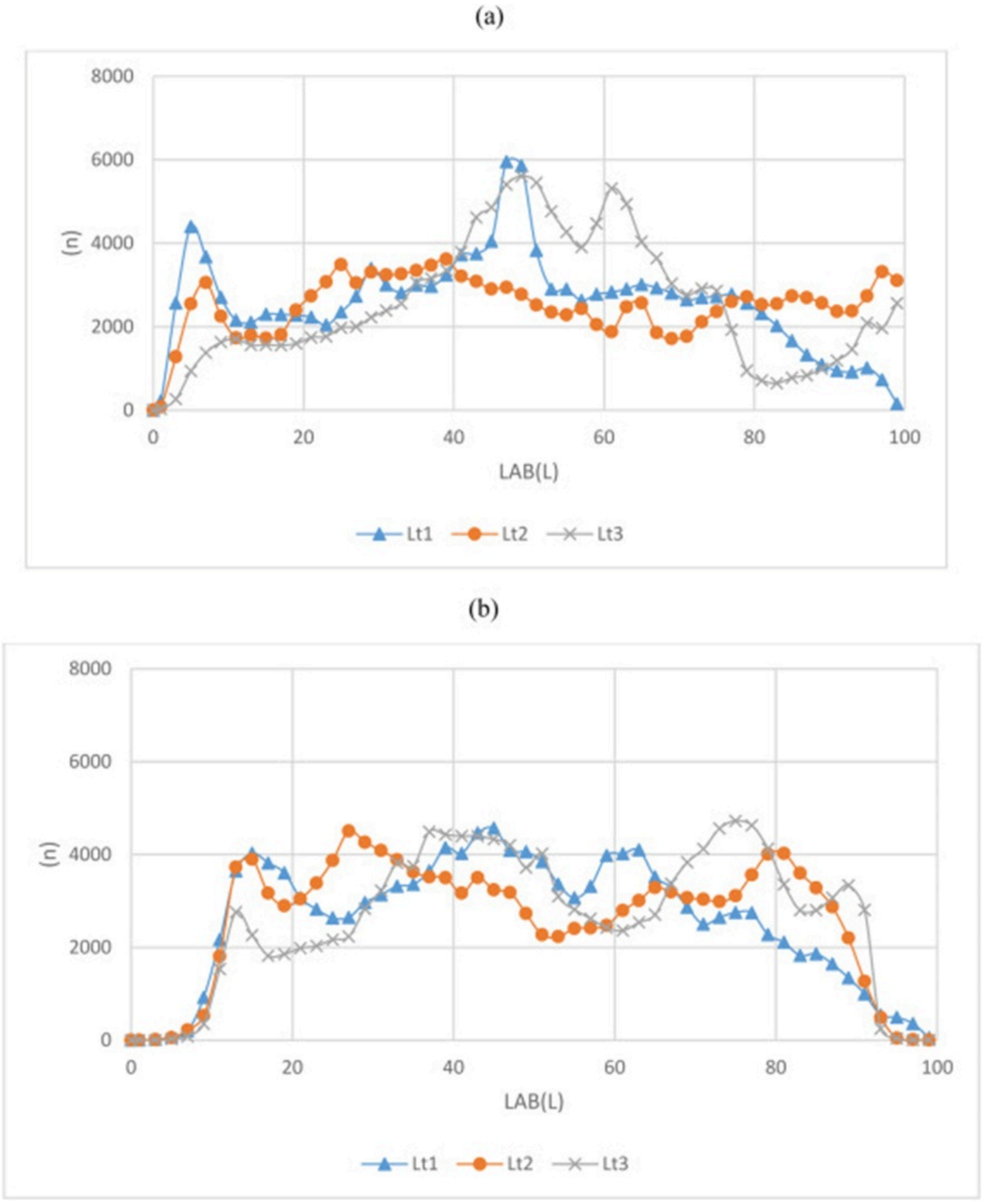


Figure 18. LAB(L) colour channel comparison of the different exposure scenarios of red cabbage (a) and transformed white cabbage images (b).

© 2024 The Author(s). | Published by Agro Intelligence Research Press

Table 14. Standard deviation (std) values of the pixel count of LAB(L) ranges of the unprocessed and adapted images from datasets Lt1 to Lt3.

Pixel count	LAB(L) 0–10	LAB(L) 90–100	LAB(L) 0–100
Unprocessed images (std)	793	920	669
Adapted images (std)	86	255	483

Figure 18b demonstrates alignment of Lt2 and Lt3 to Lt1 following red-to-white conversion by CG_R2W and CG_L. A notable correction occurs in the LAB(L) ranges 0–10 and 90–100. **Table 14** indicates pixel count standard deviation reduced from 793 to 86 in 0–10 and from 920 to 255 in 90–100 via the combined CGs. Overall standard deviation fell from 669 to 483. Relative to exposure-only correction without prior variety shift, this achieves only a 27.8% reduction, versus 54.1% with prior variety shift, clarifying the reduced segmentation quality.

Conclusion

An unsupervised image-to-image translation network was used to incorporate diverse exposure conditions and colour variations into an existing training dataset intended for instance segmentation. Two separate analyses were conducted. In the first analysis, the effects of five distinct exposure variants were examined, along with the impact of exposure modification via a CycleGAN (CG) across varying dataset sizes. Results indicated that modifying the exposure can lower the annotation workload by as much as a factor of five relative to a mixed dataset, due to the reduced generation of feature variance.

The second analysis examined how well the exposure adjustment could be transferred to alternative exposure settings, different plant growth stages, and a different cabbage variety. When adapting the image domain to previously unseen exposure variations, the CG-based method achieved a 52.1% improvement in segmentation accuracy over a network trained directly on multiple exposure scenarios. This highlighted the CG approach's stronger generalization capability under unknown exposure conditions, outperforming instance segmentation networks trained on heterogeneous data.

Extending the exposure adjustment technique across several growth stages—while training both the CG and the instance segmentation network on only a single growth stage—resulted in segmentation accuracies that were, to a certain degree, similar to those achieved by a network trained on multiple growth stages. This outcome underlined the method's transferability and its ability to reduce data-collection requirements. Subsequent investigation into the smallest viable image dataset needed to cover a full growing season could yield valuable insights into the overall data volume and time required for acquisition and processing, thereby supporting practical and reliable neural network applications in agriculture.

Two analyses were performed to determine transferability to another cabbage cultivar. In the first, the transfer from red cabbage to white cabbage was investigated. It was shown that higher segmentation precision could be obtained by converting red cabbage images to the white cabbage domain using the CG. In the second analysis, a transfer-learning strategy was implemented, aiming to apply knowledge from exposure adjustment on white cabbage images to data that had first been transferred from red cabbage to white cabbage, thereby minimizing the need to capture varied exposure conditions for a new cultivar. However, adequate accuracy was not reached for unknown exposure scenarios within this variety transfer setup.

These findings illustrate multiple ways in which unsupervised, unpaired image-to-image transfer networks can reduce the effort involved in data labelling and acquisition, potentially serving as an alternative or supplement to current techniques such as transfer learning. For future work, especially regarding plant segmentation across different growth stages, more

© 2024 The Author(s). | Published by Agro Intelligence Research Press

experiments are recommended that use varied distributions of growth stages within the training dataset to maintain stable segmentation performance throughout a complete growing season.

Acknowledgements

The project DiWenkLa (Digital Value Chains for a Sustainable Small-Scale Agriculture) is supported by funds of the Federal Ministry of Food and Agriculture (BMEL) based on a decision of the Parliament of the Federal Republic of Germany via the Federal Office for Agriculture and Food (BLE) under the innovation support programme (Grant reference 28DE106A18 | 28DE106A22). DiWenkLa is also supported by the Ministry for Food, Rural Areas and Consumer Protection Baden-Württemberg.

Conflict of interest

None

Financial support

None

Ethics statement

None

Received: 12 Mar 2023 Revised: 17 May 2023 Accepted: 01 Jun 2023
Published online: 25 November 2024

Rights and permissions

Open Access The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at https://creativecommons.org/licenses/by-nc-sa/4.0/.

References

1. Lüling N, Reiser D, Straub J, Stana A, Griepentrog HW. Fruit volume and leaf-area determination of cabbage by a neural-network-based instance segmentation for different growth stages. Sensors. 2023;23(1):129. https://doi.org/10.3390/s23010129.

2. Dyrmann M, Mortensen AK, Midtiby HS, Jorgensen RN. Pixel-wise classification of weeds and crop in images by using a fully convolutional neural network. Proceedings of the International Conference on Agricultural Engineering, Aarhus, Denmark. 2016. p. 26-9.

3. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial nets. NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems. 2014;2:2672-80.

4. Zhu J, Park T, Isola P, Efros A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy. 2017. https://doi.org/10.1109/ICCV.2017.244.

5. He K, Gkioxari G, Dollár P, Girshick R. Mask R-CNN. IEEE Trans Pattern Anal Mach Intell. 2020;42(2):386-97. https://doi.org/10.1109/TPAMI.2018.2844175.

6. Kusumam K, Krajník T, Pearson S, Cielniak G, Duckett T. Can you pick a broccoli? 3D-vision based detection and localisation of broccoli heads in the field. Proceedings of the International Conference on Intelligent Robots and Systems (IROS). 2016. p. 646-51. https://doi.org/10.1109/IROS.2016.7759121.

7. Birrell S, Hughes J, Cai JY, Iida F. A field-tested robotic harvesting system for iceberg lettuce. J Field Robot. 2019;37(2):225-45. https://doi.org/10.1002/rob.21888.

8. Lüling N, Reiser D, Stana A, Griepentrog HW. Using depth information and colour space variations for improving outdoor robustness for instance segmentation of cabbage. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). 2021. p. 2331-6. https://doi.org/10.1109/ICRA48506.2021.9561792.

9. Osorio K, Puerto A, Pedraza C, Jamaica D, Rodríguez L. A deep learning approach for weed detection in lettuce crops using multispectral images. AgriEngineering. 2020;2(3):471-88. https://doi.org/10.3390/agriengineering2030032.

10. Shim I, Lee JY, Kweon IS. Auto-adjusting camera exposure for outdoor robotics using gradient information. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Chicago, IL, USA. 2014:1011-7. https://doi.org/10.1109/IROS.2014.6942682.

11. Motamed S, Rogalla P, Khalvati F. Data augmentation using Generative Adversarial Networks (GANs) for GAN-based detection of pneumonia and COVID-19 in chest X-ray images. Inform Med Unlocked. 2021;27:100779. https://doi.org/10.1016/j.imu.2021.100779.

12. Xu M, Yoon S, Fuentes A, Park DS. A comprehensive survey of image augmentation techniques for deep learning. Pattern Recognit. 2023;137:109347. https://doi.org/10.1016/j.patcog.2023.109347.

13. Wu X, Manton J, Aickelin U, Zhu J. Online transfer learning: Negative transfer and effect of prior knowledge. Proceedings of the IEEE International Symposium on Information Theory (ISIT), Melbourne, Australia. 2021. p. 1540-45. https://doi.org/10.1109/ISIT45174.2021.9517974.

14. Kirillov A, Mintun E, Ravi N, Mao H, Rolland C, Gustafson L, et al. Segment Anything. Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France. 2023:3992-4003. https://doi.org/10.1109/ICCV51070.2023.00371.

15. Du R, Li W, Chen S, Li C, Zhang Y. Unpaired underwater image enhancement based on CycleGAN. Information. 2022;13(1):1. https://doi.org/10.3390/info13010001.

16. Meier U. Entwicklungsstadien Mono- und Dikotyler Pflanzen – BBCH Monografie (Developmental Stages of Mono- and Dicotyledonous Plants – BBCH Monograph). Biologische Bundesanstalt für Land- und Forstwirtschaft; 2001.

17. COCO. Detection evaluation. https://cocodataset.org/#detection-eval (2019) (last accessed 29.02.2024).

© 2024 The Author(s). | Published by Agro Intelligence Research Press

18. Nematzadeh Z, Ibrahim R, Selamat A. Comparative studies on breast cancer classifications with k-fold cross validations using machine learning techniques. Proceedings of the 2015 10th Asian Control Conference (ASCC), IEEE. 2015. p. 1-6. https://doi.org/10.1109/ASCC.2015.7244654.

19. Abdulla W. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. GitHub repository. 2017. https://github.com/matterport/Mask_RCNN (last accessed 29.02.2024).

20. Creswell A, White T, Dumoulin V, Arulkumaran K, Sengupta B, Bharath AA. Generative adversarial networks: an overview. IEEE Signal Process Mag. 2018;35(1):53-65. https://doi.org/10.1109/MSP.2017.2765202.

21. Schönfeld E, Schiele B, Khoreva A. A U-Net based discriminator for generative adversarial networks. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020:8204-13. https://doi.org/10.1109/CVPR42600.2020.00823.

22. Tensorflow CycleGAN. https://www.tensorflow.org/tutorials/generative/cyclegan (last accessed 29.02.2024).

© 2024 The Author(s). | Published by Agro Intelligence Research Press