Neurocomputing 562 (2023) 126914

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Weed mapping in multispectral drone imagery using lightweight vision transformers

Giovanna Castellano, Pasquale De Marinis *, Gennaro Vessio

Department of Computer Science, University of Bari Aldo Moro, Bari, Italy

ARTICLE INFO

Communicated by X. Gu

Keywords:
Computer vision
Deep learning
Drones
Precision agriculture
Semantic segmentation
Weed mapping
UAV

ABSTRACT

In precision agriculture, non-invasive remote sensing can be used to observe crops and weeds in visible and non-visible spectra. This paper proposes a novel approach for weed mapping using lightweight Vision Transformers. The method uses a lightweight Transformer architecture to process high-resolution aerial images obtained from drones and performs semantic segmentation to distinguish between crops and weeds. The method also employs specific architectural designs to enable transfer learning from RGB weights in a multispectral setting. For this purpose, the WeedMap dataset, acquired by drones equipped with multispectral cameras, was used. The experimental results demonstrate the effectiveness of the proposed method, exceeding the state-of-the-art. Our approach also enables more efficient mapping, allowing farmers to quickly and easily identify infested areas and prioritize their control efforts. These results encourage using drones as versatile computer vision flying devices for herbicide management, thereby improving crop yields. The code is available at https://github.com/pasqualedem/LWViTs-for-weedmapping.

1. Introduction

The Food and Agriculture Organization (FAO) has estimated that world population growth will reach nine billion people by 2050, and demand for food will double. This is while the natural resources that sustain agriculture will become increasingly scarce, degraded, and vulnerable to climate change [1]. To address these challenges, there is a need to better understand complex, multivariate agricultural ecosystems by monitoring, measuring, and analyzing the physical aspects and phenomena that occur. To this aim, the agricultural sector has experienced significant growth in using civilian satellites, autonomous field robots, and unmanned aerial vehicles (UAVs) [2,3]. In particular, UAVs, also known as drones, are becoming increasingly popular due to their versatility and low cost. This applies not only to precision agriculture, but also to many other areas, such as crowd analysis [4] or cinematography-oriented tasks [5].

In precision agriculture, in particular, there has been significant interest in the successful transfer of non-agricultural applications, based on deep learning, to agricultural ones [6,7]. This is due to the promise that such techniques can facilitate more informed decision-making and management. Among the various tasks studied in this context, there is weed mapping. Weeds are unwanted plants that grow in fields and take vital resources away from crop plants, reducing their yields. Weed mapping is one of the Site-Specific Weed Management (SSWM) steps, which involves applying herbicides precisely rather than spraying them over the entire field. The excessive use of herbicides can favor the evolution of herbicide-resistant weeds and affect crop growth. In addition, herbicides pose a severe threat to the environment. Finally, their use is expensive.

Recent work that solves weed mapping treats the problem as a semantic segmentation task based on Convolutional Neural Networks (CNNs) [8–10]. However, despite progress to date, it is still challenging to have models that simultaneously optimize effectiveness and efficiency, mainly because the complexity of neural networks makes them difficult to run on devices with limited hardware, such as drones. In recent years, Transformer-based models have emerged as a powerful alternative tool in computer vision tasks [11,12], but computing them is often as or more expensive. In this paper, we contribute to this research by proposing a novel approach based on lightweight Vision Transformers that achieve state-of-the-art weed mapping results without harming inference time. To this end, we have leveraged the challenging WeedMap dataset [13] and proposed specific architectural designs aimed at better reusing weights already pre-trained on RGB images so that they can be used to improve performance in a multispectral setting significantly. This technology can improve weed management practices, leading to more sustainable and efficient agriculture.

The rest of this paper is structured as follows. Section 2 reviews related work. Sections 3 and 4 describe materials and methods. Section 5

* Corresponding author.
E-mail address: pasquale.demarinis@uniba.it (P. De Marinis).

https://doi.org/10.1016/j.neucom.2023.126914
Received 6 March 2023; Received in revised form 29 August 2023; Accepted 8 October 2023
Available online 10 October 2023

reports and discusses the experimental results. Section 6 concludes the paper and outlines future developments in our research.

2. Related work

Many precision agriculture tasks have been addressed recently due to the rapid development of computer vision techniques and data collection methodologies through remote sensing. Recent tasks include disease and pest identification, abiotic stress assessment, growth monitoring, crop yield prediction, and weed classification/mapping.

Disease and pest identification is the technical basis for diagnosing and controlling crop diseases, and ensuring the safety of agricultural products. For example, Zeng et al. [14] proposed a self-attention CNN to identify crop diseases, while Liu et al. [15] developed a new dataset for forest pest identification. To achieve a more specific goal, researchers have employed quantitative analysis to assess the severity of the disease. A dataset of 7669 images of maize fields was collected in the summer of 2017 using a camera mounted on a DJI Matrice 600 UAV [16]. Using this dataset, Garg et al. developed an end-to-end deep learning model called Cascaded MRCNN [17]. They aimed to identify northern leaf blight disease in field maize, utilizing a severity index they defined. The model demonstrated an accuracy of up to 73% when evaluated against this specific task.

Abiotic stress is the negative impact of nonliving factors on living organisms in a specific environment. Crops face a severe threat when exposed to multiple abiotic stresses such as water scarcity, salinity, or heat [18]. These stresses induce various plant symptoms, observed in visible and non-visible spectral bands captured through imaging. To effectively detect and evaluate the severity of abiotic stress incidents, a promising approach involves combining distance and proximity sensor technology with deep learning techniques [19]. This integration has the potential to play a crucial role in accurately identifying and assessing the impact of abiotic stress on crops. Chandel et al. [20] compared AlexNet, GoogLeNet, and InceptionV3 to identify maize water stress. Feng et al. [21] exploited hyperspectral images to obtain high-throughput phenotyping of salt-stressed plants.

Growth monitoring is essential in decision-making and is a crucial metric for quantifying crop yield [22]. Traditional methods for assessing crop growth stages and nutritional status rely on expert visual inspection or chemical laboratory analysis. However, these approaches are time-consuming and impractical for on-site monitoring in large-scale conditions. To address this challenge, image processing technology based on machine learning has emerged as a potential solution for continuous monitoring throughout the entire crop life cycle, providing real-time information on crop health and nutrient status [23]. Abdalla and colleagues [24] combined a CNN and an LSTM to classify the nutrient status of oilseed rapes. This model can identify nutrient deficiencies that may limit yield. In 2020, Rasti et al. [25] conducted a study using a cell phone and a DJI Osmo+ camera to capture high-quality videos from a nadir direction and at a 45° angle. The collected data were used to compare different models, including CNN-based and SVM classifiers. The results showed that CNN-based models outperformed SVM classifiers, with VGG19 achieving the highest accuracy among the tested models.

Accurate crop yield prediction is crucial for trade, policy-making, decision support, and humanitarian efforts. Employing RGB or spectral images with CNNs and recurrent neural networks (RNNs) enables timely and precise crop yield predictions [26,27]. In a study by Oliveira et al. [28] three detection algorithms, namely Faster R-CNN, SSD, and SSDLite, were compared for predicting cotton yield. Among these algorithms, SSD achieved the lowest mean percentage error of 8.84%. Another research by Nevavuori et al. [29] investigated the impact of network parameters and architectures on prediction error using RGB and NDVI images. The RGB-trained models exhibited superior accuracy, with an error rate of 8.8%, particularly during the early growth stages, compared to the NDVI-trained models. Chu et al. [30] proposed an end-to-end CNN-RNN model to predict summer and winter rice crop yields. Their approach fused spatial features from neural networks with temporal features. The model converged quickly and demonstrated superior prediction performance compared to only temporal features, yielding excellent results for both summer and winter rice yields.

Among precision agriculture tasks, we focus on weed mapping, a segmentation task.

2.1. Semantic segmentation using deep neural networks

Semantic segmentation is a computer vision task that aims to infer a class for each pixel in the image. Recently, deep learning techniques have been shown to significantly improve this task over more classical techniques. Fully-convolutional networks (FCNs) are one of the most promising ways to solve this task [31]. Classical used FCNs are SegNet [32] and U-Net [33]. Zhao et al. [34] addressed the problem of capturing multi-scale information with the proposed pyramid scene parsing network (PSPNet). Chen et al. [35] proposed a new architecture called DeepLabv3+ that uses atrous spatial pyramid pooling (ASPP) and an encoder–decoder structure. ASPP is a technique that uses multiple parallel atrous convolutional layers with different dilation rates to capture multi-scale information. While CNNs were traditionally favored, Transformer-based architectures [11] have emerged as game-changers. By leveraging self-attention, Transformers capture long-range dependencies and contextual information, yielding more accurate and coherent segmentation. Their ability to efficiently process variable-sized inputs and handle high-resolution images, coupled with extensive pre-training, has significantly contributed to their growing popularity. SegFormer brought Transformers into semantic segmentation using a hierarchical encoder suited for the task [36]. Instead, the Dense Prediction Transformer maintains a high resolution to provide fine-grained segmentation [37].

2.2. Weed mapping using deep neural networks

Weed mapping is a particular type of semantic segmentation task. The work of dos Santos et al. [8] was one of the first to show the excellent results of CNNs, particularly AlexNet, compared to classical machine learning approaches such as SVM and Random Forest. Lottes et al. [38] used a CNN with two decoders, one to detect stem position and the other to segment the plant. Two datasets were used for evaluation, the BoniRob dataset and one collected with a UAV. They achieved an mAP of 79.2% and 75.3% for stem detection and 83.8% and 87.3% for segmentation, respectively. In [10], SegNet with ResNet50 as encoder was used, achieving an F1 score of 64.6%. Additionally, in [39], U-Net was applied to the CWFID dataset [40], achieving an F1 score of 89% and a mIoU of 98%. Another work on CWFID [41] achieved a mIoU of 71% using U-Net with data augmentation techniques.

2.3. Weed mapping using multispectral images and UAVs

Depending on the bands acquired, multispectral images may contain information related to a plant's growth and health status and its species. Therefore, they can improve the accuracy of deep learning models compared to models trained only with RGB. In addition, sensors for multispectral image acquisition can be easily used with UAVs.

In [39], the popular U-Net was used on a set of data available on the Internet to separate weeds from crops and soil, achieving an F1 score of 89% and a mIoU of 98%. Sa et al. [9] developed WeedNet, a semantic segmentation network based on SegNet, and trained it on a collected dataset, achieving an F1 score of 80%. This publicly released dataset was called WeedMap and includes two sets of images of sugar beet fields collected in Germany (Rheinbach) and Switzerland (Eschikon). Both were collected with UAVs equipped with multispectral cameras. The first used a 5-channel RedEdge-M camera, and the second a 4-channel Sequoia camera. The authors also trained SegNet

with multiple combinations of the acquired channels, obtaining an AUC of 84.3% [13]. In [42], the DeepLabv3 architecture for semantic segmentation was compared to SegNet and U-Net, on the WeedMap dataset, achieving, in particular, an F1 score of 81% on the Rheinbach subset. On the same dataset, also using the Eschikon subset, Mozzam et al. [43] used a patch-based training with a modified VGG model. Patches were selected manually, and those containing both classes were dropped. The results showed 92% accuracy on the Rheinbach subset and 90% accuracy on the Eschikon subset. Khoshboresh-Masouleh and Akhoondzadeh [44] also attempted to improve weed mapping using this dataset. However, they used another manual train–test split made at the field level. Their proposed model, called DeepMultiFuse, was purpose-built for weed segmentation and is based on dilated convolution, a modified inception module, a fusion module, and a gated encoder–decoder architecture. In their setting, they achieved a mIoU of 83% on the Rheinbach subset and 97% on the Eschikon subset. Since WeedMap has become a dataset of choice in several papers because of its volume and quality, we also used it for benchmarking purposes.

As noted, there is active research in weed mapping employing deep neural networks and UAVs. To our knowledge, this work is the first attempt to use a Vision Transformer for this task, which also uses a transfer learning approach based on an RGB-trained model in a multispectral setting.

3. Materials

In this section, we describe the dataset we considered and the additional preprocessing and data augmentation we applied.

3.1. Dataset

To address the problem of distinguishing weeds from crops, we considered the dataset proposed in [13], which contains orthomosaic maps of sugar beet fields, specifically *Beta vulgaris* of the "Samuela" variety. Considering also the background, in total, there are three classes: background, crop, and weed. It is worth noting that despite the limited number of classes, the dataset used in this study has a level of complexity comparable to larger reference datasets for semantic segmentation, such as Cityscapes [45], which contain dozens of different classes and scenarios. The intricacy of the task stems from the subtle distinctions between crop and weed classes and the limited number of examples, which highlights the need to use a pre-trained model or incorporate additional techniques, such as data augmentation, to improve the performance of the segmentation.

As mentioned earlier, the dataset is divided into two main subsets: one related to fields located in Germany (Rheinbach); the other related to fields located in Switzerland (Eschikon). In particular, the dataset presents eight orthomosaic maps, [000, 001, 002, 003, 004, 005, 006, 007], the first five in the Rheinbach subset and the last three in the Eschikon subset. Each orthomosaic map corresponds to a set of tiles. Multiple tiles were derived from each orthomosaic map by sliding a window over the maps. Each tile has a size of 480×360, for a total of 971 tiles for the Rheinbach subset and 700 for the Eschikon subset.

The authors performed data acquisition using two UAVs: a DJI Inspire2, equipped with a RedEdge-M camera; and a DJI Mavic Pro, equipped with a Sequoia camera. The former was used for the first location (Rheinbach), and the latter for the second location (Eschikon). The drones flew at a cruising speed of ~4.8 m/s and an altitude of 10 m. Crop plants are around 15–20 pixels in size, and individual weeds occupy 5–10 pixels: this makes the segmentation task particularly challenging. The RedEdge-M camera can acquire five raw image channels: R, G, B, Near Infrared (NIR), and Red Edge (RE). The Sequoia camera acquires the same channels except for the blue channel. From the R and NIR channels, the Normalized Difference Vegetation Index (NDVI), calculated as follows, can also be considered:

$$\mathrm{NDVI} = \frac{\mathrm{NIR} - \mathrm{R}}{\mathrm{NIR} + \mathrm{R}} \tag{1}$$

It is an indicator of vegetation cover, so it is beneficial for distinguishing plants from soil and can be used with the other channels. We also employed the Color Infrared (CIR) color combination, which consists of the NIR, R, and G channels.

3.2. Data preprocessing

Although the dataset has already been thoroughly processed by the authors [13], further preprocessing is necessary. First, since the orthomosaic maps are not rectangles, they have some black areas at the borders, which generate many totally black tiles. As a first preprocessing step, these tiles have been removed, reducing the dataset to 557 tiles for the Rheinbach subset and 561 for the Sequoia subset. In addition, the height of each tile of 360 is quite problematic because it needs to be divisible by $2^i, i > 3$ as some convolutional filters would require. For this reason, four crops of size 256×256 have been extracted from each image. This also reduces the computational load.

3.3. Data augmentation

The authors of the dataset applied a random *horizontal flip* in their experiments. In most tasks, this augmentation is applied to the training data as it does not denature the image, unlike the *vertical flip*, which overturns the image. However, having images acquired from the nadir direction allows us to use the vertical flip without problems. For the same reason, it is also possible to apply *random rotations* of the image, sampling for the full range of degrees from 0 to 360. In this case, we applied a "selective" random rotation to cope with data imbalance. It consists of applying data augmentation only to those examples that contain at least one pixel in the minority class, in our case weed. With this technique, it is possible to increase the number of images that contain weeds, which can help the model learn the minority class better. However, although it seems counterproductive when counting at the pixel level, the number of pixels in the minority class is reduced because the examples containing weeds are unbalanced. This last step was applied only to the Eschikon subset, where images containing weed account for only 0.166% of the dataset, which was increased to 0.499%. In the Rheinbach subset, the percentage is already 0.706%.

4. Methods

The methods we propose for weed mapping are inspired by a recently proposed network architecture, Lawin [46]. Lawin is a Vision Transformer suitable for semantic segmentation that has achieved state-of-the-art results on the benchmark datasets Cityscapes, ADE20K, and COCO-Stuff. However, Lawin cannot be directly applied as is. Weed mapping, like other precision agriculture tasks, benefits from some bands of the non-visible spectrum, particularly the NIR and RE bands. This hampers the application of deep learning models, typically suited to be fed with RGB images. To handle other channels besides RGB, specific architectural designs are required. Below, we describe simple modifications applied to the basic Lawin architectures, as well as two new variants we propose that can handle both RGB and non-visible channels.

4.1. Lawin modified for weed mapping

Like other semantic segmentation models, Lawin consists of an encoder and a decoder. The encoder is a Mix Transformer (MiT) [36], an architecture for semantic segmentation designed as a specific alternative to the original Vision Transformer (ViT) [11] and designed to take RGB images as input. However, unlike ViT, MiT can generate CNN-like multi-level features with different resolutions, providing a feature map for each Transformer block as output. This hierarchical representation provides high-level coarse-grained and low-level fine-grained features that usually increase performance in semantic segmentation.

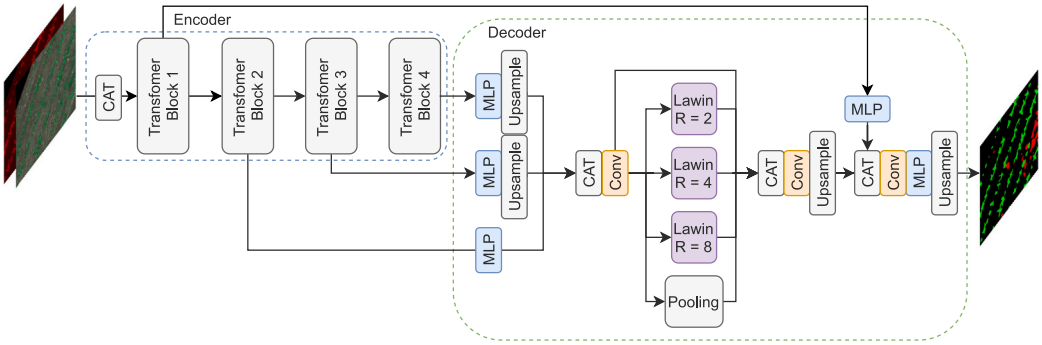

Fig. 1. Lawin. "CAT" is the concatenation layer, "Conv" is a convolutional layer with a 3 × 3 kernel and "MLP" stands for fully-connected layer. Unlike the original version, the model accepts both visible and non-visible channels as input.

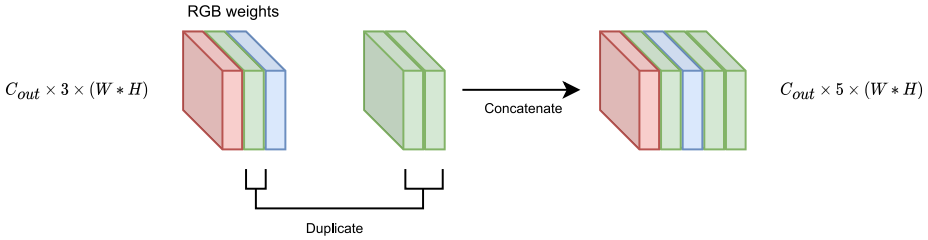

Fig. 2. The proposed weight loading strategy. In this example, the RGB weights are adapted for the (R, G, B, NIR, RE) input reusing the green related weights. The weights refer to a convolution layer and are represented as a 3D tensor collapsing width and height into a single dimension for visualization purposes.

Starting from an RGB image of size $3 \times H \times W$, the first Transformer block produces a feature map of size $C_1 \times \frac{H}{4} \times \frac{W}{4}$, where C_1 is a selected embedding dimension. Then, each subsequent Transformer block, taking the feature maps of the previous block as input, produces a feature map $F_i \in \mathbb{R}^{C_i \times \frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}}}$.

The decoder uses a Large Window Attention Spatial Pyramid Pooling (LawinASPP), consisting of five parallel branches, a pooling layer, a shortcut connection, and three large window attentions with different context sizes. The pooling branch functions as global context, while the three window attentions as local context extractors. LawinASPP is applied to the concatenation of the last three outputs of the decoder. The last two are adapted using a standard multilayer perceptron and an upsampling operation. The first output of the encoder, instead, skips the LawinASPP and is concatenated to its output. A final linear transformation, followed by an upsampling operation, is then applied to build the final segmentation map in the form of a probability distribution over the classes for each pixel.

Building on Lawin's basic architecture, in this paper, we propose a first simple variant specifically designed to solve the weed mapping problem (see Fig. 1). To handle additional input channels, we modified the first convolutional layer so that it can accept not only visible channels but also different combinations of visible and non-visible channels. Different strategies can then be adopted to cope with non-RGB channels. A first, simple one is to load pre-trained RGB weights as is. The model has learned to perform segmentation when fed with RGB inputs; starting from this configuration, the model must then refine its weights with a different combination of channels. However, this strategy can no longer be applied when a different combination involving more than three channels is to be used. In this case, the weights for the additional inputs must be initialized randomly or copied from existing ones. To provide the model with a more informed starting point, we followed the second strategy, based on selecting one of the three RGB channels and transferring the relative weights for each extra channel. One possible criterion for this choice is to consider the similarity of reflectance in plants [47]. Following this criterion, we selected the G channel, which has the highest reflectance in the visible spectrum. With this strategy (shown in Fig. 2), we can effectively transfer the knowledge and capabilities gained from RGB imagery to the multispectral domain, eliminating the need for pre-training on a broad multispectral dataset. However, in both strategies, this way of fitting the model to these extra inputs can lead to a phenomenon called *catastrophic forgetting*. The encoder weights have been trained to be used on RGB inputs, so the information stored by the weights could be "forgotten" during fine-tuning.

4.2. DoubleLawin

To explore a strategy in which the input channels are not considered together, we propose a more refined model called *DoubleLawin*. It consists of two parallel MiT encoders that work independently to extract features from the visible and non-visible channels. Its architecture is depicted in Fig. 3. The first encoder is fed with visible channels and inherits the weights already pre-trained on RGB images. These weights can be kept constant during training to avoid catastrophic forgetting. Depending on the non-visible channels to be used, the second encoder, a copy of the first, is also supplied with weights pre-trained on RGB images. Relying on the criterion of vegetative reflectance, as before, the weights are copied from the G channel.

To combine the features extracted by the two encoders, we introduce three different *fusion blocks*, from the simplest to the most complex. The first one is inspired by the fusion block proposed in [48] and consists of a summation preceded by a point-wise convolution. This operation is applied to each pair of encoder outputs and maintains the same number of channels. The output of the fusion block is a fused feature map F_i computed as:

$$F_i = \phi(X_{1i}) + \phi(X_{2i}) \tag{2}$$

where $\phi(\cdot)$ is the point-wise convolution, and X_{1i} and X_{2i} are the outputs from the two encoders at the ith stage. Note that the feature maps F_i, X_{1i} and X_{2i} share the same size S_i:

$$S_i = C_i \times \frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}} \tag{3}$$

being C_i the number of channels in the feature map and (H, W) the size of the input image. The purpose of the fusion block is to transform

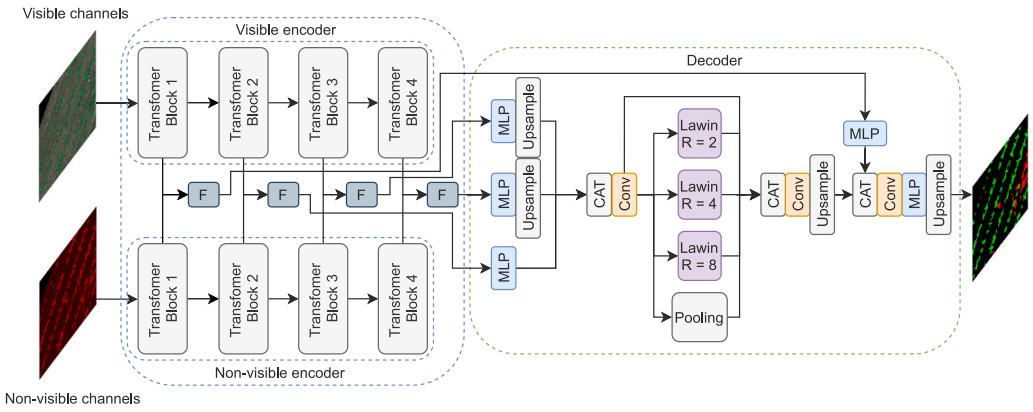

Fig. 3. DoubleLawin. "F" stands for "fusion". Each output for each block of the two encoders is fused to obtain the features that constitute the final output of the overall encoder. Different MLPs then transform the different feature maps in the decoder.

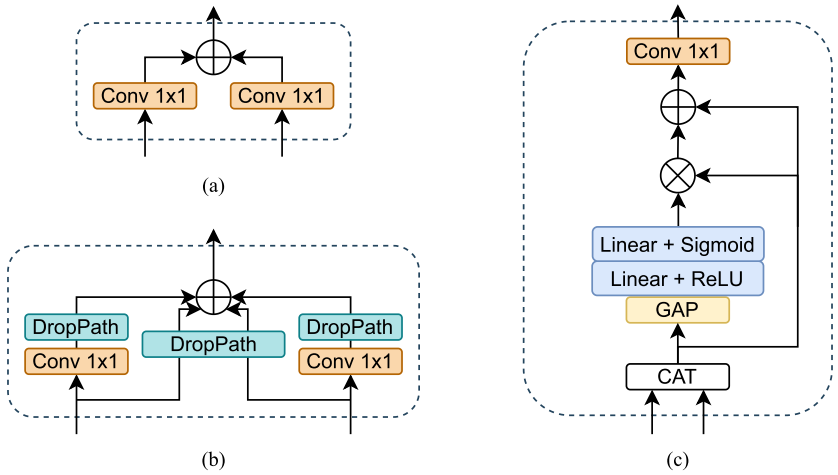

Fig. 4. Fusion block with convolution only (a), with DropPath (b), and Squeeze-and-Excitation fusion (c). "GAP" stands for "Global Average Pooling".

the channel information so that it is weighted for the subsequent summation. We adopt a point-wise convolution because it is much cheaper than a spatial convolution and the aggregation of spatial information is not necessary for the fusion block as the goal is a depth-wise fusion (Fig. 4(a)).

The simultaneous use of two encoders may cause overfitting because of the *co-adaptation* problem, which refers to the highly correlated behavior of neurons in a neural network. When multiple branches of neurons are fused, in fact, they tend to co-adapt; in particular, one path is used as an anchor, the other as a corrective term. This configuration is prone to overfitting. One possible solution is using a DropPath layer [49] that randomly drops operands in the joint layers; in this way, each branch of the network learns features independently of the others, discouraging co-adaptation. In particular, we use the *local* DropPath that applies the drop with a fixed probability p to each joint layer independently. In case no path survives, a residual connection is chosen, inspired by the so-called Stochastic Depth proposed in [50]. This is the second proposed fusion block (Fig. 4(b)). Eq. (2) becomes:

$$F_i = DP(\phi(X_{1i})) + DP(\phi(X_{2i}) + DP_e(X_{1i}, X_{2i})) \tag{4}$$

Here, DP stands for DropPath and DP_e is the DropPath that guarantees that at least one input is retained. In this work, the DropPath survival probability p for the convolutional paths was set to 0.9, while the probability for the residual connection paths was set to 0.5 to reduce the number of training steps in which both paths were active.

The third fusion block takes inspiration from Squeeze-and-Excitation networks [51]. The Squeeze-and-Excitation block (Fig. 4(c)) allows recalibrating channel-wise feature responses by explicitly modeling channel interdependencies. The block consists of two operations: a squeeze operation, which aggregates the spatial information of the feature map into a channel descriptor, and an excitation operation, which uses the channel descriptor to generate a set of channel-wise weights. The channel-wise weights are then used to re-weight the feature map. Thus, by concatenating the two feature maps before feeding them into the Squeeze-and-Excitation block, we can re-weight the features extracted by the two encoders considering the relationship between the two feature maps. After the re-weight operation, we return to the original number of channels by applying a point-wise convolution.

4.3. SplitLawin

The DoubleLawin architecture treats the non-visible channels (without pre-trained weights) as separate from the RGB channels and extracts features from the former while keeping the features extracted from the latter fixed. However, learning features separately may prevent the discovery of joint features related to exploiting the interaction of different channels. To account for this issue, we experimented with an additional variant, called *SplitLawin* (Fig. 5). Instead of adding another encoder, we take advantage of the block-based MiT architecture and duplicate only the first block. In this way, only the low-level features will be learned separately, allowing visible and non-visible channels to interact in the higher-level spaces. Since only the first block is duplicated, considering MiT-B0 as an encoder basis and three input channels, SplitLawin is lighter than DoubleLawin with 5.25M parameters versus 8.44M.

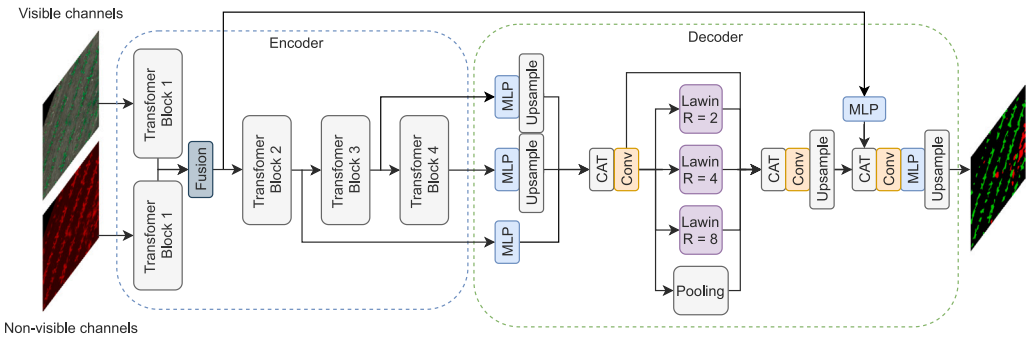

Fig. 5. SplitLawin.

Table 1
Channels used (note that the blue channel is only present in the Rheinbach subset).

Lawin	Double/SplitLawin
NDVI	R, G, (B), NDVI
R, G, (B)	R, G, (B), NIR
CIR	R, G, (B), RE
R, G, (B), NIR, RE	R, G, (B), NIR, RE

5. Experiments

This section presents our experimental setup, followed by the quantitative and qualitative results of crop/weed segmentation.

5.1. Experimental setup

We used the same train–test split applied in [13,42], that is [000, 001, 002, 004]–[003] for the Rheinbach subset and [006, 007]–[005] for the Eschikon subset. We used Adam as an optimizer for training the models, with batch size 6, a maximum number of epochs of 500, and early stopping with patience 10. Specifically, validation sets were randomly drawn from the training sets for early stopping. We used the focal loss as a loss function, weighted by the pixel class frequency:

$$FL = -w_c(1 - f(x)_c)log(f(x)_c) \tag{5}$$

where $f(x)_c$ is the probability of the true class predicted by the model and w_c the corresponding class weight, calculated as follows:

$$w_c = \frac{FoA(c)}{\widetilde{FoA(c)}} \tag{6}$$

where $FoA(c) = \frac{I_c}{I}$, $\widetilde{FoA(c)}$ is the median of $FoA(c)$ by varying c, I_c is the number of pixels in c and I is the total number of pixels.

The chosen combinations of channels are reported in Table 1. We also grid-searched over the different versions of the MiT encoder, B0 and B1, which differ only in the number of parameters. To train the models, we used two methods: fine-tuning the pre-trained weights; or freezing them. All fusion blocks were tested. For better readability, we do not report the results obtained with every possible hyperparameter configuration: we show only the best-performing hyperparameters for each model and backbone in Table 2. All tested configurations and their results can be found in the supplementary materials of this paper. The experiments were performed on a GTX 1660 Ti with 6 GB of VRAM.

We used the macro-averaged F1 score to give a quantitative evaluation of the models, calculated as:

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{7}$$

where TP stands for true positives, FP for false positives, and FN for false negatives. We also calculated AUC for comparison with other works; however, it is worth noting that it is not as representative as the F1 score for highly unbalanced datasets as in our case.

5.2. Ablation studies

As shown in Fig. 6, which concerns ablation studies we performed on the Rheinbach subset, each Lawin variant almost doubles the time per example when using MiT-B1 compared to MiT-B0, with a slight increase in F1 score, except for Lawin. This means that a lighter model could reach or exceed these scores with less inference time. Considering MiT-B0 as a backbone, Lawin is the lightest model, with 14 ms per example, while DoubleLawin needs 17 ms. SplitLawin, which duplicates only the first block instead of the entire encoder, needs 15 ms. In any case, all three of these variants are extremely light. In contrast, the baseline SegNet needs 25 ms. The calculated GFlops, as can be seen in Table 5, reflect the measured time for example.

DropPath for Double and SplitLawin has no impact on inference time as it is only applied in the training phase. Regarding the Squeeze-and-Excitation fusion block, the difference in terms of complexity is negligible. It is notable from Table 3 that, on the Rheinbach subset, all three fusion blocks obtain similar results, with the Squeeze-and-Excitation fusion having slightly better F1 for SplitLawin-B0. However, on the Eschikon subset, the most straightforward fusion block, based on convolution only, achieves the best performance, while the Squeeze-and-Excitation fusion block performs worse across all models.

In addition to the performance comparison of the fusion blocks, Table 4 presents the performance comparison between runs conducted without and with RGB pre-training on non-visible related weights. The results demonstrate that employing RGB pre-trained weights on multispectral channels enhances performance across all configurations except for the Lawin algorithm on the Eschikon subset. However, it is noteworthy that Lawin achieves the best result on the Eschikon subset by utilizing the NDVI channel with R-related weights, thereby reaffirming the effectiveness of this strategy.

5.3. Comparison with the state-of-the-art

The best results, compared with the models that currently hold the state-of-the-art, are reported in Table 5. DeepMultiFuse [44] is not included as the experiments conducted were based on a non-reproducible train–test split. In fact, the authors used a manual splitting within each field, which may have also carried the risk of introducing a bias, as the splitting is no longer at the field level. Our proposed models outperformed SegNet on both data subsets as well as DeepLabv3 on Rheinbach (the only subset used in that work). The proposed models also exceeded the state-of-the-art when considering AUC, the primary metric reported in previous work. SplitLawin achieved the best overall F1 score on the Rheinbach subset, while Lawin outperformed the other models on Eschikon. When AUC is considered, DoubleLawin-B1 obtained the best results (on Eschikon). However, the results on Eschikon may be less reliable because it contains only a few images of weeds in the test set compared to Rheinbach.

Regarding computation time, the proposed models with the B0 variant are faster even than DeepLabv3, whose calculated inference

Table 2
Hyperparameters used for the best-performing runs (Conv = fusion block with convolution only; Drop = with DropPath; SE = Squeeze-and-Excitation fusion).

Subset	Model	Backbone	Weights	Channels	Strategy	Fusion
Rheinback	Lawin	MiT-B0	G, G, R	CIR	Fine-tuning	–
	Lawin	MiT-B1	G, G, R	CIR	Fine-tuning	–
	DoubleLawin	MiT-B0	R, G, B, G	R, G, B, NDVI	Fine-tuning	Conv
	DoubleLawin	MiT-B1	R, G, B, G	R, G, B, NIR	Fine-tuning	Drop
	SplitLawin	MiT-B0	R, G, B, G, G	R, G, B, NIR, RE	Fine-tuning	SE
	SplitLawin	MiT-B1	R, G, B, G, G	R, G, B, NIR, RE	Fine-tuning	SE
Eschikon	Lawin	MiT-B0	R, G	R, G, NIR, RE	Fine-tuning	–
	Lawin	MiT-B1	R	NDVI	Freeze	–
	DoubleLawin	MiT-B0	R, G, G	R, G, NDVI	Freeze	Conv
	DoubleLawin	MiT-B1	R, G, G	R, G, NDVI	Freeze	Drop
	SplitLawin	MiT-B0	R, G, G	R, G, NDVI	Fine-tuning	Drop
	SplitLawin	MiT-B1	R, G, G	R, G, NDVI	Fine-tuning	Drop

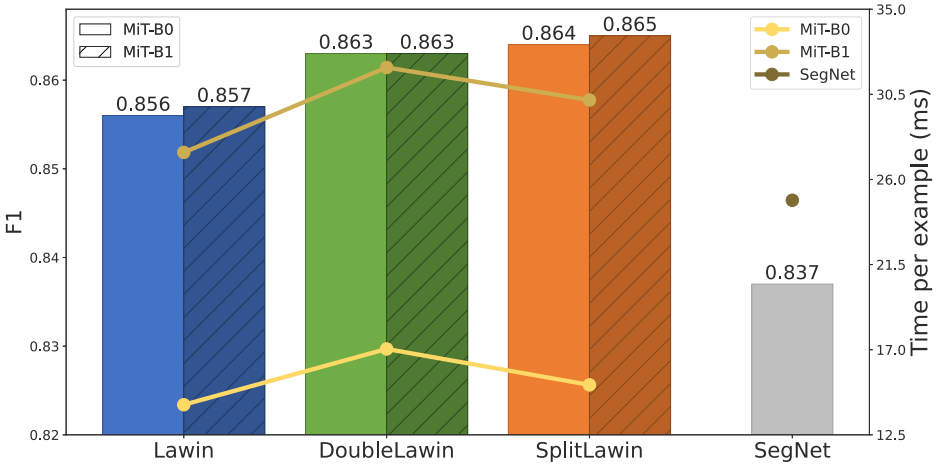

Fig. 6. Comparison of F1 score (bar chart) on the Rheinbach subset for each model and backbone with relative time per example (line chart). SegNet's F1 score was obtained by reproducing the experiments in [13].

Table 3
Comparison of different fusion blocks (Conv = fusion block with convolution only; Drop = with DropPath; SE = Squeeze-and-Excitation fusion).

Model	Rheinbach			Eschikon		
	Conv	Drop	SE	Conv	Drop	SE
DoubleLawin-B0	0.863	0.863	0.857	0.601	0.540	0.534
DoubleLawin-B1	0.862	0.863	0.857	0.499	0.580	0.557
SplitLawin-B0	0.859	0.862	0.864	0.565	0.635	0.592
SplitLawin-B1	0.865	0.862	0.865	0.571	0.630	0.584

Table 4
Comparison of random weights with pre-trained weights on non-visible related channels.

Model	Rheinbach		Eschikon	
	Random	Pre-trained	Random	Pre-trained
Lawin-B0	0.753	0.856	0.663	0.499
Lawin-B1	0.767	0.855	0.610	0.493
DoubleLawin-B0	0.851	0.863	0.519	0.601
DoubleLawin-B1	0.850	0.863	0.541	0.580
SplitLawin-B0	0.862	0.864	0.582	0.635
SplitLawin-B1	0.865	0.865	0.571	0.630

time, assuming ResNet50 as the encoder, is 26.60 ms on our hardware. In addition, the number of GFlops, a hardware-independent metric, also shows that the proposed models are more efficient than DeepLabv3.

5.4. Qualitative evaluation

High F1 scores reflect very accurate predictions whose errors are hardly visible to the naked eye, as these are errors related to imperfect segmentation rather than misclassifications among whole plants. Therefore, it becomes challenging to distinguish the predictions of the three architectures. However, there are instances where crops are mistakenly labeled as weeds. For instance, in the second sample of Fig. 7, located at the center bottom of the image, Lawin misclassifies a crop as a weed, whereas SplitLawin and DoubleLawin accurately identify it. On the other hand, SplitLawin and DoubleLawin tend to overlook more weeds, classifying them as background.

Furthermore, regarding SplitLawin, 20.8% of weed pixels are misclassified as background, while 1.4% are misclassified as a crop. We argue that weed mapping does not require perfect segmentation maps since the goal is to spray herbicide. Conversely, 11.8% of the crop is misclassified as background, and 1.7% of the crop is assigned to the weed class. For the same reason mentioned above, 11.8% is not a big problem, but spraying herbicide on 1.7% of the crop would result in significant crop yield losses. Models trained for this task should be pushed to have a crop/weed error close to zero.

Table 5
Comparison of our models with the state-of-the-art.

Model	F1		AUC		GFlops
	Rheinbach	Eschikon	Rheinbach	Eschikon	
SegNet [13]	–	–	0.828	0.843	80.52
DeepLabv3 [42]	0.837	–	0.880	–	82.00
Lawin-B0	0.856	0.663	0.959	0.977	**4.06**
Lawin-B1	0.857	**0.685**	0.929	0.970	15.34
DoubleLawin-B0	0.863	0.601	**0.964**	0.972	5.18
DoubleLawin-B1	0.863	0.580	0.952	**0.987**	19.76
SplitLawin-B0	0.864	0.635	0.957	0.970	4.28
SplitLawin-B1	**0.865**	0.630	0.928	0.976	16.20

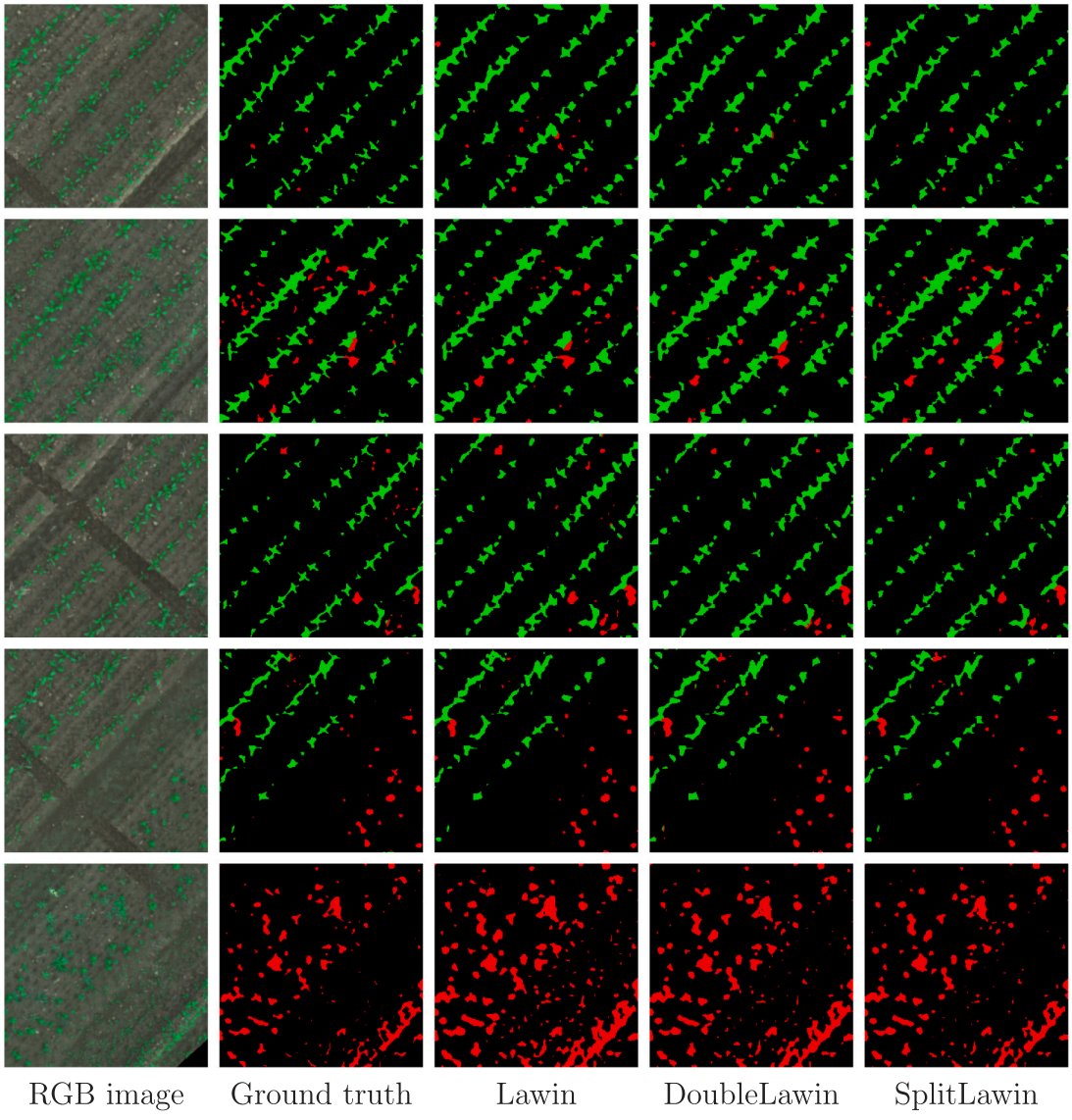

Fig. 7. Examples of segmentation performed on the [003] test field (black = background, green = crop, red = weed).

Examples of segmentation maps obtained as output by the best-performing run of Lawin, DoubleLawin, and SplitLawin are shown in Fig. 7.

5.5. *Generalization over different fields*

Generalization across various species of crops and weeds poses a significant challenge in weed mapping. The Eschikon subset consisted of three fields with crops that exhibited slight variations in shape. Notably, these crops were smaller in size in fields 005 and 007 while bigger in field 006, as depicted in Fig. 8. Furthermore, the terrain color is slightly different. Consequently, the Eschikon subset serves as a valuable test for evaluating generalization. Despite this, we assessed the ability of our proposed models to generalize also by training them on the Eschikon subset and subsequently testing them on the Rheinbach subset (field 003). We used the best-performing model on the Eschikon subset, obtaining a macro-F1 score of 0.731, which is lower than the models trained on Rheinbach (0.865) but higher than its performance on Eschikon (0.685) showing a good generalization capability over fields in different areas.

6. Conclusion

In this work, we proposed a novel approach based on lightweight Vision Transformers to perform weed mapping from drones. The results show that this type of model can achieve state-of-the-art performance while maintaining an acceptable inference time, making them suitable for running on mobile platforms such as drones. We also demonstrated that weights pre-trained on RGB images can be effectively used in a multispectral context with proper fine-tuning. In particular, the proposed SplitLawin is the model that best transfers the knowledge of RGB weights to a multispectral setting, as it obtained the best F1 score on the Rheinbach subset with an addition of only 1 ms in the inference time compared to the basic Lawin.

Building on the encouraging results obtained, several future works could be addressed. The first is constructing a region-aware metric to evaluate the model's ability to make herbicide spraying efficient and non-hazardous to crops. Current metrics, such as F1 score, Jaccard index, or accuracy, cannot highlight this. Similarly, a custom loss could be used to guide network learning better. In addition, an attention mechanism can be implemented that can capture row arrangement

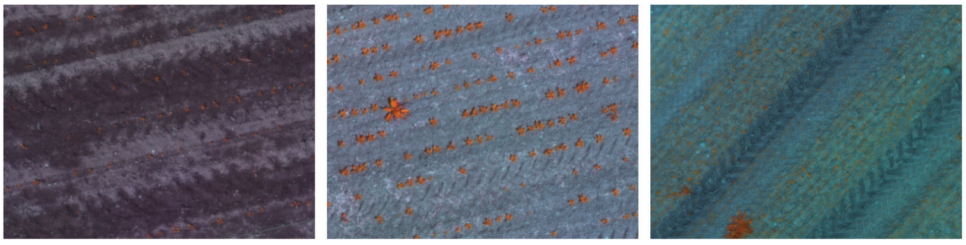

Fig. 8. Samples drawn from the three fields in Eschikon in Color Infrared, respectively 005, 006, 007. It can be seen that the terrain has different colors and the crops slightly different shapes.

information in crops. Injecting information about the arrangement of plants in fields can make it easier for the model to detect weeds. This can be added directly to the model or be a post-processing technique applied to the segmentation map. Another future work is to integrate such models into a fully autonomous system in which navigation over large fields is guided by a vision-based learning method [52]. Finally, it is worth noting that the shape of crops and weeds changes over time, and our current model does not explicitly account for time-varying properties. This future research may provide a more robust solution.

Developing effective and efficient computer vision algorithms on drones can increase the confidence and use of this technology in precision agriculture. In particular, the proposed approach has significant implications for precision agriculture, allowing farmers to quickly and easily identify infested areas and prioritize control efforts. This can significantly improve agricultural management practices, leading to more sustainable and efficient agriculture.

CRediT authorship contribution statement

Giovanna Castellano: Conceptualization, Writing – review & editing, Supervision. **Pasquale De Marinis:** Conceptualization, Methodology, Software, Investigation, Writing – original draft, Visualization. **Gennaro Vessio:** Conceptualization, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data are publicy available [13].

Acknowledgments

The research of Pasquale De Marinis is funded by a Ph.D. fellowship within the framework of the Italian “D.M. n. 352, April 9, 2022” - under the National Recovery and Resilience Plan, Mission 4, Component 2, Investment 3.3 - Ph.D. Project “Computer Vision techniques for sustainable AI applications using drones”, co-supported by “Exprivia S.p.A” (CUP H91I22000410007).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.neucom.2023.126914.

References

[1] FAO, How to Feed the World in 2050. Insights from an Expert Meet, FAO, 2009.
[2] S.G. Vougioukas, Agricultural robotics, Annu. Rev. Control Robot. Auton. Syst. 2 (2019) 365–392.
[3] M. Burke, A. Driscoll, D.B. Lobell, S. Ermon, Using satellite imagery to understand and promote sustainable development, 2020, CoRR abs/2010.06988. arXiv:2010.06988.
[4] G. Castellano, E. Cotardo, C. Mencar, G. Vessio, Density-based clustering with fully-convolutional networks for crowd flow detection from drones, Neurocomputing (2023).
[5] N. Passalis, A. Tefas, Deep reinforcement learning for controlling frontal person close-up shooting, Neurocomputing 335 (2019) 37–47.
[6] A.K. Singh, B. Ganapathysubramanian, S. Sarkar, A. Singh, Deep learning for plant stress phenotyping: Trends and future perspectives, Trends Plant Sci. 23 (10) (2018) 883–898.
[7] D. Wang, W. Li, X. Liu, N. Li, C. Zhang, UAV environmental perception and autonomous obstacle avoidance: A deep learning and depth camera combined solution, Comput. Electron. Agric. 175 (2020) 105523.
[8] A. dos Santos Ferreira, D.M. Freitas, G.G. da Silva, H. Pistori, M.T. Folhes, Weed detection in soybean crops using ConvNets, Comput. Electron. Agric. 143 (2017) 314–324.
[9] I. Sa, Z. Chen, M. Popović, R. Khanna, F. Liebisch, J. Nieto, R. Siegwart, Weednet: Dense semantic weed classification using multispectral images and mav for smart farming, IEEE Robot. Autom. Lett. 3 (1) (2017) 588–595.
[10] B. Hobba, S. Akıncı, A.H. Göktogan, Efficient herbicide spray pattern generation for site-specific weed management practices using semantic segmentation on UAV imagery, in: Australasian Conference on Robotics and Automation (ACRA-2021), 2021, pp. 1–10.
[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al., An image is worth 16x16 words: Transformers for image recognition at scale, 2020, arXiv preprint arXiv:2010.11929.
[12] C. Yan, Y. Hao, L. Li, J. Yin, A. Liu, Z. Mao, Z. Chen, X. Gao, Task-adaptive attention for image captioning, IEEE Trans. Circuits Syst. Video Technol. 32 (1) (2022) 43–51.
[13] I. Sa, M. Popović, R. Khanna, Z. Chen, P. Lottes, F. Liebisch, J. Nieto, C. Stachniss, A. Walter, R. Siegwart, WeedMap: A large-scale semantic weed mapping framework using aerial multispectral imaging and deep neural network for precision farming, Remote Sens. 10 (9) (2018) 1423.
[14] W. Zeng, M. Li, Crop leaf disease recognition based on self-attention convolutional neural network, Comput. Electron. Agric. 172 (2020) 105341.
[15] Y. Liu, S. Liu, J. Xu, X. Kong, L. Xie, K. Chen, Y. Liao, B. Fan, K. Wang, Forest pest identification based on a new dataset and convolutional neural network model with enhancement strategy, Comput. Electron. Agric. 192 (2022) 106625.
[16] T. Wiesner-Hanks, E.L. Stewart, N. Kaczmar, C. DeChant, H. Wu, R.J. Nelson, H. Lipson, M.A. Gore, Image set for deep learning: Field images of maize annotated with disease symptoms, BMC Res. Notes 11 (1) (2018) 440.
[17] K. Garg, S. Bhugra, B. Lall, Automatic quantification of plant disease from field image data using deep learning, in: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), 2021, pp. 1964–1971.
[18] R. Mittler, Abiotic stress, the field environment and stress combination, Trends Plant Sci. 11 (1) (2006) 15–19.
[19] S.S. Virnodkar, V.K. Pachghare, V.C. Patil, S.K. Jha, Remote sensing and machine learning for crop water stress determination in various crops: A critical review, Precis. Agric. 21 (5) (2020) 1121–1155.
[20] N.S. Chandel, S.K. Chakraborty, Y.A. Rajwade, K. Dubey, M.K. Tiwari, D. Jat, Identifying crop water stress using deep learning models, Neural Comput. Appl. 33 (10) (2021) 5353–5367.
[21] X. Feng, Y. Zhan, Q. Wang, X. Yang, C. Yu, H. Wang, Z. Tang, D. Jiang, C. Peng, Y. He, Hyperspectral imaging combined with machine learning as a tool to obtain high-throughput plant salt-stress phenotyping, Plant J. 101 (6) (2020) 1448–1461.

[22] K. Velumani, S. Madec, B. de Solan, R. Lopez-Lozano, J. Gillet, J. Labrosse, S. Jezequel, A. Comar, F. Baret, An automatic method based on daily in situ images and deep learning to date wheat heading stage, Field Crops Res. 252 (2020) 107793.
[23] J.G.A. Barbedo, Detection of nutrition deficiencies in plants using proximal images and machine learning: A review, Comput. Electron. Agric. 162 (2019) 482–492.
[24] A. Abdalla, H. Cen, L. Wan, K. Mehmood, Y. He, Nutrient status diagnosis of infield oilseed rape via deep learning-enabled dynamic model, IEEE Trans. Ind. Inform. 17 (6) (2020) 4379–4389.
[25] S. Rasti, C.J. Bleakley, G.C.M. Silvestre, N.M. Holden, D. Langton, G.M.P. O'Hare, Crop growth stage estimation prior to canopy closure using deep learning algorithms, Neural Comput. Appl. 33 (5) (2021) 1733–1743.
[26] T. Van Klompenburg, A. Kassahun, C. Catal, Crop yield prediction using machine learning: A systematic literature review, Comput. Electron. Agric. 177 (2020) 105709.
[27] A. Barbosa, R. Trevisan, N. Hovakimyan, N.F. Martin, Modeling yield response to crop management using convolutional neural networks, Comput. Electron. Agric. 170 (2020) 105197.
[28] D. Tedesco-Oliveira, R. Pereira da Silva, W. Maldonado, C. Zerbato, Convolutional neural networks in predicting cotton yield from images of commercial fields, Comput. Electron. Agric. 171 (2020) 105307.
[29] P. Nevavuori, N. Narra, T. Lipping, Crop yield prediction with deep convolutional neural networks, Comput. Electron. Agric. 163 (2019) 104859.
[30] Z. Chu, J. Yu, An end-to-end model for rice yield prediction using deep learning fusion, Comput. Electron. Agric. 174 (2020) 105471.
[31] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation.
[32] V. Badrinarayanan, A. Kendall, R. Cipolla, SegNet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (12) (2017) 2481–2495.
[33] U-Net: convolutional networks for biomedical image segmentation | springerlink. https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28.
[34] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2881–2890.
[35] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, H. Adam, Encoder-decoder with atrous separable convolution for semantic image segmentation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 801–818.
[36] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J.M. Alvarez, P. Luo, SegFormer: Simple and efficient design for semantic segmentation with transformers, in: Advances in Neural Information Processing Systems, Vol. 34, Curran Associates, Inc., 2021, pp. 12077–12090.
[37] R. Ranftl, A. Bochkovskiy, V. Koltun, Vision transformers for dense prediction, in: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Montreal, QC, Canada, 2021, pp. 12159–12168.
[38] P. Lottes, J. Behley, N. Chebrolu, A. Milioto, C. Stachniss, Joint stem detection and crop-weed classification for plant-specific treatment in precision farming, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2018, pp. 8233–8238.
[39] M.Á. Chicchón Apaza, H.M.B. Monzón, R. Alcarria, Semantic segmentation of weeds and crops in multispectral images by using a convolutional neural networks based on U-net, in: International Conference on Applied Technologies, Springer, 2019, pp. 473–485.
[40] S. Haug, J. Ostermann, A crop/weed field image dataset for the evaluation of computer vision based precision agriculture tasks, in: L. Agapito, M.M. Bronstein, C. Rother (Eds.), Computer Vision - ECCV 2014 Workshops, in: Lecture Notes in Computer Science, Springer International Publishing, Cham, 2015, pp. 105–116.
[41] A. Brilhador, M. Gutoski, L.T. Hattori, A. de Souza Inácio, A.E. Lazzaretti, H.S. Lopes, Classification of weeds and crops at the pixel-level using convolutional neural networks and data augmentation, in: 2019 IEEE Latin American Conference on Computational Intelligence (la-CCI), IEEE, 2019, pp. 1–6.
[42] W. Ramirez, P. Achanccaray, L.F. Mendoza, M.A.C. Pacheco, Deep convolutional neural networks for weed detection in agricultural crops using optical aerial images, in: 2020 IEEE Latin American GRSS & ISPRS Remote Sensing Conference (LAGIRS), IEEE, 2020, pp. 133–137.
[43] S.I. Moazzam, U.S. Khan, W.S. Qureshi, M.I. Tiwana, N. Rashid, W.S. Alasmary, J. Iqbal, A. Hamza, A patch-image based classification approach for detection of weeds in sugar beet crop, IEEE Access : Pract. Innov. Open Solut. 9 (2021) 121698–121715.
[44] M. Khoshboresh-Masouleh, M. Akhoondzadeh, Improving weed segmentation in sugar beet fields using potentials of multispectral unmanned aerial vehicle images and lightweight deep learning, J. Appl. Remote Sens. 15 (3) (2021) 034510.
[45] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3213–3223.
[46] H. Yan, C. Zhang, M. Wu, Lawin transformer: Improving semantic segmentation transformer with multi-scale representations via large window attention, 2022, arXiv preprint arXiv:2201.01615.
[47] E.B. Knipling, Physical and physiological basis for the reflectance of visible and near-infrared radiation from vegetation, Remote Sens. Environ. 1 (3) (1970) 155–159.
[48] P. Wang, C. Gao, Y. Wang, H. Li, Y. Gao, MobileCount: An efficient encoder-decoder framework for real-time crowd counting, Neurocomputing 407 (2020) 292–299.
[49] G. Larsson, M. Maire, G. Shakhnarovich, Fractalnet: Ultra-deep neural networks without residuals, 2016, arXiv preprint arXiv:1605.07648.
[50] G. Huang, Y. Sun, Z. Liu, D. Sedra, K.Q. Weinberger, Deep networks with stochastic depth, in: European Conference on Computer Vision, Springer, 2016, pp. 646–661.
[51] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.
[52] F. AlMahamid, K. Grolinger, Autonomous unmanned aerial vehicle navigation using reinforcement learning: A systematic review, Eng. Appl. Artif. Intell. 115 (2022) 105321.

Giovanna Castellano is an Associate Professor at the Department of Computer Science, University of Bari Aldo Moro, Italy, where she is the coordinator of the Computational Intelligence Lab. She is a member of the IEEE Computational Intelligence Society, the EUSFLAT society, and the INDAM-GNCS society. Her research interests are in the area of Computational Intelligence and Computer Vision. She is an Associate Editor of several international journals.

Pasquale De Marinis received his M.Sc. in Computer Science (2022) from the Department of Computer Science, University of Bari Aldo Moro, Italy, where he is currently a Ph.D. student. His current research interest is Drone Vision.

Gennaro Vessio received his Ph.D. in Computer Science and Mathematics (2017) from the Department of Computer Science, University of Bari Aldo Moro, Italy, where he is currently an Assistant Professor. His current research interests include Pattern Recognition, Machine and Deep Learning, and Computer Vision, and their application to several domains, including e-Health, Drone Vision, and Digital Humanities.