ORIGINAL RESEARCH
published: 22 February 2021
doi: 10.3389/fpls.2021.591333

DAM: Hierarchical Adaptive Feature Selection Using Convolution Encoder Decoder Network for Strawberry Segmentation

*Talha Ilyas[1], Muhammad Umraiz[1], Abbas Khan[1] and Hyongsuk Kim[1,2]**

[1] Division of Electronic Engineering, Intelligent Robots Research Center, Jeonbuk National University, Jeonju, South Korea,
[2] Division of Electronic and Information Engineering, Jeonbuk National University, Jeonju, South Korea

OPEN ACCESS

Edited by:
Yiannis Ampatzidis,
University of Florida, United States

Reviewed by:
Thiago Teixeira Santos,
Brazilian Agricultural Research Corporation (EMBRAPA), Brazil
Lucas Costa,
University of Florida, United States

****Correspondence:***
Hyongsuk Kim
hskim@jbnu.ac.kr

Specialty section:
This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science

Received: *04 August 2020*
Accepted: *25 January 2021*
Published: *22 February 2021*

Citation:
Ilyas T, Umraiz M, Khan A and Kim H (2021) DAM: Hierarchical Adaptive Feature Selection Using Convolution Encoder Decoder Network for Strawberry Segmentation. Front. Plant Sci. 12:591333. doi: 10.3389/fpls.2021.591333

Autonomous harvesters can be used for the timely cultivation of high-value crops such as strawberries, where the robots have the capability to identify ripe and unripe crops. However, the real-time segmentation of strawberries in an unbridled farming environment is a challenging task due to fruit occlusion by multiple trusses, stems, and leaves. In this work, we propose a possible solution by constructing a dynamic feature selection mechanism for convolutional neural networks (CNN). The proposed building block namely a dense attention module (DAM) controls the flow of information between the convolutional encoder and decoder. DAM enables hierarchical adaptive feature fusion by exploiting both inter-channel and intra-channel relationships and can be easily integrated into any existing CNN to obtain category-specific feature maps. We validate our attention module through extensive ablation experiments. In addition, a dataset is collected from different strawberry farms and divided into four classes corresponding to different maturity levels of fruits and one is devoted to background. Quantitative analysis of the proposed method showed a 4.1% and 2.32% increase in mean intersection over union, over existing state-of-the-art semantic segmentation models and other attention modules respectively, while simultaneously retaining a processing speed of 53 frames per second.

Keywords: semantic segmentation, convolutional neural network, encoder-decoder architecture, fruit segmentation, channel attention, spatial attention, segmentation grad-cam, autonomous harvesting

INTRODUCTION

Since the evolution of deep convolutional neural networks (DCNNs) from neural networks (Krizhevsky et al., 2012), machine learning has shown unprecedented performance on a number of machine vision and pattern recognition tasks such as image classification (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; He et al., 2016; Szegedy et al., 2016; ur Rehman et al., 2018), object detection and localization (Ren et al., 2015; Redmon et al., 2016; He et al., 2017; Nizami et al., 2020), and semantic and instance segmentation (Long et al., 2015; Ronneberger et al., 2015; Badrinarayanan et al., 2017; Bolya et al., 2019). Recently, unsupervised algorithms are also gaining popularity (Epifanio and Soille, 2007; Zhao and Kit, 2011; Xia and Kulis, 2017; Ilyas et al., 2020) due to their certain advantages over supervised ones (Huang et al., 2017; Lin et al., 2017c;

Ilyas et al., 2020). Moreover, deep learning has also demonstrated unparalleled performance in the field of bioinformatics and computational biology (Wahab et al., 2019, 2020; Park et al., 2020).

Where DCNNs have found several intuitive applications in various fields in our everyday lives, they are also being used in agriculture for autonomous harvesting and seeding. A lot of work has been done in literature in this regard, like crop and weed classification (Dyrmann et al., 2016, 2017; Grinblat et al., 2016; Kussul et al., 2017), plant detection (Mohanty et al., 2016; Khan et al., 2020), land cover classification (Ienco et al., 2017; Kussul et al., 2017), and crop disease identification (Fuentes et al., 2018). Just like any other machine vision task, the implementation of DCNNs in agriculture comes with its own set of problems. By the same token, the real time segmentation and detection of strawberries in an unconstrained farm environment is a challenging task, as strawberries usually grow in clusters and are occluded by leaves, branches, and other fruits. Due to different light intensities sometimes backgrounds and fruits have the same texture and color. These commonly occurring phenomena in farms makes the task more difficult and reduces the accuracy of DCNNs (Sa et al., 2016; Xiong et al., 2019).

Strawberries are some of the most highly valued crops as they give the best yield under sheltered environments, and thus have a very high production cost (Sa et al., 2016). The most crucial time for strawberry crop is harvesting time because the fruit becomes overripe quickly and if picking gets behind it effects the whole crop. Moreover, hiring skilled laborers in horticulture accounts for most of the cultivation cost. This crop also needs intensive post-harvest care (Guerrero et al., 2017). Because of all these expenses, horticulture industries in general are bound to have small profit margins. In some regions, labor cost makes up more than half of the total production cost, e.g., 60% in Norway (Xiong et al., 2019). Furthermore, there is a decline in interest of joining the agriculture industry among the new generation of workers (Adhikari et al., 2019). Under all these challenges the food industry must keep up with the demands of the ever-growing population.

To overcome such problems, one potential solution is autonomous harvesting as it can reduce labor cost to a minimum and increase the crop yield quality by timely harvesting. Due to outstanding performances of DCNNs in computer vision tasks, robotics and unmanned systems are now faster and more reliable than ever. Which in turn has allowed their adoption into many real-life applications like the detection of crop rows, weeds, and seeding beds in fields of maize and rice (Guerrero et al., 2017; Adhikari et al., 2019; Ma et al., 2019).

In this work, we proposed a DCNN named Straw-Net, to precisely segment and classify the fruits into specified classes in real time. In the case of strawberries, this is difficult to achieve because they usually grow in clusters and within the same cluster, and tend to have different sizes, shapes, and colors. In some cases, severe occlusion may also occur which renders the fruit almost invisible. By taking all these shortcomings into account, we designed an adaptive self-contained attention mechanism (i.e., dense attention module, DAM) for our network, which is capable of learning both channel and spatial interdependencies and can learn ‘what’ is important and ‘where’ to put more focus. We verify the efficacy of the proposed attention module quantitatively via benchmark metrics and qualitatively via modified Grad-CAM (Selvaraju et al., 2017). Grad-CAM is usually used for classification models, however, this paper extends its applicability to segmentation models. The dataset used in this paper is collected from different strawberry farms across the Republic of Korea under different lighting and weather conditions for better generalization of real-life scenarios. Our main contributions are listed below:

- We propose a single attention module (DAM) for both channel and spatial attention, and a parallel dilated convolution module (PDC) for aggregating multi-scale context.
- We validate the effectiveness of DAM and PDC by ample ablation experiments.
- We propose an optimal location for integrating our attention module in any existing network and compare results with other existing attention mechanisms.
- We propose a technique for visual interpretation of segmentation networks by modifying Grad-CAM.
- A new dataset for the semantic segmentation of strawberries is introduced, consisting of four classes depending on the ripeness level of fruit as shown in **Figures 1**, **2** (see section “Materials and Methods” for details).

RELATED WORK

Autonomous Harvesting

A great deal of work has been done in the field of autonomous harvesting of fruits using various classical image analysis and pattern recognition techniques (Yamamoto et al., 2010; Feng et al., 2012; Ouyang et al., 2012; Qingchun et al., 2012; Zhang et al., 2012; Hayashi et al., 2013, 2014). One autonomous strawberry harvester (ASH) in particular AGROBOT SW6010 (Agrobot, 2020) has gained a lot of popularity. It uses morphological, color, and shape analysis for the identification and selection of strawberries to harvest, and uses 24 robotic arms to perform harvesting. FFRobotics (Kahani, 2017) have introduced the FFBot for harvesting apples, which needs a human supervisor to control and monitor the harvesting process. More recently, Hravest CROO (Harvest Croo, 2021) has introduced an ASH which divides the fruit picking into three steps, (a) grab the leaves, (b) 3D inspection of plant, and finally (c) pick the fruit. This simple robotic framework allows them to increase harvesting speed. Autonomous harvesters are also gaining popularity in other areas of precision agriculture like weeding. NAIO Technologies (Barthes, 2010) has introduced multiple autonomous weeding robots like OZ, TED, and DINO for easier vegetable weeding on large-scale farms.

The performance and accuracy of any autonomous harvester relies heavily on the perception system used and how the visual information obtained is being processed. An earlier era of vision-based harvesters used monocular devices to obtain 2D visual

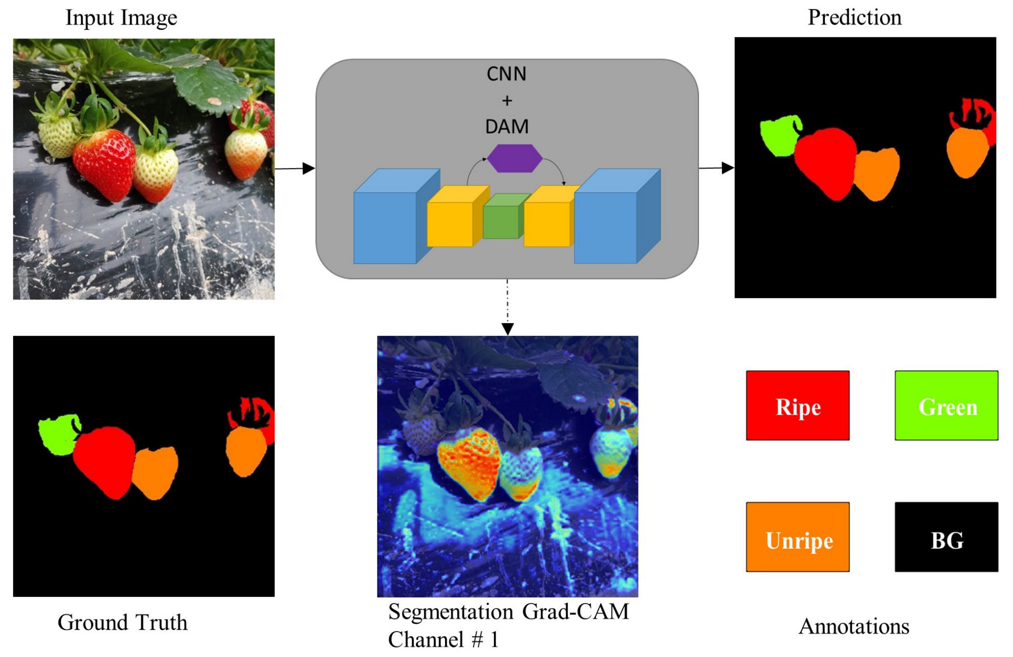

FIGURE 1 | Overview of the proposed DCNN and the dataset with corresponding annotations. Seg-Grad-CAM represents the attention maps for ripe class (i.e., channel # 1).

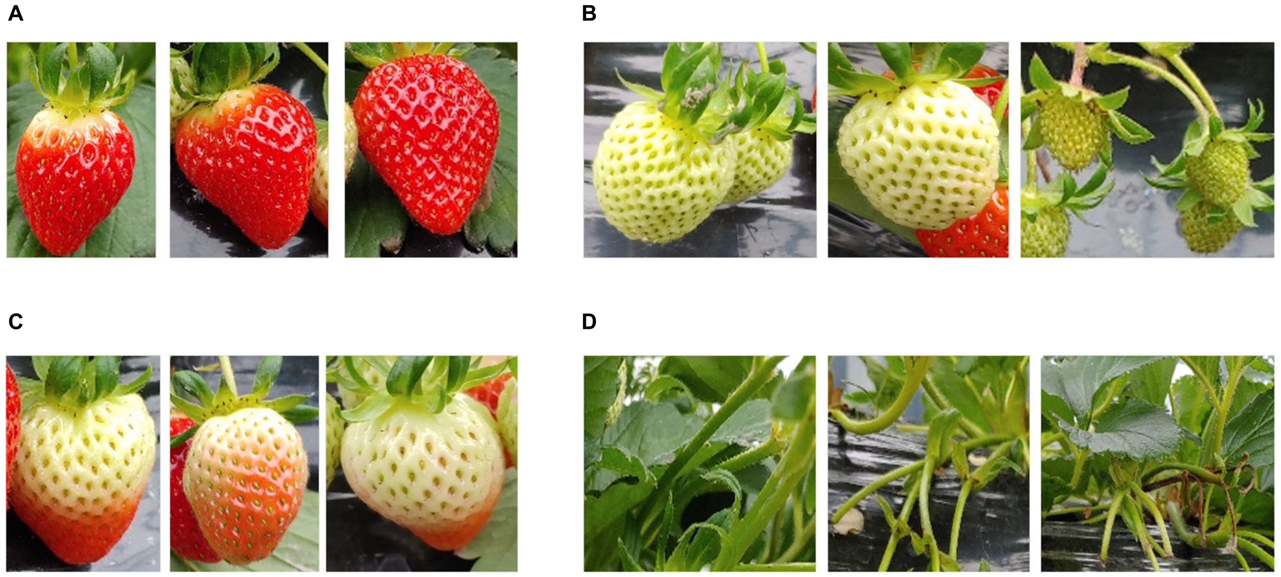

FIGURE 2 | A few representative instances belonging to each class (stage of ripeness) in the strawberry segmentation (SS1K) dataset. **(A)** Ripe; edible quality, **(B)** green; not ready for harvest, **(C)** unripe; export quality, **(D)** background.

data, e.g., Grand d'Esnon et al. (1987) classified fruits based on their texture and geometry and Edan et al. (2000) used two monochromatic cameras to produce a stereo-like effect for better localization of melons. Similarly, Yamamoto et al. (2014) introduced a stationary strawberry picking mechanism which used three different monochromatic light sources for the coloration measurement and spectral analysis of fruits and leaves to precisely localize the fruit for picking. With this mechanism they achieved an effective yield rate of 67%. But these monochromatic vision systems were highly susceptible to light intensity changes. Later, some works used stereovision to obtain 3D map of fruits via triangulation (Sun et al., 2011). A predecessor of AGROBOT SW6010 used stereo RGB-D images for tomato harvesting (Buemi, 1995; Buemi et al., 1996). Using

RGB-D images obtained via a Binocular-stereo vision camera, Ge et al. (2019) constructed a 3D point cloud to localize the pickable fruit. They used a Mask-RCNN as a backbone of their computer vision-based control to classify the strawberry into two classes, i.e., ripe and unripe. Following this pipeline they were able to improve the picking accuracy to 74%. Similarly, for recognizing clustered tomatoes and classifying them into overlapping and adhering regions, Xiang et al. (2014) used stereovision to obtain a depth-map, the reported accuracy for clustered tomato detection was 87.9%. Sensor calibration plays a vital role in the performance of stereovision systems. Recently, laser-based distance measuring systems (LiDAR) and spectral imaging are also doing wonders in precision agriculture. Zhang et al. (2015) combined computer vision with near-infrared structured lighting, and using a single multispectral camera was able to reconstruct the 3D surface of the apple for calyx and stem recognition. The results showed a 97.5% average accuracy.

These aforementioned ASHs rely heavily on classical mathematical algorithms; Ouyang et al. (2012) introduced a pipeline consisting of a series of image preprocessing and denoising techniques and then used the optimal thresholding (Otsu) algorithm for strawberry segmentation. Zhao et al. (2016) used a grayscale co-occurrence matrix to extract features of fruits from various color spaces. Whereas, different color spaces have different properties and their own application domain. Wei et al. (2014) used an OHTA color space, a modified version of the Otsu algorithm, to achieve an impressive detection accuracy of more than 95%. But it had a major drawback of not being able to detect green-colored fruits. Similarly, Qingchun et al. (2012) used a HSV color space to extract features for strawberry harvesting, results showed a 86% successful harvest rate. All the above-mentioned methods can detect fruits under controlled environments, but the detection accuracy drops when illumination changes. Moreover, depending on the variation in orientation, size, and shape of fruit, these methods require a lot of parameter tuning (Durand-Petiteville et al., 2017). In short, although several of the ASH prototypes have been developed to segment and classify fruits accurately in real time, their performance remains susceptible to unconstrained environments. This is where machine learning comes in.

Agricultural farms are unconstrained natural environments or semi-constrained at very best. Machine learning has found intuitive applications in many fields, because of its adaptive learning ability, like in healthcare (Ronneberger et al., 2015; Işın et al., 2016; Kauanova et al., 2017), autonomous driving (Fujiyoshi et al., 2019; Hofmarcher et al., 2019; Imai, 2019), and weed and crop detection (Grinblat et al., 2016; Mohanty et al., 2016; Dyrmann et al., 2017; Kussul et al., 2017; Fuentes et al., 2018). But very little work has been done in detecting fruits and classifying them according to their ripeness level. Lamb and Chuah (2018) used a single-stage detector SSD (Liu et al., 2016) to detect strawberries and attained a maximum average precision of 87.7%, but were not able to achieve real-time performance (see section "Real-Time Performance Barrier") even after using various network compression techniques. Bargoti and Underwood (2017) proposed an image processing framework using a simple CNN and a multi-scale multi-layer perceptron (ms-MLP) to detect and count apples, with an F1-score of 85.8%. Their algorithm was a multi-stage setup which used watershed and circular Hough transform to detect the individual fruits. Hence falling short of real-time performance. Chen et al. (2019) used a faster region-based convolutional neural network (F-RCNN) (Ren et al., 2015) for predicting strawberry production rate using aerial farm images. Sa et al. (2016) presented an approach for fruit detection in field farms using an F-RCNN and showed its generalization to many different farm fields. Moreover, Yu et al. (2019) combined a Mask-RCNN with a feature pyramid network (FPN) for better feature extraction, to detect mature strawberries (one class) with a precision rate of 95%, but were not able to break the real-time performance barrier (see section "Real-Time Performance Barrier"). Whereas our proposed encoder-decoder based CNN is able to explicitly detect and classify fruits according to specified ripeness levels while still maintaining a processing speed of 53 fps on standard resolution images.

Semantic Segmentation

Since the dawn of fully convolutional networks (FCNs) (Long et al., 2015) semantic segmentation has gained a lot of popularity. Following the main idea of embedding low contextual information in a progressive manner to preserve spatial and temporal information, a lot of encoder-decoder architecture has been introduced in literature. Deconv-Nets (Noh et al., 2015) introduced transposed convolution called deconvolution, for learning the upsampling process. SegNets (Badrinarayanan et al., 2017) introduced unpooling (i.e., inverse of pooling) to upsample the score maps in a gradual way. To remedy the loss of localization information by the subsequent downsampling of feature maps, U-net (Ronneberger et al., 2015) proposed skip-connections between the encoder and decoder to preserve spatial information. Further, the intermediate layers were exploited by RefineNet (Lin et al., 2017a) with skip-connections, which uses multipath refinement via different convolutional modules to get final predictions. Global Convolutional Network (Peng et al., 2017) tried to increase the receptive field by factorizing large kernels into smaller ones to get global contextual embeddings. PSP-Net (Zhao et al., 2017) used spatial pyramid pooling at different scales, and Deeplab (Chen et al., 2017) used atrous convolutions with different dilation rates for exploiting multi-scale information. Contrary to previous works that exploited intermediate layers by modifying identity skip-connections (Lin et al., 2017a; Peng et al., 2017) and those that use contextual multi-scale embedding for context gathering, our proposed network integrates the representational power of both of these types of networks to achieve better segmentation results.

Dilated Separable Convolution

More recently, networks like Dilated ResNet (DRN) (Yu et al., 2017) used dilated convolutions (Yu and Koltun, 2015) to increase the valid receptive field size while still maintaining the same computational cost (i.e., number of parameters and FLOPs). Furthermore, Deeplab-v3+ (Chen et al., 2018) combined dilated convolution with depth-wise separable convolution (Chollet, 2017). By doing so they achieved a significant performance boost while keeping the model complexity to a minimum. These convolutions have been adopted by many recent algorithms (Jin et al., 2014; Wang et al., 2016; Howard et al., 2017; Zhang et al., 2018). In our network, we have also used the dilated separable convolution for better performance.

Attention Mechanism

Attention plays a vital role in human perception (Rensink, 2000; Corbetta and Shulman, 2002). As a matter of fact, neurons present in the primary visual cortex of cats (Hubel and Wiesel, 1962) have inspired the construction of DCNNs (LeCun et al., 1989). Neurons in the human visual system do not process the whole semantic scene at once. Instead the neurons try to process the scenery in a sequence and they adaptively focus on only the salient features of the scenery in front of them (Woo et al., 2018).

Recent algorithms have also tried to equip DCNNs with such attention mechanisms to improve their performance (Lin et al., 2017b; Shen et al., 2018). More recently, Fu et al. (2019) proposed a self-attention mechanism for integrating local and global semantic features. Their mechanism consisted of two modules, one for position attention (PAM) and one for channel attention (CAM). Because of heavy matrix multiplications, both modules were far too computationally expensive. Whereas, squeeze and excite (SE) networks (Hu et al., 2018) recalibrated the feature maps depending upon their importance, while keeping the computational overhead to a minimum. Although in Hu et al. (2018), the authors implicitly refer to the SE module as an attention mechanism, this can be explicitly considered as one, as shown by Park et al. (2018) and Woo et al. (2018). Recently, a block attention module (BAM) (Park et al., 2018) and convolution block attention module (CBAM) (Woo et al., 2018) achieved a significant performance boost in an ImageNet-1K classification challenge by adding spatial attention to SE modules. These aforementioned modules also consisted of two separate blocks for generating channel and spatial attention. In contrast to these works, we extend the use of attention mechanisms to the segmentation task. Moreover, different from existing works, instead of using two separate blocks for channel and spatial attention, we propose one block for both tasks, to avoid computational overhead and reduce inference time. We propose a gating mechanism to control the flow of multi-scale information from different stages of the backbone network (encoder) to suitable upsampling stages of the decoder. By doing so, we are able to achieve better category specific attention masks. Our adaptive self-contained attention mechanism can learn both channel and spatial interdependencies and can dynamically emphasize or suppress features according to their importance. Detailed ablation experiments verify the effectiveness of our module (see section "Results and Discussion").

MATERIALS AND METHODS

Image Acquisition

Strawberry images were collected from several strawberry farms across Jeonju-si District, Jeollabuk-do, Republic of Korea during the growing season (2019). All the strawberry farms adopted a hedgerow planting system (Strawberry, 2020) as shown in **Figure 3**. The data acquisition was carried out at a distance of 40 cm, using a 24.1 MPx Canon EOS-200D-based platform with a CMOS sensor. We chose this distance so that the device could capture sufficiently large scenery for processing, and at this distance ASH would be able to perform suitable target searching and harvesting. During different time periods and under varying weather and lighting conditions, we acquired 1500 images. Images were stored in the JPEG format and all had a resolution of 6288 × 4056 pixels. We stored the data in high resolution to avoid being limited in available resolution at later processing stages.

Dataset Construction and Annotation

We started with primary data filtering and removed the images which were blurred or contained no strawberry fruit at all. After the primary filtering step, we ended up with 1000 unlabeled images. Out of the total 1000 unlabeled images we randomly selected 750 images for training, 100 for validation, and 150 for testing.

Then with the help of experts in the strawberry harvesting field, we divided the strawberry fruits into four classes depending on the ripeness level. We labeled them as follows: (a) ripe; (edible quality) ready for harvesting, (b) green; not ready for harvesting, (c) unripe; (export quality) that can be harvested if the farm had to export the strawberries to far away destinations, and (d) background. Some representative instances belonging to each class (stage of ripeness) are shown in **Figure 2**. According to field experts and the Food and Agriculture Organization (FAO, 2020), a strawberry which is less than 70% matured should be considered as export quality. Because any more than that and there is a chance that the fruit may rot over long journeys. So, one might say that labeling the unripe class is somewhat intuitive. After deciding the ripeness level, we labeled the images as shown in **Figure 1**. There also exists a data imbalance between the classes, such that per batch there are a greater number of ripe and green strawberries than unripe ones, as shown in **Figure 4**. We will discuss this problem of data

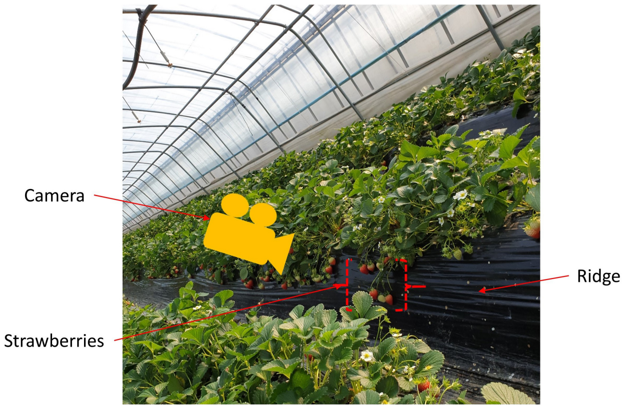

FIGURE 3 | Data acquisition process.

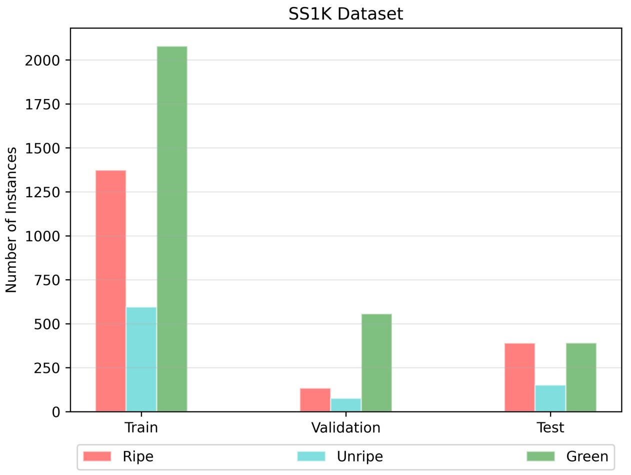

FIGURE 4 | The number of instances of each class in the strawberry segmentation (SS1K) dataset.

imbalance in the performance analysis (see section "Results and Discussion"). From this point onward for ease of notation we will call this strawberry segmentation dataset SS1K (1K for the total number of samples).

NETWORK ARCHITECTURE

First, we will describe the backbone of our architecture, i.e., the encoder part and the blocks used within, and then we will discuss the design considerations for our attention module. Finally, we will describe the decoder design choice and how to integrate the attention module in any existing network.

Encoder Design Considerations

A modified FCN for real-time segmentation of strawberry fruit, named Straw-Net is shown in **Figure 5A**. The encoder consists of SE-ResNet (Hu et al., 2018)-like blocks, with a few modifications. The SE-ResNet block consists of two parts, one being the ResNet bottleneck and the other being the SE-module as shown in **Figure 5B**. In the ResNet bottleneck, instead of using simple convolution, we decided to use the dilated separable convolution, which is a combination of dilated and depth-wise separable convolution (Chen et al., 2018). It allows the network designer to freely control the feature map's size and filter's effective receptive field (ERF), while significantly reducing the network computational cost. Depth-wise separable convolution disentangles the normal convolution into a depth-wise (or channel-wise) convolution followed by a point-wise convolution. This decomposition allows the DCNN to achieve better performance with much fewer parameters. In dilated convolutions, ERF can be easily changed by changing the dilation rate 'd_i' (Yu et al., 2017), where normal convolution is a special case of dilated convolution with $d = 1$. Increasing the ERF at each stage of the network helps the convolutional filters to aggregate multi-scale contextual information more efficiently.

After processing the features by bottleneck layers, next these feature-maps are passed through the SE-module (Hu et al., 2018), shown in **Figure 5B**. SE-modules recalibrate the feature maps by obtaining their channel-wise statistics via global average pooling (GAP). The GAP outputs a vector of size n, where n is same as the number of filter channels. Then this vector is passed through a multi-layer perceptron (MLP) to obtain a weighing vector of size n. This vector is then used to adaptively emphasize or suppress the feature maps according to their importance. For more details about SE-modules, we refer interested readers to Hu et al. (2018). Moreover, skip-connection allows for uninterrupted gradient flow to the earlier layers for better training.

Data in raw images are mostly redundant so a large kernel size with high stride can be used to process the raw image and make it ready for deeper layers to process. Using a high stride also reduces the dimensions which will in turn reduce the computational overhead (Hasanpour et al., 2016; He et al., 2016). Keeping that in mind, firstly the image is passed through a normal convolution layer with 16 filters of size 7 × 7 and stride 2. Now this processed input is passed through the successive SE-ResNet blocks as shown in **Figure 5A**. Each convolutional layer in SE-ResNet is followed by a batch normalization (BN) and ReLu activation, unless explicitly stated.

The network backbone consists of six stages. All stages consist of two SE-ResNet blocks. Among those, the first three stages $S_{i\in\ \{1,2,3\}}$ are followed by subsequent pooling operations for reducing feature map size. In the next stages $S_{i\in\ \{4,5,6\}}$ we do not perform a pooling operation. Because, after using the stride = 2 in the first layer and the three subsequent pooling operations in the first three stages, the extracted feature map size is 16 times smaller than the input at the end of the encoder. Reducing it further will result in the loss of a lot of useful localization information, making the decoding process more difficult. In the first two stages, the dilation rate is set to $d = 1$, and in the next three stages, the dilation rate is doubled for every next stage, i.e., $d_{i\in\ \{3,4,5\}} = \{2,4,8\}$ for stages $S_{i\in\ \{3,4,5\}}$. The final stage S_6 again has a dilation rate of $d_6 = 1$ to avoid the gridding artifact (Yu and Koltun, 2015).

Parallel Dilated Convolution Module (PDC)

We go deeper into the DCNNs, even though the deeper layers have a large theoretical receptive field (TRF) but their effective receptive field (ERF) is much smaller than the theoretical one as shown by Zhou et al. (2014). Information regarding global context plays a vital role in scene segmentation (Peng et al., 2017; Zhao et al., 2017). So, at the end of the encoder we probe the feature maps of the last stage (i.e., S_6) for aggregating global and sub-region context by incorporating the PDC module shown in **Figure 6B**. PDC acts as a hierarchical global module prior to using dilated convolution at different dilation rates to extract global contextual information from S_6's feature maps at multiple scales. We perform detailed ablation experiments to show the effectiveness of PDC and compare it with other multi-scale feature aggregation modules of Zhao et al. (2017) and Chen et al. (2018) in Section "Ablation Study for (PDC)."

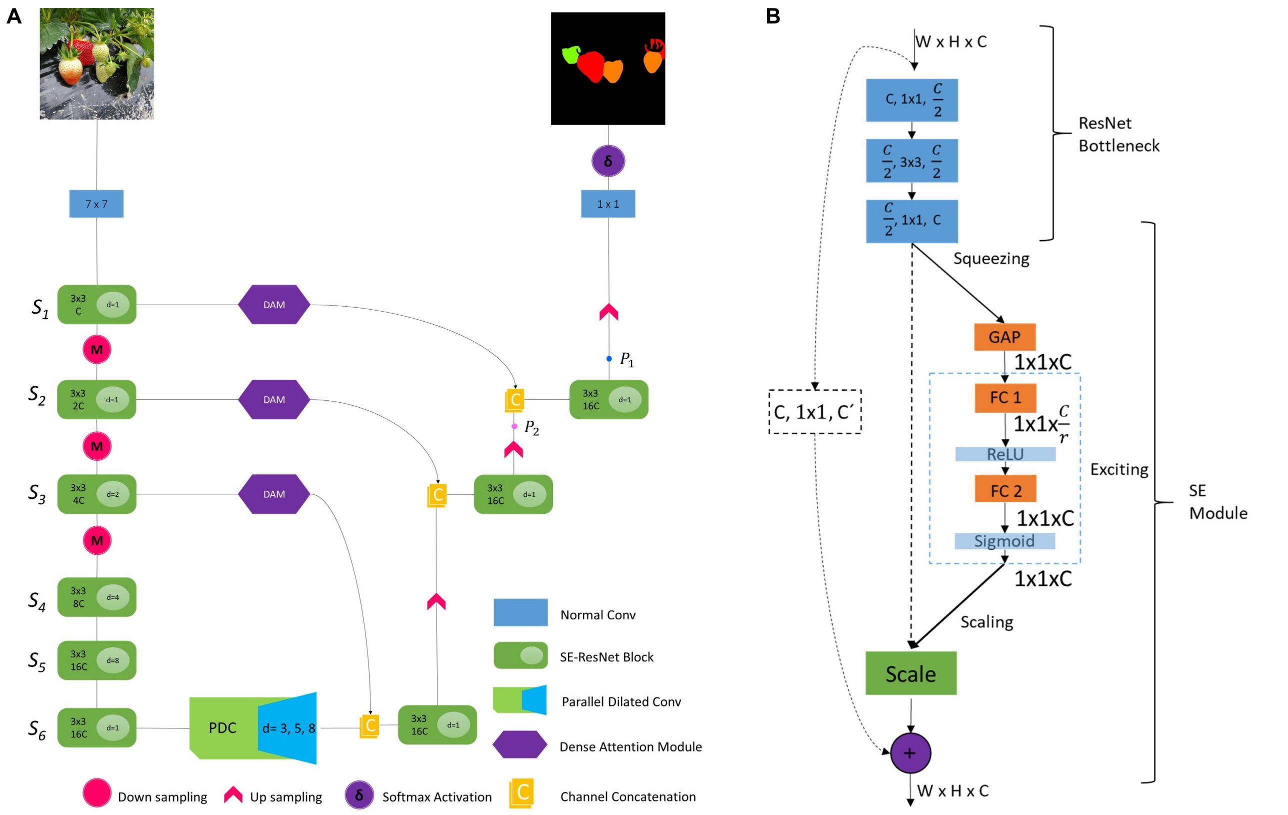

FIGURE 5 | (A) Straw-Net, complete architecture. Here S_i represents different stages of network $i \in \{1,2,\ldots 6\}$. c represents the number of channels inside each SE-ResNet block (we set $c = 16$), and d is the dilation rate. For details on points P_1 and P_2 see Section "Ablation Study for DAM." **(B)** SE-ResNet module; in the first three convolutional blocks, the first value represents the number of input feature maps, the second value represents the kernel size, and the third value represents the number of output channels. The dashed arrows represent the identity mapping. GAP, FC, and r represent global average pooling, densely connected layers, and reduction ratio, respectively.

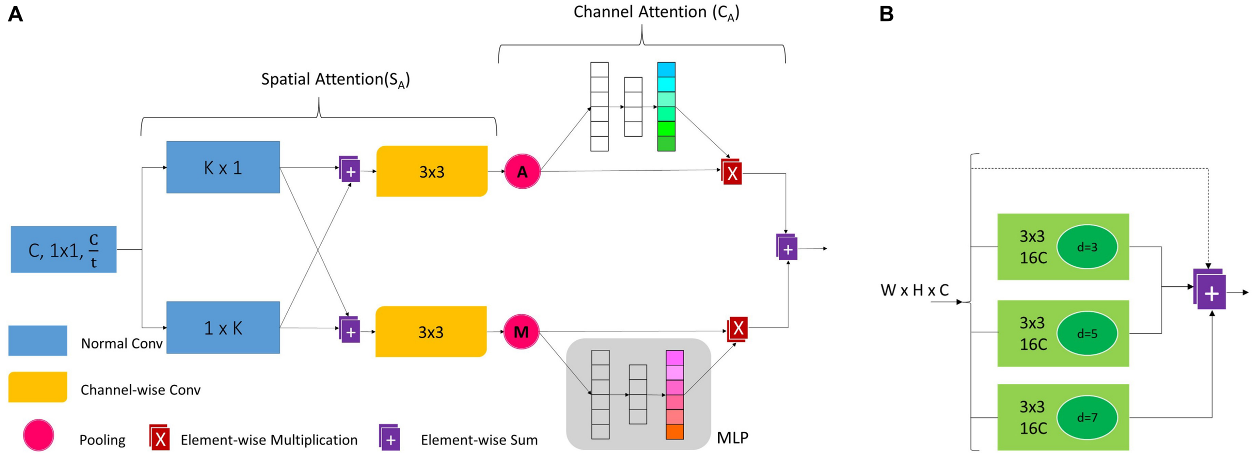

FIGURE 6 | (A) Dense attention module, here K is kernel size and t is the reduction factor. **(B)** Parallel dilated convolution module, here C represent the no. of feature maps and d is the dilation rate.

Dense Attention Module (DAM)

To control the flow of information from encoder to decoder via skip-connections we incorporate the dense attention modules on skip-connections. We found that this is the best location to make the most out of these attention modules. These modules perform *'feature surgery'* on the feature maps coming from the encoder which are rich in localization information. They help in efficient feature fusion between encoder feature maps (which focus on 'where' the target object is) and decoder feature maps (which focus on 'what' the target object is). The whole operation can be summarized as follows;

$$F^{''}_{s_i} = DAM\left(F_{s_i}\right) \tag{1}$$

Given an input feature map $F_{s_i} \in \mathbb{R}^{W_{S_i} \times H_{S_i} \times C_{S_i}}$ from stage S_i of the encoder, the DAM computes the refined feature map $F^{''}_{s_i} \in \mathbb{R}^{W^{'} \times H^{'} \times C^{'}}$ to be concatenated with decoder feature maps. Usually, the low-level feature maps have a large number of channels (e.g., 128 or 256). So, DAM first reduces the number of channels of the corresponding low-level feature maps by a factor of t such that $F_{s_i} \in \mathbb{R}^{W_{s_i} \times H_{s_i} \times C^{'}}$ where $C^{'} = \frac{C_{s_i}}{t}$. To avoid the suppression of information in rich decoder feature maps by the low-level encoder feature maps, we set $t = 4$ in our experiments. The contextual information is aggregated using a large kernel size $f_n^{K \times K}$. To reduce the number of computations and inference time we decompose the one $f_n^{K \times K}$ filter into two parallel $f_n^{1 \times K}$ and $f_n^{K \times 1}$filters. Here 'n' represents the normal convolutional filter and we set $K = 7$. Then the results of both these convolutions are added in their respective parallel branches as shown in **Figure 6A**. Next, we pass these feature maps through a depth (channel)-wise convolutional layer (in their respective branches) of filter size 3 × 3, i.e., $f_c^{3 \times 3}$ where 'c' represents the depth-wise convolutional filter. In the depth-wise convolution, one filter convolves spatially on only one feature map making the output feature maps spatially enhanced as shown by Gao et al. (2018). So, the channel specific spatial attention for both branches is computed as,

$$F^{S_A}_{s_i} = f_c^{3 \times 3}\left(f_n^{1 \times K}\left(F_{S_i}\right) + f_n^{K \times 1}\left(F_{S_i}\right)\right) \tag{2}$$

Here, the superscript S_A refers to spatial attention in the top (S_t) and bottom (S_m) branch. Next, this channel-specific spatial attention is recapitulated using both average and max pooling operations generating different feature descriptors.

$$F^{avg}_{S_i} = AvgPool\left(F^{s_t}_{s_i}\right) \tag{3}$$

$$F^{max}_{S_i} = MaxPool\left(F^{s_m}_{s_i}\right) \tag{4}$$

Here, $F^{pool}_{s_i} \in {'}\mathbb{R}^{W^{'} \times H^{'} \times C^{'}}$ where W' and H' represent the pooled (average and max) width and height of the feature maps. Unlike previous work (Hu et al., 2018), we argue that instead of using only average pooling, exploiting both pooling operations to gather distinct global characteristics helps the module to infer distinct channel-wise attention in both branches independently. Exploiting both average and max pooling features greatly improves the network's representational power (see section "Ablation Study for DAM"). After pooling, these 3D feature descriptors are passed through an MLP to obtain a 1D descriptor vector $F^{C_A}_{s_i} \in \mathbb{R}^{1 \times 1 \times C^{'}}$, for obtaining channel attention C_A for both the top (C_t) and bottom (C_m) branch. MLP consists of one GAP layer for obtaining channel-wise statistical data and two neuron layers. These 1D vectors can now be used to scale their respective 3D feature maps according to their importance. In short, the channel attention is obtained as follows;

$$F^{C_t}_{s_i} = \; MLP\left(F^{avg}_{S_i}\right) \tag{5}$$

$$F^{C_m}_{s_i} = \; MLP\left(F^{max}_{S_i}\right) \tag{6}$$

$$F^{C_t}_{s_i} = \sigma\left[W_2\left(W_1\left(GAP\left(F^{avg}_{S_i}\right)\right) + b_1\right) + b_2\right] \tag{7}$$

$$F^{C_m}_{s_i} = \sigma\left[W_2\left(W_1\left(GAP\left(F^{max}_{S_i}\right)\right) + b_1\right) + b_2\right] \tag{8}$$

Here $W_1 \in \mathbb{R}^{C^{'} \times C^{'}/r}$ and $b_1 \in \mathbb{R}^{C^{'}/r}$ are the weights and biases of the hidden neuron layer while $W_2 \in \mathbb{R}^{C^{'}/r \times C^{'}}$ and $b_2 \in \mathbb{R}^{C^{'}}$ belong to the output neuron layer. Finally, the output of the module is now calculated as;

$$F^{''}_{s_i} = \left(F^{C_t}_{s_i} \otimes F^{avg}_{S_i}\right) + \left(F^{C_m}_{s_i} \otimes F^{max}_{S_i}\right) \tag{9}$$

Where, $\bigotimes$ denotes element-wise multiplication.

Decoder Design Choices

We propose a simple yet effective decoder for our network as shown in **Figure 5A**. Our decoder bilinearly upsamples the feature map by a factor of 16 in subsequent steps. In the first step, the output of PDC is concatenated with the refined feature maps (i.e., output of DAM) of the third stage of the encoder then processed through a SE-ResNet block and finally upsampled by a factor of 2. The second and third steps also upsample the feature maps after concatenating and processing the feature maps in the same way. The only difference is that the second step upsamples by a factor of 2 while the third step upsamples by a factor of 4. This subsequent upsampling of feature maps after obtaining attention from DAM helps the network to further refine the segmentation results after each step. Lastly, the network's output is obtained by performing a 1 × 1 convolution followed by Softmax activation.

Implementation Details

Firstly, we resized all the images and segmentation masks to a 512 × 512 resolution without preserving the aspect ratio, to reduce training time and computational requirements. We also carried out extensive data augmentation during training to increase dataset size and to avoid overfitting. As for augmentation techniques used, we only selected those transformations which were suitable for segmentation problems and increased the network's robustness. To be precise, we used random crop-and-resize, random mirroring along the vertical axis, random rotation, and lastly, random brightness and saturation distortion.

In the encoder, for the number of channels in each stage, we set $C = 16$. At each stage, the number of channels (C) and the dilation rate (d) were successively increased as shown in **Figure 5A**. For the SE-ResNet block (**Figure 5B**), following Hu et al. (2018), we set the reduction ratio to $r = 8$. In the PDC module for global context aggregation, we set dilation rate to d = {3,5,7}, respectively, for the three parallel branches as shown in **Figure 6B**. Regarding DAM, implementation details are provided in Section "Dense Attention Module (DAM)." For training, following Chen et al. (2018) and Fu et al. (2019), we employed an Adam optimizer along with poly learning rate policy where,

$$l_{r\_new} = l_r * \left(1 - \frac{iter}{total\_iter}\right)^{power} \tag{10}$$

Here, we set power = 0.9, $l_r = 0.005$ and we used weighted cross entropy as a loss function. We adopted dropout of 0.25 and set the mini batch_size = 4. The network is trained for 9K iterations.

RESULTS AND DISCUSSION

Ablation Study for DAM

To evaluate the effectiveness of DAM we performed several experiments and the results are reported in **Table 1**. Our baseline consisted of a simple encoder and decoder as described in Section "Network Architecture" along with simple U-Net (Ronneberger et al., 2015)-like skip-connections. Baseline did not include DAM and PDC modules. It can be seen clearly from **Table 1** that DAM significantly increases the mean IoU from 79.57% to 88.79%, with a slight increase in computational cost. Furthermore, the experiments also show that if we use different pooling operations AMP (i.e., average and max) in different branches then the network performs better as compared to the attention module with only one pooling (i.e., AP or MP). Ablation studies also show the effect of inclusion and exclusion of channel (C_A) and spatial attention (S_A) from DAM [see section "Dense Attention Module (DAM)"].

Ablation Study for PDC

There are a number of modules available for multi-scale context aggregation for obtaining better feature representation from encoder feature maps, like the pyramid pooling module (PPM) of PSP-Net (Zhao et al., 2017). Furthermore, our PDC module is closer to the atrous spatial pyramid pooling modules namely ASPP (v2 and v3) introduced by Deeplab_v2 and Deeplab_v3 (Chen et al., 2017, 2018), respectively. We used a PDC module because it has smaller memory requirement, less floating-point operations (FLOPs), and number of parameters with almost identical performance. The results are summarized in **Table 2**.

DAM Visualization With Segmentation Grad-CAM

For the qualitative analysis, we apply the Grad-CAM (Selvaraju et al., 2017), to show the effects of DAM. Grad-CAM is a gradient-based visualization method, which tries to explain the reasoning behind the decisions made by the DCNNs. It was mainly proposed for classification networks. We propose a modified version of Grad-CAM to evaluate the results of the semantic segmentation model making it into Segmentation Grad-CAM (SGC). If $\left\{A^k\right\}_{k=1}^{K}$ represents the feature map of a selected layer with K feature maps then Grad-CAM calculates the heatmaps by taking the gradient of y^c(logit for a given class) w.r.t to all N pixels (indexed by u, v), in all feature maps of $\left\{A^k\right\}_{k=1}^{K}$. But in the case of segmentation models, instead of y^c (a single value), for each class we have y^c_{ij} (a whole feature map). In this case, the gradients are computed by taking the mean of all M pixels (indexed by i, j) in the feature map of class 'c.' Finally, the weighing vector α^c_k is calculated as;

$$\alpha^c_k = \frac{1}{N} \sum_{u,\, v} \left(\frac{\delta \frac{1}{M} \sum_{(i,j)} y^c_{ij}}{\delta A^k_{u,v}} \right) \tag{11}$$

The heatmaps are then generated by;

$$L^c_{SGC} = ReLu\left(\sum_k \left(\alpha^c_k A^k \right) \right) \tag{12}$$

Thus, SGC can produce heatmaps which explain the reasoning behind the grouping of individual pixels of the input image in one segmented region in the output. We display the activated attention maps of our network at two points in the decoder as shown in **Figure 5A**: firstly, after obtaining attention from DAM of stage S_1 (i.e., point P_1) and secondly after obtaining attention

TABLE 1 | Ablation studies on DAM.

Baseline	AP	MP	AMP	PDC	CA	SA	mIoU	FLOPs	Memory usage	Param.
							(%)	(G)	(GB)	(M)
✓				✓			79.57	**8.11**	1.82	**0.7**
✓	✓			✓	✓	✓	87.13	14.6	1.69	1.26
✓		✓		✓		✓	87.58	14.6	1.69	1.26
✓			✓	✓	✓		87.06	9.49	**1.56**	1.09
✓			✓		✓	✓	87.17	11.01	1.67	1.05
✓			✓	✓	✓	✓	**88.79**	14.6	1.8	1.26

Here AP and MP represent only using average or max pooling in both branches of DAM. AMP represents average and max pooling both being used in respective parallel branches of DAM. PDC is parallel dilated convolution module. C_A and S_A are channel and spatial attention sub-parts of DAM. Bold is used to highlight the best results.

TABLE 2 | Ablation studies for PDC.

Metric		PDC	PPM	ASPP_v2	ASPP_v3
		Dil_rate = {3,5,7} + Skip-connect	Bin-size = {1,2,3,6}	Dil-rate = {6,12,18,24}	Dil-rate = {1,6,12,18} + img_pooling
mIoU	(%)	88.79	86.49	84.67	**88.92**
FLOPs	(G)	14.6	**11.21**	15.86	15.31
Memory	(GB)	**1.8**	1.8	1.83	1.82
Param.	(M)	1.26	**1.22**	3.43	1.64

PPM represents pyramid pooling module of Zhao et al. (2017) and ASPP_v2 and v3 represents the atrous spatial pyramid pooling modules of Chen et al. (2017) and Chen et al. (2018), respectively. Bold is used to highlight the best results.

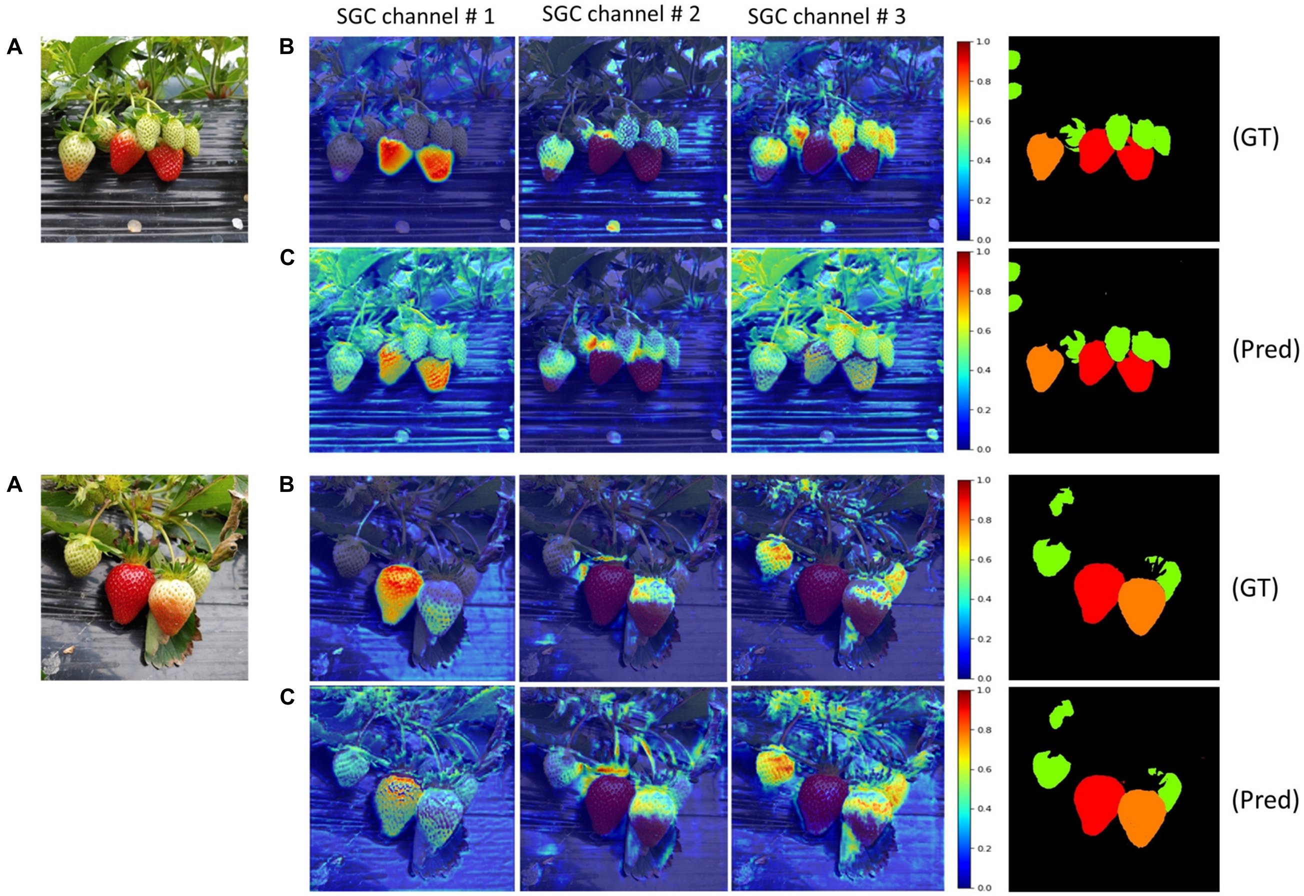

FIGURE 7 | (A) Input images; **(B)** heatmaps at point P1; **(C)** heatmaps at point P2. GT represents ground truth and Pred. is the network's prediction (best viewed in color).

TABLE 3 | Comparison of results on SS1K dataset.

Method	IOR	FLOPS (G)	Param. (M)	Memory usage (GB)	Test mIoU (%)	Precision (%) thresh = 0.75			Recall (%) thresh = 0.75			FPS (sec)
						Ripe	Unripe	Green	Ripe	Unripe	Green	
FCN-8s	1:1	20.82	2.67	5.23	76.1	92.25	74.89	90.04	75.08	78.93	87.02	39
Seg-Net	1:1	41.33	5.46	–	80.7	94.48	**87.85**	84.88	92.49	80.72	90.06	7
U-Net	1:1	70.52	4.65	8.24	83.68	94.27	77.72	89.75	84.52	82.4	89.11	32
DAN	8:1	34.4	**0.65**	5.2	77.12	95.35	65.47	87.58	56.78	65.86	77.12	10
BAM	1:1	20.6	1.1	5.28	87.3	95.63	79.46	**94.31**	91.13	84.86	86.39	36
CBAM	1:1	19.8	1.1	4.8	86.47	95.74	76.84	89.68	86.68	**86.62**	**91.21**	40
PSP-Net	8:1	34.5	0.71	5.4	85.65	94.91	86.42	91.68	84.84	64.64	86.60	18
GCN	1:1	**8.6**	0.85	2.79	80.51	90.98	72.84	87.85	92.64	60.01	83.46	51
Deeplab_v2	1:1	28.6	3.09	2.94	87.07	**95.45**	81.47	90.4	87.13	81.1	87.44	49
Deeplab_v3+	1:1	14.2	1.2	2.93	84.69	93.51	80.21	85.05	88.59	68.44	87.97	**53**
Straw-Net	1:1	14.6	1.26	**1.8**	**88.79**	94.35	86.51	94.16	**94.45**	78.19	89.64	**53**

IOR, input to output ratio of network; FPS, frames per second. For methods; bold is used to highlight the proposed method. For results; bold is used to highlight best results.

from DAM of stage S_2 (i.e., point P_2). The channel #s {1,2,3} correspond to the ripe, unripe, and green class of strawberry, respectively. It can be seen from **Figure 7B** that the heatmaps of all the classes at point P_1 gets further refined and have clearer semantic meaning than those at point P_2. Which shows the effectiveness of incorporating the DAM on skip-connections.

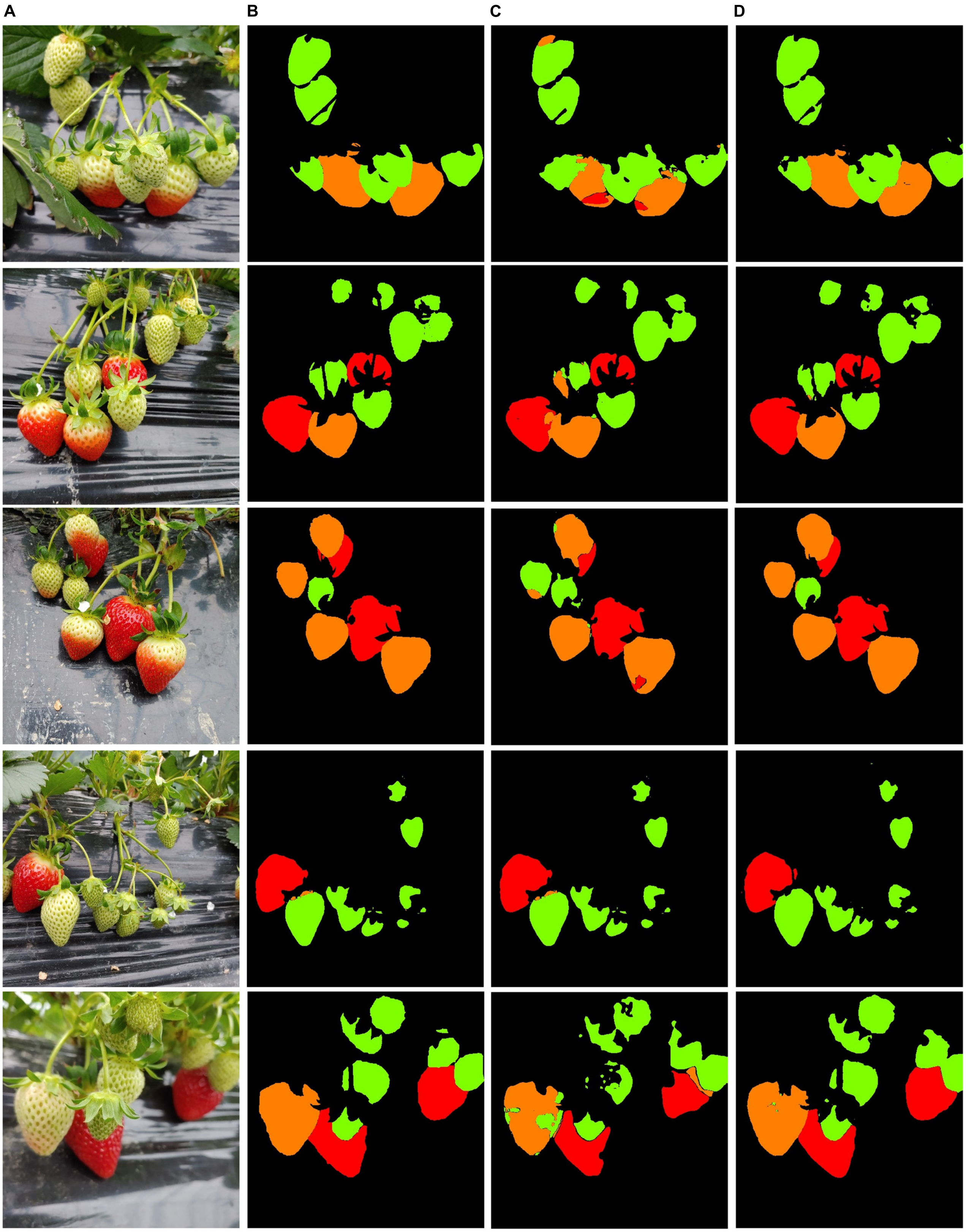

FIGURE 8 | Visualization of some semantic segmentation results on the SS1K dataset. **(A)** Raw images. **(B)** Ground truth. **(C)** Deeplab v3+. **(D)** Straw-Net (best viewed in color).

For better visualization, all the heatmaps in **Figure 7** have been rescaled to the same size.

Comparison With State-of-the-Art Networks

In this sub-section, we compare the results of our network on the SS1K dataset with other existing state-of-the-art models in semantic segmentation. We evaluate all the models on different benchmark metrics and report the results in **Table 3** and **Figure 8** shows some visual semantic segmentation results. All the values reported in **Table 3** are an average of 10 runs by default. Here IOR represents the input image to segmented output ratio. All the networks have a 1:1 ratio which means they output feature maps of the same size as the input, except DAN (Fu et al., 2019) and PSP-net (Zhao et al., 2017), their segmented output is eight times smaller than the input. Intersection over union value is averaged over all four classes. For precision and recall, the values are reported for each class separately and are calculated at a threshold of 0.75. It can be seen from **Table 3** that our Straw-Net outperforms all other existing networks overall for real-time semantic segmentation of strawberry fruits. All the metrics including frames per second (fps) are calculated for 512 × 512 resolution images, on a single Nvidia Titan RTX-2080 GPU. From **Table 3**, it can be seen that our proposed network, even though incorporating an attention mechanism is much faster, requires less memory (GB) and less floating-point operations (FLOPs) as compared to other attention networks like DAN (Fu et al., 2019), BAM (Park et al., 2018), and CBAM (Woo et al., 2018). On the other hand, compared to other existing state-of-the-art segmentation models like Deeplab_v2 and Deeplab_v3 (Chen et al., 2017, 2018), our proposed approach is able to achieve a highest mean intersection over union (mIoU) value and comparable precision recall scores. The detailed architectures of all the networks used for comparison are provided as **Supplementary Material**.

Further Analysis

To analyze the results further and to see which classes confuse the network resulting in lower performance, we plot a precision-recall (PR) curve and confusion matrix of the final segmentation results, as shown in **Figures 9**, **10**. From the results we can analyze the networks performance visually and see which classes or features are highlighted by neurons. Moreover, it will also help us to take precautionary measures to avoid inter-class confusions. For instance, in **Figure 9** the confusion matrix shows that the network is more confused between ripe and unripe strawberries rather than between unripe and green strawberries.

Another reason for this instability in confusion matrix is data imbalance, as shown in **Figure 4**. Because, there are fewer samples of unripe strawberries per batch as compared to the other two classes. We can analyze the effect of this imbalance from PR-Curves. We plot a PR-Curve for each class in **Figure 10**. PR-Curves represent a trade-off between precision and recall at different thresholds. The area under the precision-recall curve is usually denoted as AUC (i.e., area under the curve). A high value of AUC means high precision and recall. Whether we want high precision or high recall depends on the application domain. In **Figure 10**, ISO-F1 curves represents the lines in precision-recall space which have the same F1-values. We can see from **Figure 10** that the AUC for unripe strawberries is much less than the ripe and green strawberries which in turn means that precision and recall values are also low for unripe strawberries. Moreover, the micro-average curve represents the mean of PR-Curves of all classes considering data imbalance. Whereas, the macro-average curve represents the mean PR-Curve without considering data imbalance.

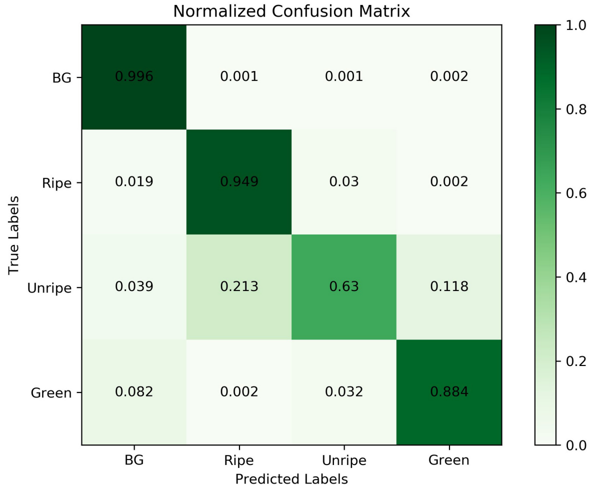

FIGURE 9 | Confusion matrix for Straw-Net architecture for semantic segmentation of the SS1K dataset.

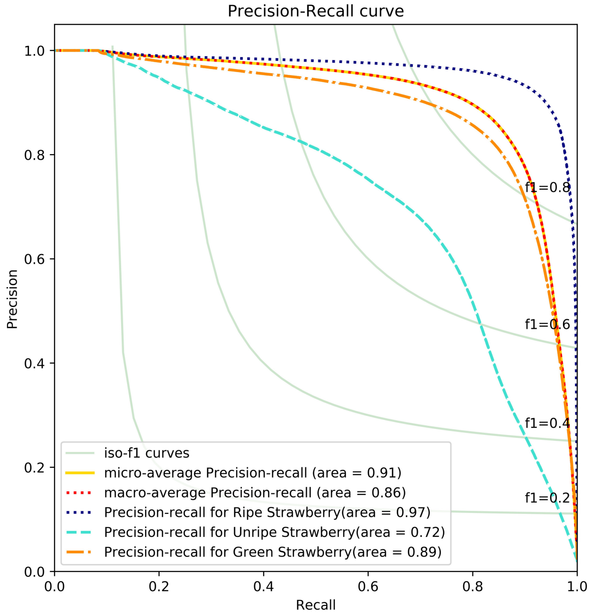

FIGURE 10 | Precision recall curve obtained for Straw-Net architecture on the SS1K dataset (Best viewed in color).

TABLE 4 | Comparison of different system configurations on network's (Straw-Net) inference speed.

System	OS	CPU	Clock speed (GHz)	GPU (Nvidia)	FLOPS (Tera)	Power consumption (Watts)	FPS (sec)
Server	Linux 18.04	Core i9-9940X	3.3	RTX-2080	14.2	–	53
Desktop PC	Linux 16.04	Core i7-9700	3.0	GTX-1650	5.5	180 ∼ 300	28.9
Desktop PC	Linux 16.04	Core i7-9700	3.0	None	–	180 ∼ 300	13.8
Portable devices							
Laptop0	Windows 10	Core i7-10750	2.6	RTX-2070	6.6	170	40
Laptop1	Windows 10	Core i7-9750	2.59	None	–	70	6.32
Laptop1	Windows 10	Core i7-9750	2.59	GTX-1650	3.2	120	21.3
Laptop2	Windows 10	Core i5-8265	1.8	None	–	48	3.38
Embedded systems							
Nvidia Jetson TX2	Linux 16.04	ARM-Cortex A57	2.0	Pascal GPU	1.3	35	15.3

Real-Time Performance Barrier

Neurons in the human visual system can interpret 10 to 12 fps and perceive them individually (Read and Meyer, 2000), whereas higher frame rates are perceived as motion. To reduce eye strain, the standard frame rate was set to be anywhere between 16 and 25 fps (Brown, 2014). Nowadays, all available video cameras have the minimum frame rate of 24 fps (Brunner, 2021).

For example, let us assume that a camera is generating 24 fps and sending those frames as an input to the proposed architecture, then the proposed algorithm should be able to process all those frames within a second to produce an output that is perceivable to the human eye. Therefore, if an algorithm can achieve a speed above this threshold (≥16 fps) it is said to have crossed the real-time barrier, where this limitation is mainly generated by the human perception system. In the case of ASH, if an algorithm has a processing speed of ≥24 fps it means that it will generate outputs (i.e., strawberry segments) after processing all the input frames. The processing speed of 53 fps was the maximum frame rate that was achieved during the experiments with the highest system configuration, i.e., RTX-2080 GPU and Core i9-9940X CPU as shown in **Table 4**. In contrast, for most sluggish situations, let us consider that a system can only process 3 fps (Laptop2 Core i5-8265 no GPU). In this case we might have to quantize our frames so that the network can process them before the next batch arrives. Thus, the statistical value of the output generated by a 53 fps system would be higher than the output generated by a 3 fps system.

Our model is adaptive, easily scalable, has a small computational footprint of 14.6 GFLOPS (**Table 3**), and an even smaller memory footprint of 1.8 GB (**Table 3**). Therefore, it can be easily implemented on machines with low computational power like laptops with (40 and 21.3 FPS) or without (3.38 FPS) GPU or even on embedded systems like an NVIDIA Jetson TX2 board (15.3 FPS) without any loss in precision and accuracy. Therefore, any system configuration in ASH operating at the speed of ≥16 fps would overcome the real-time barrier and will be suitable for autonomous harvesting.

Effect of Input Resolution

To demonstrate the effect of change in resolution on the inference speed and precision of the network, we consider two more resolutions in addition to 512 × 512, i.e., 256 × 256 (low resolution) and 1024 × 1024 (high resolution). The results are reported in **Table 5** (all experiments were performed under the same conditions).

From the results we can see that if we reduce the resolution to 256 × 256 the computational complexity of the network is reduced considerably, and the speed is increased. Moreover, there is no significant decline in mIOU, but if we look at the AP, it is decreased by 14.8%. In contrast, if we analyze the case of high resolution (1024 × 1024), we can see that there is a little increase of about 0.58% in the network's performance, but the computational complexity has exploded, and inference speed is now considerably slower than the 512 × 512 version. Therefore, we recommend using the 512 × 512 resolution.

TABLE 5 | Performance comparison for input images of varying resolution.

Metric		**Image resolution**		
		256 × 256	***512 × 512***	***1024 × 1024***
mIoU	(%)	85.31	**88.79**	88.21
mAP	(%)	76.87	91.67	**92.06**
FLOPs	(G)	**3.6**	14.6	58
Memory	(GB)	**0.46**	1.8	7.19
Param.	(M)	1.26	1.26	1.26
FPS	(sec)	**69**	53	20

*GPU used for comparison is RTX-2080 (**Table 4**). Bold is used to highlight the best results.*

CONCLUSION

In this paper, a new dataset (i.e., SS1K) is introduced for the segmentation of strawberries into four classes depending upon the ripeness of the fruit (including a background class). The proposed segmentation network named Straw-Net improves the performance of ASHs in unconstrained and natural farming environments. Also, a real-time attention mechanism (DAM) is developed for integrating local and global semantic features efficiently. DAM controls the flow of information between the network's encoder and decoder, enabling efficient feature fusion. Integrating adaptive feature fusion on skip-connections results in improved segmentation and classification ability of the network as shown by Segmentation Grad-CAM. The proposed attention mechanism can be integrated with any existing DCNN without any modification. By incorporating DAM in our baseline model, we achieved a significant performance boost while keeping the computational complexity to a minimum. Moreover, the effectiveness of DAM is verified by performing extensive ablation experiments. To verify the overall efficacy of the proposed approach, we compared the results with other attention mechanisms as well as with existing state-of-the-art segmentation models. Results demonstrated enhanced performance, i.e., improved mIoU, recall, and precision score with the proposed method on the strawberry segmentation problem. Our future work involves incorporating the proposed approach with ASH for deployment in strawberry farms.

DATA AVAILABILITY STATEMENT

The datasets generated for this study is available on request to the corresponding author.

AUTHOR CONTRIBUTIONS

TI designed the study, collected the data, performed the experiments, analyzed the data, and wrote the manuscript. MU and AK collected the data and performed the experiments. HK supervised and administered the overall project and reviewed and edited the writing. All authors contributed to the article and approved the submitted version.

FUNDING

This work was supported in part by the Korean government (NRF-2019R1A2C1011297 and NRF-2019R1A6A1A09031717) and the US Air Force Office of Scientific Research under grant number FA9550-18-1-0016.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fpls.2021.591333/full#supplementary-material

REFERENCES

Adhikari, S. P., Yang, H., and Kim, H. (2019). Learning semantic graphics using convolutional encoder-decoder network for autonomous weeding in paddy field. *Front. Plant Sci.* 10:1404. doi: 10.3389/fpls.2019.01404

Agrobot (2020). *Agrobot - Agricultural Robots*. Available online at: https://www.agrobot.com/ [Accessed July 6, 2020]

Badrinarayanan, V., Kendall, A., and Cipolla, R. J. (2017). Segnet: a deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 2481–2495. doi: 10.1109/tpami.2016.2644615

Bargoti, S., and Underwood, J. (2017). Image segmentation for fruit detection and yield estimation in apple orchards. *J. Field Robot.* 34, 1039–1060. doi: 10.1002/rob.21699

Barthes, A. (2010). *NAIO Technologies, Automated Robots and Farming Tools*. Available online at: https://www.naio-technologies.com/en/agricultural-equipment/ [Accessed October 10, 2020]

Bolya, D., Zhou, C., Xiao, F., and Lee, Y. J. (2019). "Yolact: real-time instance segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, (New Jersey, NJ: IEEE), 9157–9166.

Brown, J. (2014). "Audio-visual palimpsests: resynchronizing silent films with "Special" music," in *The Oxford Handbook of Film Music Studies*, ed. D. Neumeyer. (New York, NY: Oxford University Press), 588–610.

Brunner, D. (2021). *Frame Rate: A Beginner's Guide*. Available online at: https://www.techsmith.com/blog/frame-rate-beginners-guide/#:~{}:text=24fps%20%E2%80%93This%20is%20the%20standard,and%20viewed%20at%2024%20fps. [Accessed January 1, 2021]

Buemi, F. (1995). "Agrobot: a robotic system for greenhouse operations", in *Proceedings of the 4th Workshop on Robotics in Agriculture & the Food Industry*, (Toulouse: IARP)

Buemi, F., Massa, M., Sandini, G., and Costi, G. (1996). The agrobot project. *Adv. Space Res.* 18, 185–189. doi: 10.1016/0273-1177(95)00807-q

Chen, L. -C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017). Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach Intell.* 40, 834–848. doi: 10.1109/tpami.2017.2699184

Chen, L. -C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. (2018). "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European conference on computer vision ECCV*, (Munich: ECCV), 801–818.

Chen, Y., Lee, W. S., Gan, H., Peres, N., Fraisse, C., Zhang, Y., et al. (2019). Strawberry yield prediction based on a deep neural network using high-resolution aerial orthoimages. *Remote Sens.* 11:1584. doi: 10.3390/rs11131584

Chollet, F. (2017). "Xception: deep learning with depthwise separable convolutions", in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (New Jersey, NJ: IEEE), 1251–1258.

Corbetta, M., and Shulman, G. L. (2002). Control of goal-directed and stimulus-driven attention in the brain. *Nat. Rev. Neurosci.* 3, 201–215. doi: 10.1038/nrn755

Durand-Petiteville, A., Vougioukas, S., and Slaughter, D. C. (2017). Real-time segmentation of strawberry flesh and calyx from images of singulated strawberries during postharvest processing. *Comput. Electron. Agric.* 142, 298–313. doi: 10.1016/j.compag.2017.09.011

Dyrmann, M., Jørgensen, R. N., and Midtiby, H. S. (2017). RoboWeedSupport-detection of weed locations in leaf occluded cereal crops using a fully convolutional neural network. *Adv. Anim. Biosci.* 8, 842–847. doi: 10.1017/s2040470017000206

Dyrmann, M., Karstoft, H., and Midtiby, H. S. (2016). Plant species classification using deep convolutional neural network. *Biosyst. Eng.* 151, 72–80. doi: 10.1016/j.biosystemseng.2016.08.024

Edan, Y., Rogozin, D., Flash, T., and Miles, G. E. (2000). Robotic melon harvesting. *IEEE Trans. Rob. Autom.* 16, 831–835. doi: 10.1109/70.897793

Epifanio, I., and Soille, P. J. (2007). "Morphological texture features for unsupervised and supervised segmentations of natural landscapes," in *Proceedings of the IEEE Transactions on Geoscience and Remote Sensing*, (New Jersey, NJ: IEEE), 45, 1074–1083. doi: 10.1109/tgrs.2006.890581

FAO (2020). *Chapter 1: Requirements for Producing and Exporting Organic Products to Major Markets*. Rome: FAO.

Feng, Q., Wang, X., Zheng, W., Qiu, Q., and Jiang, K. (2012). New strawberry harvesting robot for elevated-trough culture. *Int. J. Agric. Biol. Eng.* 5:1.

Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., et al. (2019). "Dual attention network for scene segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (New Jersey, NJ: IEEE) 3146–3154.

Fuentes, A. F., Yoon, S., Lee, J., and Park, D. S. (2018). High-performance deep neural network-based tomato plant diseases and pests diagnosis system with refinement filter bank. *Front. Plant Sci.* 9:1162. doi: 10.3389/fpls.2018.01162

Fujiyoshi, H., Hirakawa, T., and Yamashita, T. (2019). Deep learning-based image recognition for autonomous driving. *IATSS Res.* 43, 244–252. doi: 10.1016/j.iatssr.2019.11.008

Gao, H., Wang, Z., and Ji, S. (2018). "Channelnets: compact and efficient convolutional neural networks via channel-wise convolutions. *Adv. Neural Inf. Process. Syst.* 5197–5205.

Ge, Y., Xiong, Y., Tenorio, G. L., and From, P. J. (2019). Fruit localization and environment perception for strawberry harvesting robots. *IEEE Access* 7, 147642–147652. doi: 10.1109/access.2019.2946369

Grand d'Esnon, A., Rabatel, G., Pellenc, R., and Journeau, A. (1987). *Magali—A Self-Propelled Robot to Pick Apples*. ASAE Paper No. 87-1037. St. Joseph, Mich: ASAE.

Grinblat, G. L., Uzal, L. C., Larese, M. G., and Granitto, P. M. (2016). Deep learning for plant identification using vein morphological patterns. *Comput. Electron. Agric.* 127, 418–424. doi: 10.1016/j.compag.2016.07.003

Guerrero, J. M., Ruz, J. J., and Pajares, G. (2017). Crop rows and weeds detection in maize fields applying a computer vision system based on geometry. *Comput. Electron. Agric.* 142, 461–472. doi: 10.1016/j.compag.2017.09.028

Harvest Croo (2021). *Agricultural Robotics*. Available online at: https://harvestcroo.com/ [Accessed January 7, 2021]

Hasanpour, S. H., Rouhani, M., Fayyaz, M., and Sabokrou, M. (2016). Lets keep it simple, using simple architectures to outperform deeper and more complex architectures. *arXiv [Preprint]* Available online at: https://arxiv.org/abs/1608.06037 (accessed January 3, 2020).

Hayashi, S., Yamamoto, S., Saito, S., Ochiai, Y., Kamata, J., Kurita, M., et al. (2014). Field operation of a movable strawberry-harvesting robot using a travel platform. *Jpn. Agric. Res. Q.* 48, 307–316. doi: 10.6090/jarq.48.307

Hayashi, S., Yamamoto, S., Saito, S., Ochiai, Y., Nagasaki, Y., and Kohno, Y. (2013). Structural environment suited to the operation of a strawberry-harvesting robot mounted on a travelling platform. *Eng. Agric. Environ. Food.* 6, 34–40. doi: 10.1016/s1881-8366(13)80015-8

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, (New Jersey, NJ: IEEE) 2961–2969.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (New Jersey, NJ: IEEE), 770–778.

Hofmarcher, M., Unterthiner, T., Arjona-Medina, J., Klambauer, G., Hochreiter, S., and Nessler, B. (2019). "Visual scene understanding for autonomous driving using semantic segmentation," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, eds W. Samek, G. Montavon, A. Vedaldi, L. Hansen, and K. R. Müller (Berlin: Springer), 285–296. doi: 10.1007/978-3-030-28954-6_15

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017). Mobilenets: efficient convolutional neural networks for mobile vision applications. *arXiv [Preprint]* Available online at: https://arxiv.org/abs/1704.04861 (accessed December 14, 2019).

Hu, J., Shen, L., and Sun, G. (2018). "Squeeze-and-excitation networks,", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (New Jersey, NJ: IEEE) 7132–7141.

Huang, Z., Pan, Z., and Lei, B. (2017). Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data. *Remote Sens.* 9:907. doi: 10.3390/rs9090907

Hubel, D. H., and Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* 160, 106–154.2.

Ienco, D., Gaetano, R., Dupaquier, C., and Maurel, P. (2017). Land cover classification via multitemporal spatial data by deep recurrent neural networks. *IEEE Geosci. Remote Sens. Lett.* 14, 1685–1689. doi: 10.1109/lgrs.2017.2728698

Ilyas, T., Khan, A., Umraiz, M., and Kim, H. (2020). SEEK: a framework of superpixel learning with CNN Features for unsupervised segmentation. *Electronics* 9:383. doi: 10.3390/electronics9030383

Imai, T. (2019). Legal regulation of autonomous driving technology: current conditions and issues in Japan. *IATSS Res.* 43, 263–267. doi: 10.1016/j.iatssr.2019.11.009

Işın, A., Direkoğlu, C., and Şah, M. (2016). Review of MRI-based brain tumor image segmentation using deep learning methods. *Procedia Comput. Sci.* 102, 317–324. doi: 10.1016/j.procs.2016.09.407

Jin, J., Dundar, A., and Culurciello, E. (2014). Flattened convolutional neural networks for feedforward acceleration. *arXiv [Preprint]* Available online at: https://arxiv.org/abs/1412.5474 (accessed January 30, 2020).

Kahani, A. (2017). *The Future of Fresh Fruit Harvest*. Available online at: https://www.ffrobotics.com/ [Accessed January 8, 2020]

Kauanova, S., Vorobjev, I., and James, A. P. (2017). "Automated image segmentation for detecting cell spreading for metastasizing assessments of cancer development," in *Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (New Jersey, NJ: IEEE), 2382–2387

Khan, A., Ulhaq, A., Robinson, R., and Rehman, M. U. (2020). "Detection of vegetation in environmental repeat photography: a new algorithmic approach in data science," in *Statistics for Data Science and Policy Analysis*. (Berlin: Springer), 145–157. doi: 10.1007/978-981-15-1735-8_11

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 25, 1097–1105.

Kussul, N., Lavreniuk, M., Skakun, S., and Shelestov, A. (2017). Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geosci. Remote Sens. Lett.* 14, 778–782. doi: 10.1109/lgrs.2017.2681128

Lamb, N., and Chuah, M. C. (2018). "A strawberry detection system using convolutional neural networks," in *Proceedings of the IEEE International Conference on Big Data (Big Data)*, (New Jersey, NJ: IEEE), 2515–2520.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., et al. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Comput.* 1, 541–551. doi: 10.1162/neco.1989.1.4.541

Lin, G., Milan, A., Shen, C., and Reid, I. (2017a). "Refinenet: multi-path refinement networks for high-resolution semantic segmentation", in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (New Jersey, NJ: IEEE), 1925–1934.

Lin, Z., Feng, M., Santos, C. N. D., Yu, M., Xiang, B., Zhou, B., et al. (2017b). A structured self-attentive sentence embedding. *arXiv [Preprint]* Available online at: https://arxiv.org/abs/1703.03130 (accessed February 13, 2020).

Lin, Z., Ji, K., Kang, M., Leng, X., and Zou, H. (2017c). Deep convolutional highway unit network for SAR target classification with limited labeled training data. *IEEE Geosci. Remote Sens. Lett.* 14, 1091–1095. doi: 10.1109/lgrs.2017.2698213

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., et al. (2016). "Ssd: single shot multibox detector," in *Proceedings of the European Conference on Computer Vision*, (Berlin: Springer), 21–37.

Long, J., Shelhamer, E., and Darrell, T. (2015). "Fully convolutional networks for semantic segmentation", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (New Jersey, NJ: IEEE) 3431–3440.

Ma, X., Deng, X., Qi, L., Jiang, Y., Li, H., Wang, Y., et al. (2019). Fully convolutional network for rice seedling and weed image segmentation at the seedling stage in paddy fields. *PLoS One* 14:e0215676. 10.1371/journal.pone.0215676 doi: 10.1371/journal.pone.0215676

Mohanty, S. P., Hughes, D. P., and Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Front. Plant Sci.* 7:1419. doi: 10.3389/fpls.2016.01419

Nizami, I. F., Ur Rehman, M., Majid, M., and Anwar, S. (2020). *Natural Scene Statistics Model Independent No-Reference Image Quality Assessment using Patch Based Discrete Cosine Transform*. Berlin: Springer.

Noh, H., Hong, S., and Han, B. (2015). "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, (New Jersey, NJ: IEEE) 1520–1528.

Ouyang, C., Li, D., Wang, J., Wang, S., and Han, Y. (2012). "The research of the strawberry disease identification based on image processing and pattern recognition," in *Proceedings of the International Conference on Computer and Computing Technologies in Agriculture* (Berlin: Springer), 69–77. doi: 10.1007/978-3-642-36124-1_9

Park, J., Woo, S., Lee, J. -Y., and Kweon, I. (2018). Bam: bottleneck attention module. *arXiv [Preprint]* Available online at: https://arxiv.org/abs/1807.06514 (accessed March 19, 2020).

Park, S., Wahab, A., Nazari, I., Ryu, J. H., and Chong, K. T. (2020). i6mA-DNC: prediction of DNA N6-Methyladenosine sites in rice genome based on dinucleotide representation using deep learning. *Chemometr. Intell. Lab. Syst.* 204:104102. doi: 10.1016/j.chemolab.2020.104102

Peng, C., Zhang, X., Yu, G., Luo, G., and Sun, J. (2017). "Large kernel matters–improve semantic segmentation by global convolutional network", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (New Jersey, NJ: IEEE) 4353–4361.

Qingchun, F., Wengang, Z., Quan, Q., Kai, J., and Rui, G. (2012). "Study on strawberry robotic harvesting system," in *Proceeding of the 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)* (New Jersey, NJ: IEEE), 320–324.

Read, P., and Meyer, M.-P. (2000). *Restoration of Motion Picture Film*. Amsterdam: Elsevier.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (New Jersey, NJ: IEEE) 779–788.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). "Faster r-cnn: towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* 1, 91–99.

Rensink, R. A. (2000). The dynamic representation of scenes. *Vis. Cogn.* 7, 17–42. doi: 10.1080/135062800394667

Ronneberger, O., Fischer, P., and Brox, T. (2015). "U-net: convolutional networks for biomedical image segmentation," in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention* (Berlin: Springer), 234–241.

Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., and Mccool, C. J. (2016). Deepfruits: a fruit detection system using deep neural networks. *Sensors* 6:1222. doi: 10.3390/s16081222

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, (New Jersey, NJ: IEEE) 618–626.

Shen, T., Zhou, T., Long, G., Jiang, J., Pan, S., and Zhang, C. (2018). "Disan: Directional self-attention network for rnn/cnn-free language understanding," in *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. (Menlo Park, CA: AAAI)

Simonyan, K., and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv [Preprint]* Available online at: https://arxiv.org/abs/1409.1556 (accessed October 19, 2019).

Strawberry (2020). *Growing Strawberries*. Available online at: https://strawberryplants.org/growing-strawberries/ [Accessed July 27, 2020].

Sun, J., Lu, B., and Mao, H. (2011). Fruits recognition in complex background using binocular stereovision. *J. Jiangsu U. Nat. Sci. Ed.* 32, 423–427.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (New Jersey, NJ: IEEE) 2818–2826.

ur Rehman, M., Khan, S. H., Rizvi, S. D., Abbas, Z., and Zafar, A. (2018). "Classification of skin lesion by interference of segmentation and convolotion neural network," in *Proceedings of the 2018 2nd International Conference on Engineering Innovation (ICEI)* (New Jersey, NJ: IEEE), 81–85.

Wahab, A., Ali, S. D., Tayara, H., and Chong, K. T. (2019). iIM-CNN: intelligent identifier of 6mA sites on different species by using convolution neural network. *IEEE Access* 7, 178577–178583. doi: 10.1109/access.2019.2958618

Wahab, A., Mahmoudi, O., Kim, J., and Chong, K. T. (2020). DNC4mC-deep: identification and analysis of DNA N4-methylcytosine sites based on different encoding schemes by using deep learning. *Cell* 9:1756. doi: 10.3390/cells9081756

Wang, M., Liu, B., and Foroosh, H. (2016). Design of efficient convolutional layers using single intra-channel convolution, topological subdivisioning and spatial" bottleneck" structure. *arXiv [Preprint]* Avilable at: https://arxiv.org/abs/1608.04337 (accessed October 26, 2019).

Wei, X., Jia, K., Lan, J., Li, Y., Zeng, Y., and Wang, C. J. O. (2014). Automatic method of fruit object extraction under complex agricultural background for vision system of fruit picking robot. *Optik* 125, 5684–5689. doi: 10.1016/j.ijleo.2014.07.001

Woo, S., Park, J., Lee, J.-Y., and So Kweon, I. (2018). "Cbam: convolutional block attention module," in *Proceedings of the European Conference on Computer Vision* (Munich: ECCV), 3–19. doi: 10.1007/978-3-030-01234-2_1

Xia, X., and Kulis, B. (2017). W-net: a deep model for fully unsupervised image segmentation. *arXiv [Preprint]* Available online at: https://arxiv.org/abs/1711.08506 (accessed December 7, 2020).

Xiang, R., Jiang, H., and Ying, Y. (2014). Recognition of clustered tomatoes based on binocular stereo vision. *Comput. Electron. Agric.* 106, 75–90. doi: 10.1016/j.compag.2014.05.006

Xiong, Y., Peng, C., Grimstad, L., From, P. J., and Isler, V. (2019). Development and field evaluation of a strawberry harvesting robot with a cable-driven gripper. *Comput. Electron. Agric.* 157, 392–402. doi: 10.1016/j.compag.2019.01.009

Yamamoto, S., Hayashi, S., Saito, S., Ochiai, Y., Yamashita, T., and Sugano, S. J. (2010). Development of robotic strawberry harvester to approach target fruit from hanging bench side. *IFAC Proc. Vol.* 43, 95–100. doi: 10.3182/20101206-3-jp-3009.00016

Yamamoto, S., Hayashi, S., Yoshida, H., and Kobayashi, K. (2014). Development of a stationary robotic strawberry harvester with a picking mechanism that approaches the target fruit from below. *Jpn Agric. Res. Q.* 8, 261–269. doi: 10.6090/jarq.48.261

Yu, F., and Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *arXiv [Preprint]* Available online at: https://arxiv.org/abs/1511.07122 (accessed March 21, 2020).

Yu, F., Koltun, V., and Funkhouser, T. (2017). "Dilated residual networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (New Jersey, NJ: IEEE) 472–480.

Yu, Y., Zhang, K., Yang, L., and Zhang, D. (2019). Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. *Comput. Electron. Agric.* 163:104846. doi: 10.1016/j.compag.2019.06.001

Zhang, B., Huang, W., Wang, C., Gong, L., Zhao, C., Liu, C., et al. (2015). Computer vision recognition of stem and calyx in apples using near-infrared linear-array structured light and 3D reconstruction. *Biosyst. Eng.* 139, 25–34. doi: 10.1016/j.biosystemseng.2015.07.011

Zhang, K., Yang, L., Wang, L., Zhang, L., and Zhang, T. (2012). Design and experiment of elevated substrate culture strawberry picking robot. *Trans. Chin. Soc. Agric. Machinery* 43, 165–172.

Zhang, X., Zhou, X., Lin, M., and Sun, J. (2018). "Shufflenet: an extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (New Jersey, NJ: IEEE) 6848–6856.

Zhao, H., and Kit, C. (2011). Integrating unsupervised and supervised word segmentation: the role of goodness measures. *Inf. Sci.* 181, 163–183. doi: 10.1016/j.ins.2010.09.008

Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. (2017). "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (New Jersey, NJ: IEEE) 2881–2890.

Zhao, Y., Gong, L., Huang, Y., and Liu, C. (2016). A review of key techniques of vision-based control for harvesting robot. *Comput. Electron. Agric.* 127, 311–323. doi: 10.1016/j.compag.2016.06.022

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2014). Object detectors emerge in deep scene cnns. *arXiv [Preprint]* Available online at: https://arxiv.org/abs/1412.6856 (accessed May 6, 2020).

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.