Automatic counting of mounds on UAV images: combining instance segmentation and patch-level correction

Majid Nikougoftar Nategh[1, 2], Ahmed Zgaren[1, 2], Wassim Bouachir[2], and Nizar Bouguila[1]

[1]Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada
[2]Department of Science and Technology, TÉLUQ University, Montreal, Canada

***Abstract*—Site preparation by mounding is a commonly used silvicultural treatment that improves tree growth conditions by mechanically creating planting microsites called mounds. Following site preparation, the next critical step is to count the number of mounds, which provides forest managers with a precise estimate of the number of seedlings required for a given plantation block. Counting the number of mounds is generally conducted through manual field surveys by forestry workers, which is costly and prone to errors, especially for large areas. To address this issue, we present a novel framework exploiting advances in Unmanned Aerial Vehicle (UAV) imaging and computer vision to estimate the number of mounds on a planting block accurately. The proposed framework comprises two main components. First, we exploit a visual recognition method based on a deep learning algorithm for multiple object detection by segmentation. This enables a preliminary counting of visible mounds, as well as other frequently seen objects (e.g., trees, debris, accumulation of water), to be used to characterize the planting block. Second, since visual recognition could be limited by several perturbation factors (e.g., mound erosion, occlusion), we employ a machine learning estimation function to predict the final number of mounds based on the local block properties extracted in the first stage. We evaluate the proposed framework on a new UAV dataset representing numerous planting blocks with varying features. The proposed method outperformed manual counting methods in terms of relative counting precision, indicating that it has the potential to be advantageous and efficient under challenging situations.**

***Index Terms*—Object counting, Mound detection, UAV imagery, Instance segmentation, Mask-RCNN, Computer vision, Precision forestry**

I. Introduction

Mechanical site preparation by mounding has been promoted and recognized as a popular technique in forest industry due to the abiotic and biotic characteristics of North American terrains. Mounding is also referred to a silvicultural technique for constructing elevated planting spots that are free of water logging, and with little vegetation competition in the soil [1]. A key issue after mounding is to precisely estimate the number of mounds created on each planting block, which corresponds to the number of tree seedlings to be planted.

Manual counting is a commonly used method in forest industry. To do so, forest workers count mechanically prepared mounds on a section of the site and use the result to estimate a total number for the entire site, assuming that mound density remains constant on a given plantation block. However, this approach is time-consuming, expensive, and prone to human error. Furthermore, mound density often varies on the same block depending on the characteristics of each zone. Motivated by recent advances in sensor technology used in drone platforms for data collection, forestry managers also used visual interpretation of Unmanned Aerial Vehicle (UAV) images as an alternative to field manual counting. Image interpretation and analysis are thus performed by human operators, in order to detect, identify, and count mounds on UAV orthomosaics. However, this requires a skilled human interpreter, and due to perception variation among humans on the nature of the objects, data quality, and scale from one site to the next, this method is often inefficient.

The purpose of our work is to develop a computer vision framework based on a UAV platform to make human work easier and more efficient. We propose a new method for mound counting a combination of two models: 1) local image segmentation using deep learning methods and 2) patch-level correction by applying regression models. First, the local segmentation model is trained on UAV images that have been manually annotated to detect and segment mounds, as well as other relevant objects, including trees, woody debris, and water accumulations. In fact, pixel-wise instance segmentation is used to count visual mounds and other objects of interest using local image segmentation as a first stage. The first processing step is essential in our framework, in order to characterize each image region by quantifying the presence of relevant object instances. Second, a patch-level correction model is applied to produce the final prediction of mound count, based on preliminary object count. Since mounds can be destroyed or occluded following their creation (e.g., due to erosion, the presence of debris, and trees), we cannot rely only on visual detection to estimate their number). Therefore, we formulate the task of counting mounds using a sequential approach that includes local pixel-wise object segmentation and patch-level correction.

We evaluated our framework on a dataset of UAV orthmosaics with different properties. The obtained results emphasize the importance of using both models sequentially.

The rest of this article is organized as follows. Section II

introduces background concepts and related works. Section III, provides a detailed description of the proposed framework. Experimental results are presented in section IV. Finally, section V concludes the paper.

II. Related works

Visual object counting, also known as crowd counting, is a computer vision task that encompasses all issues and challenges associated with estimating the number of times a specific object appears in an image [2]. Methods towards object counting in the literature can be divided into three categories based on the features used: traditional approaches, deep learning approaches, and hybrid approaches.

A. Traditional crowd counting approaches

Traditional crowd counting methods rely on hand-crafted features and are categorized into two categories based on the extracted feature category: direct detection methods and indirect detection methods.

Direct detection method (also known as detection-based method) has been widely used in many approaches [3], [4], [6], [7]. These approaches localize the position of each object in a single input image, and the number of detections is subsequently used as the crowd count. Traditionally, low-level features including Haar wavelets [4], histograms of oriented gradients (HOG) [3], edgelet [6], and shapelet [8] are used as region descriptors. Then a mainstream classifiers such as Support Vector Machine (SVM) [9], boosted trees [10] and random forests [11] is trained for classification. Finally, the number of object instances that the classifier produces on a test image is considered as the crowd count. Although detection-based methods have been effectively employed in low-density crowds, their performance decreases substantially when applied in high-density crowds with small and obscured objects, since they are based on low-level features.

Unlike direct techniques, object counting with indirect approach (also known as the feature-based method or regression method) starts by taking the entire crowd as an object, extracting local, global, and texture features of the crowd, and then establishing a mapping to the number of dense crowds to estimate the number of crowds indirectly [5]. These approaches have the advantage of not relying on the learning detector [5], and they are more effective because recognition is based on features rather than objects. However, regression-based counting directly maps from the extracted features of images to the number of objects, and because they ignore the object distribution information within the region, these methods do not have the ability to explicitly detect and localize each object [12]. Although density map regression-based approaches have achieved significant progress, they are still inadequate for real-world applications, particularly when large-scale variation is present. In addition, they are not capable of capturing semantic information since they rely on low-level features.

Traditional crowd counting methods are generally successful in moderately crowded scenes and are faster to process because they do not require considerable computational resources. However, they require a large number of training sets in order to have an effective training model. The applicability of these methods is also limited, and they are ineffective in dense crowds scenes with severe occlusion.

B. Deep learning approaches

Deep learning models have outperformed traditional machine learning approaches in recent years in the field of crowd counting. To learn and classify crowd regions of an image, deep learning algorithms rely on deep neural networks to extract semantic invariant features. Therefore, current research has shifted its focus to developing CNN-based techniques, as CNNs provide a more robust feature representation than the hand-crafted features utilized in traditional approaches.

1) Basic CNN approaches: These methods incorporate basic CNN layers as initial deep learning approaches for crowd counting. For counting people in highly dense crowds, Wang et al. [16] developed an end-to-end deep CNN regression model. In their architecture, they modified the original AlexNet network [17] by replacing the final fully connected layer with a single neuron to obtain an object count. Furthermore, training data augmented with additional negative samples are used to eliminate false responses in the backdrop of the images. Fu et al. [18] presented an optimized CNN approach based on the multi-stage ConvNet to estimate crowd density. Their optimization method is centered on removing some network connections based on the similar feature maps. Crowd images were then classified into one of five classes using two CNN cascade classifiers: very high density, high density, medium density, low density, and very low density.

2) Scale-aware CNN approaches: Single scale CNN models are less effective at accurately predicting density maps due to image scale variance. As a result of the foregoing constraint, basic CNN-based techniques evolved into more sophisticated models that were scale-resistant. To capture both high-level semantic information and low-level features, Boominathan et al. [19] combined deep and shallow fully convolutional networks to address crowd scales as well as perspective variations. With the adoption of CNN-based density map regression methods, more advanced CNN architectures by incorporating multi-column networks such as Multi-column CNN architecture (MCNN) [20], Switched CNN (Switch-CNN) [21], and Congested Scene Recognition Network (CSRNet) [22] have seen a remarkable improvement in performance.

3) Context-aware CNN approaches: Context-aware models were designed with the aim of reducing estimation errors by combining local and global contextual information into the CNN architecture. Using different coefficient weights, Sheng et al. [23] proposed a generalized variant of weighted VLAD. To do so, semantic information was incorporated into learning locality-aware feature (LAF) sets designed to investigate the spatial context and local information of crowds. In [24], a count estimation method based on an end-to-end CNN architecture was developed. Instead of partitioning the image into patches, the final crowd count outputs the entire image

in this method. As a result, the complexity is reduced due to the shared computations on overlapping regions achieved by integrating multiple stages of processing.

C. Hybrid approaches

Hand-crafted features and deep features are the two most common feature representations used in hybrid approaches. Lin et al. [25] proposed a low-cost method for counting people in videos. To train a small Local Binary Patterns (LBP) cascade classifier, the suggested architecture leverages a knowledge distillation strategy to transfer knowledge from a CNN object detector. Following the detection of people in video frames using YOLO [26], images of people with a confidence level greater than 30% are used to train the LBP cascade classifier, which is utilized to detect and track pedestrians. In [27], a combination of the two-stage detector was proposed to automate the task of detecting and counting the number of planting microsites using multispectral UAV imagery. Object proposals are firstly generated by applying a cascade detector based on LBP features. Then, candidate objects in the second stage were classified by a trained CNN network. Due to the fusion of various features and the use of more than one classifier, hybrid approaches in general may have an excessive computational cost.

III. Proposed Method

A. Motivations and overview

In this work, we aim to precisely count the number of mounds on each planting block represented by an orthomosaic. For each planting block, a batch of images captured using UAV is reconstructed to produce a high-resolution orthomosaic. We divided each orthomosaic into fixed cell sizes due to the high resolution of images, and used them as the input for our framework, as shown in Fig. 1. Visual inspection of different blocks (see Fig. 2) shows that the number of mounds varies from one patch to another. In fact, this variation is due to many factors, such as mechanical site specificities, environmental factors (dry, wet, and snow), and the presence of other objects, such as debris and trees (e.g., mound occlusion by debris, and appearance change due to tree shadows).

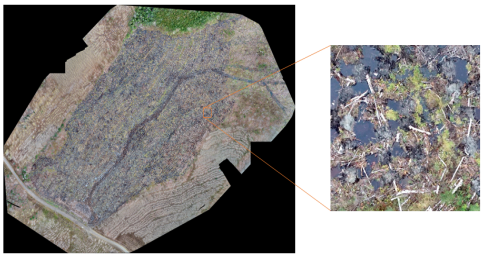
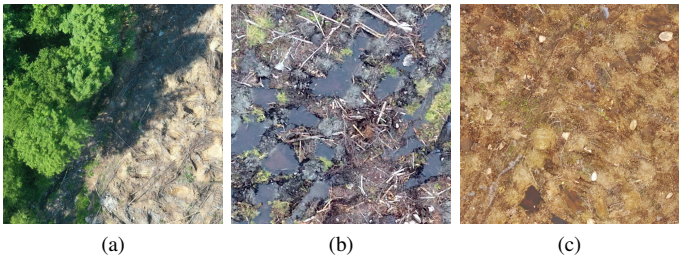


Fig. 1. Orthomosaic of one planting block captured and reconstructed (left) along with the example of one extracted patch (right).

To handle such difficult factors, a sequential two-step paradigm is used for system training. Firstly, we propose to use an instance segmentation model to detect mounds and quantify the presence of different objects in local patches. Secondly, we employ patch-level correction to obtain a final number of mounds. Fig. 3 illustrates the system training procedure.



(a) (b) (c)

Fig. 2. Examples of challenges for three patches from different orthomosaics. (a) Presence of tree shadow causing partial occlusion of mounds. (b) Water accumulation due to heavy rain. (c) Mounds with similar texture to the surrounding areas (background) in dry terrain.

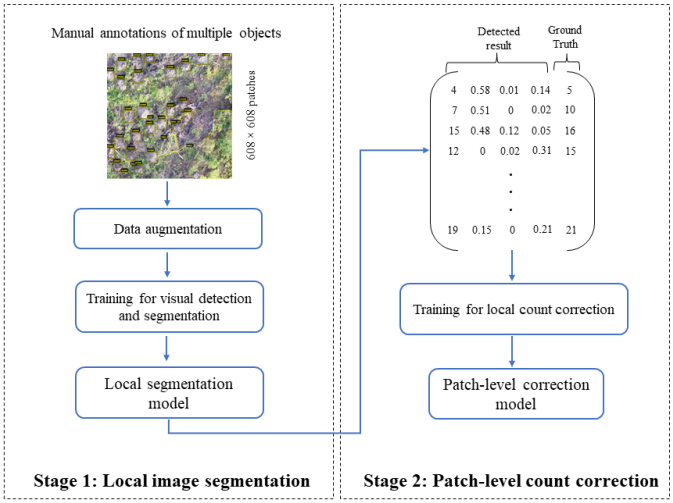


Fig. 3. Pipeline of the system training for two stages.

Once the entire system has been trained, the two models are used to perform mound counting on a new orthomosaic. That is, the local image segmentation is applied as a preliminary method to segment the objects at pixel-level on each patch of a planting block. The results of this step, which include the number of visually detected mounds and the ratio of other objects (e.g., tree, debris, and water), are then fed as input features to the second model for a final patch-level correction.

As stated above and explained further, the efficacy of merely performing the local object detection method degrades due to the presence of multiple objects and the limitations of occluded mounds. Therefore, our two-stage strategy is important for achieving accurate and precise counting under our application constraints. The procedure of analyzing a new patch of an orthomosaic to predict mound counting is depicted in Fig. 4. The details of each phase of our framework are presented in the following sub-sections.

B. Local image segmentation

The first step of our method is to perform local instance segmentation to identify and segment multiple objects in each

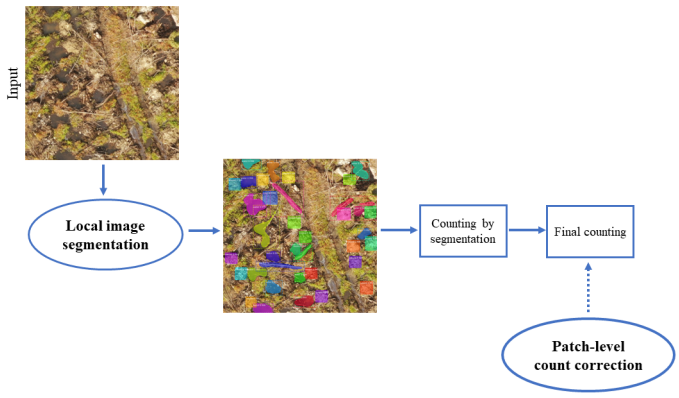


Fig. 4. The procedure for evaluating a new patch using our framework.

patch. The main motivation for this step is that different plantation blocks have different properties, and mound density highly depends on block characteristics. In this regard, quantifying the presence of mounds and other objects is an accurate objective indicator to obtain the properties of a bock. To accomplish this, local object instance segmentation is used to detect distinct objects belonging to the same category and to assign a unique instance label to the associated pixels.

Instance object segmentation combines object detection, which outputs bounding box coordinates, and semantic segmentation, which outputs segmentation masks. In this work, we use Mask R-CNN [28] for instance object segmentation to detect mounds and segment all objects in the image. Mask R-CNN is a cutting-edge instance segmentation technique that adds a segmentation mask generating branch to its predecessor, Faster-RCNN, to accomplish proper object detection and pixel-level instance segmentation. We employ Mask R-CNN because it is a two-stage object detector that takes advantage of anchor boxes. Anchor boxes enable this method to detect multiple objects in different scales, which improves the efficiency of the method and provide more accurate localization and classification. Mask R-CNN comprises two stages to produce a final mask segmentation of objects. The first stage scans over the image and generates the proposals, and the second stage predicts the class and box offset and produces a binary mask in parallel [28]. In these two stages, the Mask-RCNN architecture employs three modules: backbone, Region Proposal Network (RPN), and ROI.

The feature maps are constructed in the first stage by extracting image features of various scales using the backbone network such as ResNet (deep residual networks) [29], which is also known as the feature extraction network. After that, the obtained feature map is sent to the RPN, which generates proposals. In the second stage, the corresponding target features of the shared feature maps are extracted by mapping ROIs to feature layers. The RoIAlign is used to modify the feature map to a fixed-size feature map. Finally, the task of mask prediction is completed through FCN branch, and object classification and bounding box regression are completed by two branches of Fully Connected (FC) layers.

1) Training strategy: Deep learning approaches, in general, require a significant amount of data to be trained properly; otherwise, these techniques could fail to yield high accuracy. Therefore, we apply transfer learning due to the lack of training data and the purpose of applying Mask-RCNN as a deep learning-based approach. We trained all of the layers including the RPN, classifier, and mask head of our model network using pre-trained weights from the Common Objects in Context (COCO) [31] dataset. In addition to transfer learning, we used data augmentation process to address the issue of a limited number of real-world mounds to improve the recognition rate of our model. In this way, a range of some augmentation techniques were used to increase the diversity of the original training dataset.

Once Mask-RCNN is trained through the adaptation of the preceding methodologies, it is fitted to new datasets using a multi-loss function throughout the learning step. As shown in equation (1), the goal is to optimize model parameters by minimizing a multi-tasking loss function that incorporates a three-module combination loss: classification, localization, and segmentation.

$$L = L_{cls} + L_{box} + L_{mask}, \tag{1}$$

In this equation, L_{cls} represents the loss of classification, L_{box} represents the loss of prediction bounding box, and L_{mask} represents the loss of mask.

Based on the Mask-RCNN network, the mask branch contains a Km^2-dimensional output for each identified ROI that encodes K binary masks with a resolution of $m \times m$, representing K number of classes [28]. Thus, L_{mask} is defined as the average binary cross-entropy loss on the k-th mask, which is calculated using per-pixel sigmoid on $mask_k$, as defined below:

$$L_{mask} = Sigmoid(mask_k) \tag{2}$$

Local segmentation is done based on annotated patches of planting blocks for the whole terrain. The number of mounds and the ratio of the other three objects in each patch are then used as input to the local count correction.

C. Patch-level correction

The purpose of this stage is to accurately predict the number of mounds in a given orthomosaic representing a planting block. In our first stage, local image detection and segmentation is used to detect visible mounds. However, the number of visible mounds in an orthomosaic rarely corresponds to the actual number of planted seedlings, due to multiple factors, such as occlusion caused by woody debris or tree from neighboring zone (see Fig.5 (a) and (b)), and destroyed mounds by water flow (see Fig. 5 (c)).

Therefore, the number of detected mounds in a local patch is generally underestimated when relying only on detection techniques. To reduce this error, we use regression algorithms

(a) (b) (c)

Fig. 5. Examples of mound occlusion and destruction. (a) Occlusion due to the presence of woody debris. (b) Presence of tree from neighboring zone. (c) Destroyed mounds by water flow due to heavy rain.

based on Mask-RCNN counting results from the previous stage. The objective of regression analysis in this context is to map a function $X \rightarrow Y$, with X and Y for N sample images specified as follows:

$$\begin{aligned} \mathrm{X} &= \left\{x_i^j\right\}, \mathrm{j} \in (1,4), \mathrm{i} = 1,2,\ldots,\mathrm{N} \\ \mathrm{Y} &= \{\mathrm{y_i}\}, \mathrm{i} = 1,2,\ldots,\mathrm{N} \end{aligned} \tag{3}$$

where x^1 represents the number of detected mounds and x^2, x^3, x^4 represent the ratios of trees, water and debris respectively, and Y is defined as a corresponding ground-truth.

We investigated regression methods including, linear, Support Vector Regression (SVR), lasso, and Multilayer Perceptron (MLP) as pre-trained models on one orthomosaic in order to find the best predictor according to the best Relative Counting Precision (RCP) and perform it as a regression prediction on patches of any given block.

To determine the final total number of mounds for each block, we use the outcomes of the algorithm predictions for each patch as follow:

$$\mathrm{Count_{final}}\left(\mathrm{Block}_i\right) = \sum_{j=1}^{M}\left(P_j\right), \tag{4}$$

where P_j represents the j-th patch for a given Block(i).

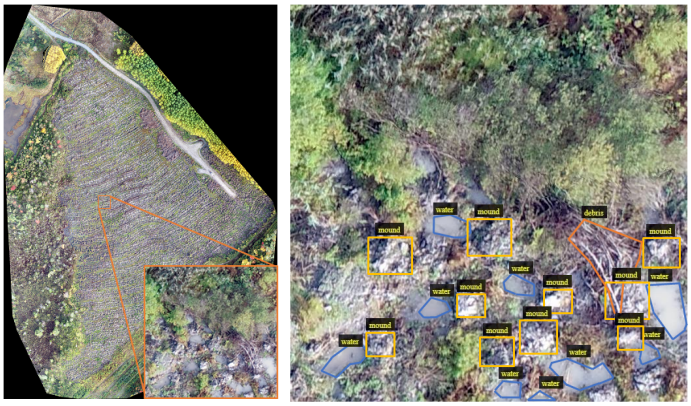
IV. Experiments and Results

A. Dataset construction

To provide input data, a total number of 20 orthomosaics reconstructed from UAV images from different zones with varying characteristics were used. We divided our images into two distinct groups as follows:

- **Group 1**: consists of 3 training orthomosaics that were manually annoated for local image segmentation training.
- **Group 2**: includes 18 testing orthomosaics used to evaluate the performance for the entire framework.

The aerial multispectral images were taken with a drone equipped with a high-resolution sensor set vertically and images captured with a high overlap percentage to maximize orthomosaic reconstruction quality at a height of 120 meters. Because the sensor produced images with a high resolution of 23610 × 18151, we performed a patch-based approach to trim orthomosaic and create non-overlapped patches with regular and stable pixel sizes of 608 × 608. As a result, 1352 patches were employed in total for the model. The data was then processed before being fed into the local instance segmentation method for training. The region of interest was carefully investigated, and the ground-truth of mounds and other objects including trees, water, and woody debris was manually annotated using the open-source VGG Image Annotator (VIA) tool [32]. An example of a patch cropped to a fixed cell size of 608 × 608 pixels, showing manually annotated objects, is presented in Fig. 6.



(a) (b)

Fig. 6. (a) Example of one orthomosaic and a sample patch cropped to a fixed dimension. (b) Manually annotated objects (mound, tree, water, debris).

B. Evaluation metric

We used the relative counting precision metric to measure the overall system performance by evaluating the regression predictors and obtaining a final counting estimate, as shown below:

$$RCP = 1 - \left|\frac{\#predicted\ mound - \#gt}{\#gt}\right| \tag{5}$$

where *#predicted-mound* represents the predicted number of mounds and $\#gt$ represents the number of mounds from the ground-truth.

C. Experimental results

The process of counting mounds on a new planting block comprised two steps:

1) Applying the local image segmentation model to detect visible mounds and quantify the presence of trees, debris and water.
2) Performing patch-level correction using a regression function.

Note that we only trained our local image segmentation using three annotated orthomosaics from Group 1. To ensure that the models work properly and verify the performance of our proposed method, we used 18 orthomosaics from Group 2 that had not been utilized in the training processes.

Fig. 7 depicts one sample patch and its corresponding qualitative result. The number of mounds and the ratio of

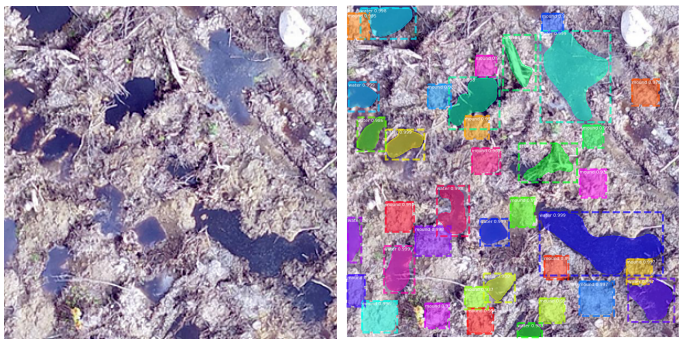
TABLE I

QUANTITATIVE RESULTS OF OUR PROPOSED APPROACH. GROUNDTRUTH REFERS TO THE FINAL NUMBER OF PLANT SEEDLINGS PLANTED IN A BLOCK, LOCAL SEGMENTATION-BASED COUNT IS THE NUMBER OF MOUNDS DETECTED AND SEGMENTED USING LOCAL IMAGE SEGMENTATION METHOD, COUNT IS THE NUMBER OF LOCALLY CORRECTED MOUNDS AND RCP CORRESPONDS TO THE RELATIVE COUNTING PRECISION. AVERAGE PRECISION MEASURE REPRESENTS THE AVERAGE OVER ALL PRECISION VALUES, BUT THE OVERALL RESULT INDICATES THE COUNTING PRECISION WHEN THE ENTIRE NUMBER OF MOUNDS IN THE DATASET IS TAKEN INTO ACCOUNT.

Orthomosaic	**GroundTruth**	**Local** *segmentation-based count*		**Patch-level corrected count**							
				Linear		***SVR***		***Lasso***		***MLP***	
		Count	***RCP***	***Count***	***RCP***	***Count***	***RCP***	***Count***	***RCP***	***Count***	***RCP***
Block 01	16450	14458	88%	15820	96%	15180	*92%*	15504	94%	15828	96%
Block 02	2650	2609	98%	2816	94%	2760	**96%**[a]	2851	92%	2780	95%
Block 03	750	712	95%	724	97%	737	**98%**	771	97%	728	97%
Block 04	800	784	98%	851	94%	844	*94%*	891	89%	837	95%
Block 05	2350	2233	95%	2495	94%	2436	**96%**	2562	91%	2462	95%
Block 06	1700	1513	89%	1623	95%	1600	*94%*	1683	99%	1621	95%
Block 07	2050	1853	90%	1868	91%	1879	*92%*	1933	94%	1864	91%
Block 08	3950	3443	87%	3676	93%	3571	*90%*	3649	92%	3629	92%
Block 09	6847	6632	97%	7041	97%	6915	**99%**	7091	96%	6923	99%
Block 10	30200	28301	94%	28973	96%	29145	**97%**	30107	99.7%	28733	95%
Block 11	2950	2742	93%	2778	94%	2797	*95%*	2894	98%	2765	94%
Block 12	25450	24251	95%	2765	96%	25447	**99.99%**	25994	98%	25848	98%
Block 13	7400	6658	90%	7825	94%	7551	**98%**	8079	91%	7824	94%
Block 14	5250	5009	95%	5620	93%	5468	**96%**	5751	90%	5563	94%
Block 15	3557	3424	96%	3636	98%	3653	**97%**	3842	92%	3643	98%
Block 16	5150	4320	84%	5362	96%	5032	**98%**	5418	95%	5331	96%
Block 17	4900	4759	97%	5164	95%	5025	**97%**	5236	93%	5128	95%
Block 18	2650	2267	86%	2478	94%	2492	*94%*	2670	99.2%	2471	93%
Overall result	125054	115968	93%	101515	81%	122532	**98%**	126926	99%	123978	99%
Average precision			93%		95%		**96%**		94%		95%

[a]Highlighted numbers contribute more significantly to overall precision.

other detected and segmented objects were then used as input features in the second step of the pre-trained regression algorithm to produce the final mound counting.



(a) (b)

Fig. 7. (a) Example of one patch. (b) Corresponding qualitative result.

Table I shows the quantitative results of our proposed approach. According to the findings, the RCP of local image segmentation method is 93%, which indicates that our instance segmentation method has the ability to detect and segment the planting microsites efficiently. However, we could significantly improve the average detection precision of counting mounds by applying patch-level correction methods. From table I, this improvement reached 96% by performing SVR. Compared with other regression methods, SVR yields the greatest results since it employs a kernel function to concurrently minimize prediction errors and model complexity.

Although the experimental results show the efficiency of our strategy, this study is subject to several challenges. According to the results of local-segmentation mound counting, the RCP for blocks 01, 06, 08, 16, and 18 are less than 90% compared to the other ones, which are equal to or greater than 90%. This can be caused by the fact that the new plantation block may exhibit many unseen properties when we test our object-pixel-wise instance detector. This could include mound shapes and block characteristics that the detector was not exposed to during training. In addition, table I indicates that while our framework was able to improve the final results by applying SVR, the outcomes for some blocks are lower than others. As can be seen from block 08, our correction method could only correct 128 more mounds than the local approach, bringing the total number of mounds from 3443 up to 3571, which happened due to the challenging situations on the captured images of the related blocks. For instance, because the images of block 08 were taken under dry conditions, the texture of mounds is similar to surrounding regions, which makes object detection very challenging.

Despite the aforementioned difficulties, the overall results demonstrate that using the two stages consecutively results in an average improvement of 3%. Additionally, our framework achieved an average RCP of 96%, and thus outperforms the manual method whose RCP is around 85%.

V. CONCLUSION

In this paper, we proposed a new computer vision framework to detect and segment multiple objects, followed by

counting mechanically created mounds for planting from UAV images. The proposed system consists of an hybrid approach, which combines a local image segmentation method with patch-level count correction. In this regard, the objective of local image segmentation method was to segment pixel-level visual mounds. The accurate number of final mounds was then determined using the patch-level correction approach. The experimental results demonstrate that our approach is effective in dealing with a variety of difficult conditions involving environmental factors and the existence of multiple objects in each terrain patch. That is, the local segmentation approach leverages visual mounds from aerial images to detect a preliminary count, which is subsequently improved by the patch-level correction method. According to qualitative and quantitative performance assessments, our approach outperforms traditional counting methods (i.e., field work) in terms of precision, and reduces the financial cost of planning planting operations significantly.

REFERENCES

[1] M. Löf, D. C. Dey, R. M. Navarro, and D. F. Jacobs, "Mechanical site preparation for forest restoration," New Forests, vol. 43, no. 5, pp. 825-848, 2012.

[2] W. Jingying, "A survey on crowd counting methods and datasets," Advances in Computer, Communication and Computational Sciences, pp. 851-863: Springer, 2021.

[3] N. Dalal, and B. Triggs, "Histograms of oriented gradients for human detection," IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 886-893, 2005.

[4] P. Viola, and M. J. Jones, "Robust real-time face detection," International journal of computer vision, vol. 57, no. 2, pp. 137-154, 2004.

[5] Z. Fan, H. Zhang, Z. Zhang, G. Lu, Y. Zhang, and Y. Wang, "A survey of crowd counting and density estimation based on convolutional neural network," Neurocomputing, vol. 472, pp. 224-251, 2022.

[6] B. Wu, and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors," International Journal of Computer Vision, vol. 75, no. 2, pp. 247-266, 2007.

[7] Y. Wang, J. Hou, X. Hou, and L.-P. Chau, "A self-training approach for point-supervised object detection and counting in crowds," IEEE Transactions on Image Processing, vol. 30, pp. 2876-2887, 2021.

[8] P. Sabzmeydani, and G. Mori, "Detecting pedestrians by learning shapelet features," IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, 2007.

[9] J. Ilao, and M. Cordel, "Crowd estimation using region-specific HOG With SVM," 15th International Joint Conference on Computer Science and Software Engineering (JCSSE), pp. 1–5, 2018.

[10] B. Zhou, M. Lu, and Y. Wang, "Counting people using gradient boosted trees," IEEE Information Technology, Networking, Electronic and Automation Control Conference, pp. 391–395, 2016.

[11] V.-Q. Pham, T. Kozakaya, O. Yamaguchi, and R. Okada, "Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation," IEEE International Conference on Computer Vision (ICCV), pp. 3253–3261, 2015.

[12] D. Kang, Z. Ma, and A. B. Chan, "Beyond counting: Comparisons of density maps for crowd analysis tasks—counting, detection, and tracking," IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 5, pp. 1408-1422, 2018.

[13] N. Paragios, and V. Ramesh, "A MRF-based approach for real-time subway monitoring," Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 1, pp. I–I, 2001.

[14] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–7, 2008.

[15] A. Marana, L. d. F. Costa, R. Lotufo, and S. Velastin, "On the efficacy of texture analysis for crowd monitoring," Proceedings SIBGRAPI'98. International Symposium on Computer Graphics, Image Processing, and Vision (Cat. No.98EX237), pp. 354–361, 1998.

[16] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, "Deep people counting in extremely dense crowds," Proceedings of the 23rd ACM international conference on Multimedia, pp. 1299-1302, 2015.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems, pp. 1097-1105, vol. 25, 2012.

[18] M. Fu, P. Xu, X. Li, Q. Liu, M. Ye, and C. Zhu, "Fast crowd density estimation with convolutional neural networks," Engineering Applications of Artificial Intelligence, vol. 43, pp. 81-88, 2015.

[19] L. Boominathan, S. S. Kruthiventi, and R. V. Babu, "Crowdnet: A deep convolutional network for dense crowd counting," In proceedings of the 24th ACM international conference on Multimedia, pp. 640-644, 2016.

[20] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 589-597, 2016.

[21] D. Babu Sam, S. Surya, and R. Venkatesh Babu, "Switching convolutional neural network for crowd counting", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5744-5752, 2017.

[22] Y. Li, X. Zhang, and D. Chen, "Csrnet:Dilated convolutional neural networks for understanding the highly congested scenes", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1091-1100, 2018.

[23] B. Sheng, C. Shen, G. Lin, J. Li, W. Yang, and C. Sun, "Crowd counting via weighted VLAD on a dense attribute feature map," IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 8, pp. 1788-1797, 2016.

[24] C. Shang, H. Ai, and B. Bai, "End-to-end crowd counting via joint learning local and global count," IEEE International Conference on Image Processing (ICIP), pp. 1215-1219, 2016.

[25] Y. K. Lin, C. F. Wang, C.-Y. Chang, and H. L. Sun, "An efficient framework for counting pedestrians crossing a line using low-cost devices: the benefits of distilling the knowledge in a neural network, " Multimedia Tools and Applications, vol. 80, no. 3, pp. 4037-4051, 2021.

[26] J. Redmon, and A. Farhadi, "Yolov3: An incremental improvement," ArXiv, vol. abs/1804.02767, 2018.

[27] W. Bouachir, K. E. Ihou, H.-E. Gueziri, N. Bouguila, and N. Bélanger, "Computer vision system for automatic counting of planting microsites using UAV imagery," IEEE Access, vol. 7, pp. 82491-82500, 2019.

[28] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," IEEE International Conference on Computer Vision (ICCV), pp. 2980-2988, 2017.

[29] He, K., Zhang, X., Ren, S., Sun, J., "Deep Residual Learning for Image Recognition," In IEEE conference on computer vision and pattern recognition, pp. 770-778, 2016.

[30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," Advances in neural information processing systems, vol. 28, 2015.

[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," In European conference on computer vision, pp. 740-755. Springer, Cham, 2014.

[32] A. Dutta, and A. Zisserman, "The VIA annotation software for images, audio and video," Proceedings of the 27th ACM International Conference on Multimedia, pp. 2276-2279, 2019.