MoundCount: A detection-based approach for automatic counting of planting microsites on UAV images

Ahmed Zgaren[1,2] Wassim Bouachir[1] Nizar Bouguila[2] Riad I. Hammoud[3]
[1]Data Science Laboratory, University of Québec (TÉLUQ)
[2]Institute for Information Systems Engineering, Concordia University
[3]PlusAI Inc.
ahmed.zgaren@concordia.ca, wassim.bouachir@teluq.ca, nizar.bouguila@concordia.ca, riad.hammoud@plus.ai

Abstract

Planting by mounding is a commonly used forestry technique that improves soil quality and ensures optimal tree growth conditions. During planting operations, one of the main planning steps is to estimate the number of mechanically created mounds in each planting block. Traditional counting methods involve manual field surveys or human photo-interpretation of UAV images, which are generally subject to errors and time-consuming. In this work, we propose a new approach to count mounds on UAV orthomosaics. Our framework is designed to estimate the required number of seedlings for a given planting block, based on a visual detection approach and a global estimation module. Firstly, a deep local detection model is applied on local patches to recognize and count visible mounds. Then, an estimation model, based on global features is used to predict the final number of plant seedling required for a given plantation block. To evaluate the proposed framework in real-world conditions, we constructed a large UAV dataset, including 18 UAV orthomosaics, comprising 111,000 mounds. We have conducted extensive experiments in our dataset, including a comparison with the state-of-the-art counting methods, as well as an analysis of Human-Level Performance (HLP) in identifying and annotating mounds. The experimental results show that our model reaches the best performance in terms of MAE and MSE, by comparison to state-of-the-art automatic counting mehtods.

1. Introduction

In forest industry, mechanical site preparation is commonly used before planting seedlings on a new plantation block, which consists in preparing forest floor mechanically by forming planting microsites, also called mounds. An important step when planning planting operations is to estimate the number of mounds created on each site. Mound counting errors may impact inventory reports and logistic plans, which could result in considerable waste of time and money.

Forestry managers often use traditional counting approaches, where several workers move through the plantation block and visually count the number of mounds in a specific portion, then extrapolate to neighboring regions. These operations are time consuming and subject to errors due to the extrapolation method, which assumes that mound density remains constant for a given plantation block. Recently, due to the flexibility and the low cost of Unmanned aerial vehicles (UAVs), aerial imagery was also used to manually count planting micro-sites through human photo-interpretation. The photo-interpretation method involves visual mound identification by human operators on UAV images. Counting is done in some regions of the image, to be extrapolated over the entire planting block. This method is also subject to errors due to two main reasons: human errors during mound identification, and the constant mound density assumption, which is often invalid.

In this work, we propose a new visual automated method to count mounds on UAV images. We formulate the mound counting task as a supervised detection problem, by sequentially performing object detection and global count correction. Our two-stage approach for object counting effectively leverages both detection-based and regression-based models. First, an object detection algorithm is used to identify and locate visual mounds in an image. Object detection generally allows for accurate counting of objects in crowded and complex scenes, even when objects are overlapping or partially occluded. However, we argue that our counting problem cannot be addressed by merely relying on object detection. This is due to the non trivial nature of our task, with a high scene appearance variability, and where objects of interest (mounds) could be completely invisible due to

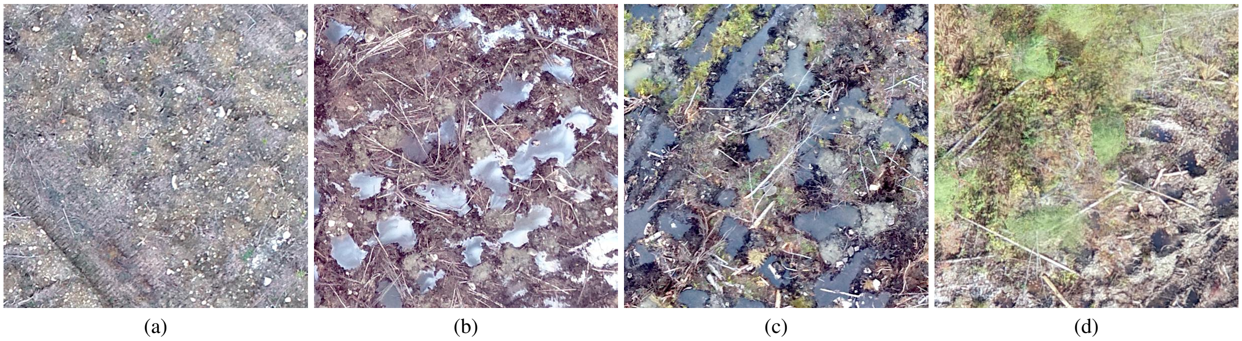


Figure 1. Examples of patches from different planting blocks. (a) mounds have a similar texture to that of the background (surrounding regions). (b) & (c) the planting block was flooded by water flow and mounds have been partially or completely destructed or occluded. (d) the presence of trees may occlude mounds in the border.

several perturbation factors. This includes total occlusion by woody debris, water accumulation, and mound erosion and destruction (see figure 1). We thus design a global estimation module to be applied following the detection stage. This module uses global features, in addition to detection results, to estimate the overall count of objects in an image, regardless of their visibility. In our framework, these two stages are essential and complementary, as object detection is limited by the visibility of objects, while global estimation is not.

By using these two stages, our method is able to achieve high accuracy in mound counting, even in challenging and complex scenes. Additionally, our method is computationally efficient, as it uses object detection to reduce the search space for global estimation. This reduces the amount of computation required, which is often computationally expensive with standard exhaustive methods.

The main contributions of our paper can be summarized as follows:

- We propose a new UAV-based method to automate mound counting in forest industry.
- The proposed hybrid counting framework can be generalized for similar counting problems, using a wide range of UAV image types, including RGB, multispectral, near infrared (NIR), and thermal IR.
- We construct a new UAV dataset including 18 orthomosaics and a total number of 111,000 mounds.
- We present an analysis of Human Level Performance (HLP) in the studied visual object identification task.
- We present a comprehensive experimental evaluation of our framework by comparison to state-of-the-art object counting methods.

2. Related works

Computer vision has been often used in environmental applications, such as for animal counting [3] and tree inventorying [13]. Several of these applications are based on the task of estimating the number of objects present in images, referred to as crowd counting. This section reviews the most significant studies and methods proposed to count objects in crowded scenes. Methods from the literature can be categorized into two main approaches: 1) the traditional approach, mostly based on hand-crafted features and classical machine learning models, and 2) the Deep Learning (DL) approach, where DL models are applied to learn and classify crowd regions.

2.1. Traditional counting methods

Traditional approaches are mostly based on image processing techniques to extract hand-crafted features [25], in addition to machine learning methods to estimate object count. Early approaches used detection methods [7] coupled with nonlinear classifiers to learn object patterns, such as support vector machine (SVM) [11], boosted trees [34] and random forests [23]. Despite detection-based methods were generally successful in low density counting, these methods often failed in moderately and highly crowded scenes. To overcome the problem of moderately crowded scenes, researchers used regression methods, which aim to find the final count from input local features. Regression counting methods generally comprise two steps. The first step corresponds to low level features extraction, while the second is regression modeling. Low level visual information extraction is performed to obtain different types of features, such as foreground [5], edge [19], texture and gradient [9, 29] features. Then, various regression modeling techniques can be used to learn a mapping between the low-level features and the final count, such as linear regression [22], piece-wise linear regression [4], and neural networks [18]. Idrees et al. [10] proposed to combine different feature extraction methods to capture multiple information, subsequently used by a global multiscale Markov Random Field (MRF) to estimate the final count.

Density map methods were introduced by Lempitsky et al. [15] to linearly map between local patch features and their corresponding object density maps. They formulated the density map learning as a minimization of a regularized risk quadratic cost function using a new loss function for learning. Several improvements have been subsequently added by researchers. Pham et al [23] proposed a crowdedness prior to overcome the problem of large variation in appearance and shape.

2.2. Deep learning counting methods

The first proposed DL methods in crowd counting were based on classification-based CNNs. Wang et al. [31] proposed an end-to-end CNN regressor for people counting. The model is based on AlexNet [14] architecture, where the final fully connected layer is replaced by a single neuron to predict the final count. Fu et al. [8] optimized a multistage CNN by removing some network connections. Then, two CNN cascade classifiers were designed to classify crowd images into five classes: very low, low, medium, high, and very high. To optimize the model on cross-scene crowd counting, Zhang et al. [32] proposed to alternatively train a CNN for two tasks: density estimation and crowd counting. To increase counting accuracy and time processing, Walach et al. [30] proposed an improved CNN architecture with layered boosting and selective sampling technique during training.

To ensure robustness against object scale variation, advanced DL models were introduced. Two main architectures were used: multi-columns based architecture [6, 33] and multi-resolution based architecture [1, 21]. Different approaches have attempted to integrate local or global context information to solve the crowd-counting problem using DL models. The authors in [28] proposed to integrate semantic information by learning locality aware feature (LAF) sets. Further, Ilyas et al. [12] have designed an end-to-end CNN model for crowd counting, which combined a features extraction network and a scale-aggregation module, with dilated convolution to collect large scale contextual information. A hybrid approach was proposed in [2] to count planting microsites in multispectral orthomosaics by generating region proposals based on local binary patterns (LBP) features extracted from near-infrared (NIR) patches. Then, a convolutional neural network (CNN) is used for classifying candidate regions. A recent work was proposed by Nategh et al. [20] to count mounds using instance segmentation. Mask R-CNN was applied to local patches to quantify the presence of objects of interest, in addition to distractors. Counting is then refined using local correction at the patch-level.

3. Proposed method

3.1. Motivations and overview

Our purpose is to precisely estimate the number of plant seedlings to be planted in a plantation block, which should correspond to the number of created planting microsites. Mound detection is a complex task due to several challenges, as shown in Figure 1. This includes similarity in appearance with background, variation in shape between mounds, and variability in appearance between planted blocks (as seen in examples a, b, c, and d). State-of-the-art deep learning-based counting methods may face significant challenges and limitations under our application constraints. One of the main challenges is the need for large amounts of labeled training data. Another challenge is that the accuracy of the model is highly dependent on the quality and diversity of the training data. If the training data does not adequately represent the range of variability in the appearance and shape of mounds, then the model may not be able to accurately perform counting on new, unseen images. Furthermore, deep learning models are known to struggle with generalizing to new environments or image types that are significantly different from those of the training set. This means that a deep learning model trained to count mounds in a given type of block or environmental condition may not be accurate when applied to a different context.

To address these challenges, we designed a system based on two distinct, yet complementary prediction models. First, we train a DL object detector using manually annotated images to detect visible mounds in a local patch. After estimating a preliminary number of mounds by detection, a global correction model is applied using global features extracted from each block. The correction model is trained beforehand to estimate the final number of mounds in each block, based on global block-level features, such as block area, global mound density, and detection-based count. This global estimation model is crucial to handle the limitations of visual object detection dicussed above, including the total occlusion of mounds or their complete destruction due to erosion. The proposed framework is illustrated in figure 2 and detailed in the following sub-sections.

3.2. Mound detection in local patches

In our approach, we use the supervised one-stage detector YOLO [24], to perform mound detection in a single block. YOLO has shown its robustness to scale and perspective variation in a variety of difficult situations and has achieved high-precision results in challenging large-scale datasets. Moreover, YOLO is a real-time object detector that takes only $50ms$ to process a 608×608 image using a GPU device. The YOLO model has a fully convolutional architecture, which could be divided into two parts: Back-

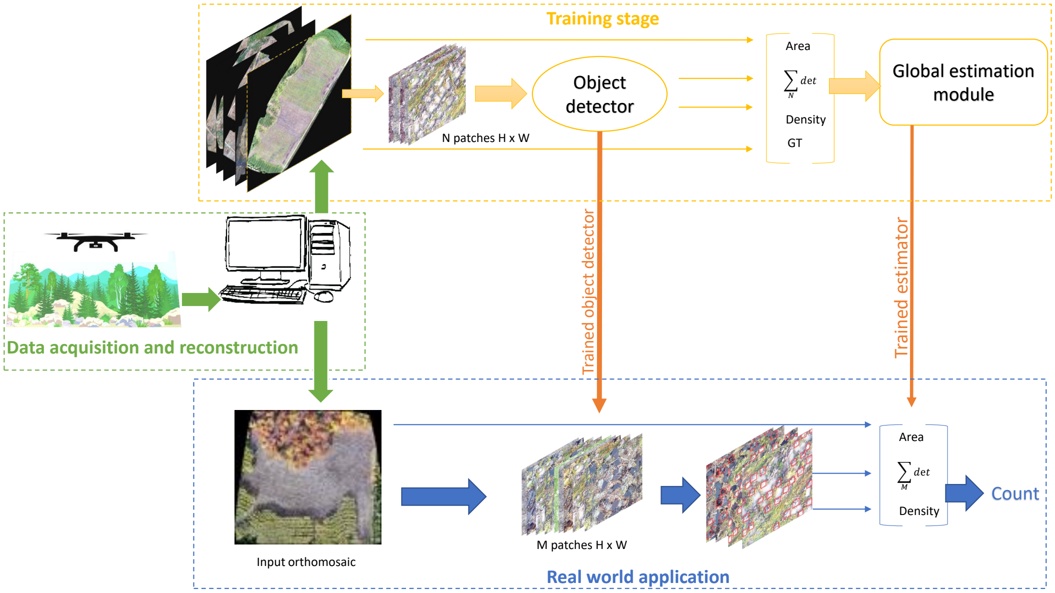


Figure 2. Overview of the proposed framework. UAV images are firstly captured and used for constructing orthomosaics representing the plantation blocks. Then, the object detector and the global estimator are trained using a sample dataset. Once the system is trained, it can be used in a real-world scenario by processing a new orthomosaic.

bone, and head detector. The backbone is a CNN that produces smaller feature maps using Darknet-63 architecture. The output of the backbone is used as input for the head detector, which corresponds to M_f stacked feature maps within a tensor size of $M_f \times H_f \times W_f$. Due to the Features Pyramid Network (FPN) architecture [17] integrated into YOLO design, detection is performed in three different scales: small, medium, and large.

Given the requirement of a large annotated dataset for training deep neural networks, transfer learning is implemented utilizing pre-trained weights on ImageNet [14]. This approach enables the model to leverage the learned weights to extract deep features and accurately localize over 1000 objects in images. To evaluate the model's performance, we trained it on a subset of orthomosaics and tested it on the remaining samples.

We use data augmentation to overcome the lack of annotated data during training. Since precise mound localization is not the goal of our application, we generate new images by applying modifications to bounding boxes, including scale change and translation.

- **Scale:** we modify the bounding box size to include more background information with respect to object size using

$$\begin{cases} H_{new} = H_{old} \times Z \\ W_{new} = W_{old} \times Z \end{cases}, \tag{1}$$

where H and W are respectively the height and the width of the bounding box, and Z is the scale.

- **Translation:** we generate a new centered bounding box using

$$\begin{cases} Cx_{new} = Cx_{old} + (L \times cos(\alpha)) \\ Cy_{new} = Cy_{old} + (L \times sin(\alpha)) \end{cases}, \tag{2}$$

where Cx and Cy are the coordinates of the center of the bounding box, L is the translation in pixels, and α is the translation direction angle.

3.3. Final count estimation using global features

The final estimated number of mounds should correspond to the required number of plant seedling to be planted in a given block. It is important to note that the number of visible mounds is generally different from the final number of seedlings to be planted because of the previously mentioned factors, including mound destruction, occlusion, and variability (see figure 1). It is important therefore to note that the mound detection result obtained during the first stage is considered as a preliminary count, and that our final goal is to estimate the number of plant seedling to be planted. For this purpose, we define a global count estimator that maps between the global information on a planting block and the final number planted seedlings on the considered block. We train the model using global features extracted from planted blocks, such as the number of detected mounds, block area, and density. Given a set of N training observations $X = \{x^i\}, i = 1, 2..., N$ (representing

N planting blocks), and corresponding ground-truth count $Y = \{y^i\}, i = 1, 2..., N$, for each observation x^i we define a set of M global features $x^i = \{x^i_j\}, j = 1, ..., M$.

To model the relationship between predictor variables (x^i) and the target variable (y^i), we train a mapping function $F : X \to Y$ to estimate the final count from the set of global features for each block, using regression analysis. We adopt ridge regression to find an M-dimensional weight vector $W = \{w_j\}, j = 1, ..., M$ minimizing the loss function

$$\underset{w}{min} \sum_{i=1}^{N} \left(y_i - \sum_{j=1}^{M} w_j \times x^i_j \right)^2 + \lambda \sum_{j=1}^{M} w_j^2 \qquad (3)$$

where y_i is the target variable, x^i_j is the predictor variable, w_j is the weight to be learned, and λ the shrinkage parameter controlling the balance between prediction error and regularization of W.

To train the regression model, we construct a training dataset where each training example is a global feature vector describing an orthomosaic. Note that prior to constructing the feature vector of an orthomosaic, the local detector is applied in order to obtain a detection-based count as described in section 3.2. In addition to the detection-based count, the feature vector x^i includes other block-level information, namely, the average detection density per patch, the average mound density according to user input (i.e. fine-tuning annotations), and the planting block area.

4. Experiments and results

In this section, we evaluate the effectiveness of our proposed visual counting method. We compare it to the state-of-the-art methods for counting by density estimation, MC-CNN [6] and CSRNET [16]. Additionally, we evaluate our method by comparison to the popular object detection methods YOLO [24] and Faster R-CNN [26].

Further, we conduct an analysis of Human Level Performance (HLP) to evaluate how well humans can recognize mounds in images. This analysis provides a benchmark for evaluating the performance of our proposed method in relation with human ability. Moreover, it helps us understand the impact of human recognition performance on model training and performance evaluation. By conducting such an analysis, we aim to gain insights into the strengths and limitations of both human and machine perception, which can inform the design and development of more effective visual counting and object detection models.

4.1. Dataset

We constructed a new dataset consisting of 18 orthomosaics. The dataset was created by conducting a drone flyover the study area in the south of Quebec, Canada, followed by capturing overlapped images using a vertical camera. These images were then processed to construct the orthomosaics for each studied mechanically prepared plantation block. This process was repeated for all 18 blocks in the dataset. By following this methodology, we were able to create a high-quality dataset that accurately represents the study area and allows for a comprehensive evaluation of our proposed method's performance.

The final orthomosaics constructed from drone images has a resolution of approximately $20,000 \times 30,000$ pixels, which is larger than the input size accepted by the models used in our study. We thus divided each orthomosaic into smaller patches of size 608×608 pixels. The training set consisted of six blocks, each being further subdivided into small patches. All patches were meticulously annotated and augmented to produce a labeled dataset comprising a total number of $90,000$ mounds. This process enabled us to optimize the use of the large orthomosaic while also facilitating the training of our models on smaller, more manageable input sizes.

4.2. Implementation

Python was used to implement the proposed method on a PC with an i7-8700 CPU, (6 cores) running at 3.2GHz, and equipped with an Nvidia Geforce GTX 1070 GPU. To train YOLO for object detection, the batch size was set to 16, and the learning rate was set to 0.001 with a decay of 0.0005, while the momentum was set to 0.9. The detector was trained for 30 epochs, with the number of epochs fixed. Note that in our implementation we used Yolov5 the most recent version of yolo during our experiments.

For data augmentation, we set the parameters Z, L, and α to random numbers in the range of $[0.8, 1.2]$, $[1, 10]$, and $[0, 2\pi]$, respectively, while the regressor parameter λ was set to 10. To construct the object detector, we used a patch size of 256×256 pixels, which resulted in approximately 10,000 training patches after data augmentation, comprising around 95,000 annotated mounds over the 30 epochs. We performed transfer learning for YOLO by using the pre-trained model on the ImageNet dataset [27] for weight initialization. During the detection process, we set a confidence threshold of 0.25 to ensure that the most probable mounds were detected while minimizing the number of false positives.

4.3. Evaluation metrics

To evaluate the performance of our proposed counting framework, we utilized several metrics, including the Relative Counting Precision (RP), Mean Absolute Error (MAE), and Mean Squared Error (MSE). These metrics were chosen due to their ability to provide a comprehensive assessment of the accuracy and precision of the counting framework. The RP metric is defined as the ratio of the absolute difference between predicted and ground-truth counts to the ground-truth count. The MAE measures the average magnitude of errors between the predicted and ground-truth counts, while the MSE metric computes the average of the squared errors between the predicted and ground-truth counts. The mathematical expressions of the used metrics are as follows:

$$RP = 1 - (|\frac{\#\text{Predicted\_mounds} - \#\text{GT}}{\#\text{GT}}|) \quad (4)$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\#\text{Predicted\_mounds} - \#GT| \quad (5)$$

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\#\text{Predicted\_mounds} - \#GT)^2} \quad (6)$$

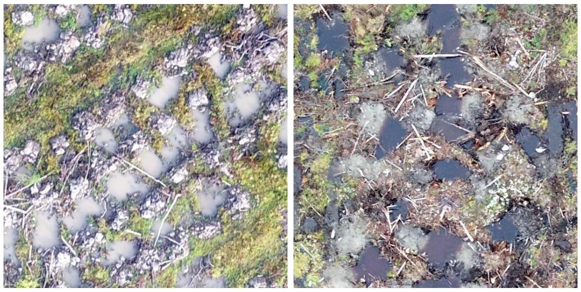
where, $\#GT$ is the number of planted seedlings and N is the number of blocks in the testing set. Together, these metrics enable us to quantify the performance of the counting framework across different image sizes, scales, and densities of mounds, providing valuable insights into its effectiveness and potential limitations.

4.4. Comparison with state-of-the-art count methods

To evaluate the performance of our proposed model, we use the constructed dataset to train and test the model. Five over 18 blocks were used for training and validation of all methods. The remaining 13 blocks were used for testing. We conducted a comparative analysis of various counting methods, including state-of-the-art models. Specifically, we compared our model, named MoundCount, to the two visual object detection-based methods FasterR-CNN [26] and YOLO [24], as well as the two density-based methods, MCCNN [33] and CSRNet [16]. The results of this comparison are presented in Table 1, which includes the count results as well as the relative precision results over 13 blocks.

Firstly, our proposed method outperforms baseline visual detection methods on 12 over 13 test blocks, which confirms that our global estimation module improves the detection count performance. We observe from table 1 that the proposed global estimation module improves the final count by 70% in block 12, to achieve a 92% precision, by comparison to merely using YOLO. Secondly, the comparison with the density estimation based-methods (MCCNN and CSRNET) shows that our framework achieved the best overall performance.

Table 2 presents the overall performance of various counting methods on the test set, measured by MRP (Mean Relative Precision), MAE (Mean Absolute Error), and MSE (Mean Squared Error). Notably, our approach ranked first in MRP with 88%. On the other hand, our method outperformed the others in terms of MAE and MSE, achieving the best results of 608 and 703, respectively. This represents a segnificant improvement of 63.59% and 76.62% on MAE and MSE, respectively, compared to the MCCNN method.


(a) Block A

(b) Block B

Figure 3. Example of patches from block A (a) and block B (b)

These results highlight the effectiveness of our approach in accurately estimating counts, outperforming state-of-the-art methods. The results also suggest that our method is particularly useful in real-world applications, where precise counting is essential.

4.5. Human Level Performance

Recognizing mounds in orthomosaics is a challenging task, even for humans. Therefore, we defined an experimental protocol to evaluate the Human Level Performance (HLP) in the mound identification task. The primary objective of this study was to analyze the challenges faced by humans in identifying mounds and their impact on model training.

To achieve this goal, we defined a procedure for the recognition of mounds in orthomosaic images. The procedure involved visual recognition by two participants, covering multiple blocks, and training each participant separately. To ensure unbiased results, participants were not provided any feedback or correction during the annotation process, and were prohibited from communicating with each other. By following this procedure, we aimed to accurately evaluate the HLP and gain insights into the strengths and limitations of human perception in mound identification.

For this experiment, we prepared two blocks, consisting of a total number of 637 patches (refer to Figure 3). We trained two participants for two days, providing both theoretical and practical training on different blocks. To assist with the visual recognition process, we provided a guidance file as a reference document. The block patches were sent separately to each participant. Once they had finished visually recognizing the mounds in the first block, we provided them with the second one.

To record the visually recognized mounds, we used the

name	GT	YOLO [24]		Faster R-CNN [26]		CSRNET [16]		MCCNN [33]		MoundCount	
		count	RP	count	RP	count	RP	count	RP	count	RP
Block1	6500	6866	*94%*	5533	85%	5651	87%	7455	85%	6172	**95%**
Block2	4350	2506	58%	132	3%	1545	36%	2273	*52%*	3781	**87%**
Block3	8900	7713	87%	4441	50%	8010	90%	9011	**99%**	8318	*93%*
Block4	1400	1023	73%	1058	76%	1201	**86%**	1920	63%	1734	*76%*
Block5	12350	8442	68%	5702	46%	6099	49%	8594	*70%*	11773	**95%**
Block6	6150	5829	*95%*	2354	38%	4652	76%	5623	91%	6019	**98%**
Block7	16450	15661	**95%**	13467	82%	21050	72%	26088	41%	15226	*93%*
Block8	3750	3214	86%	1586	42%	4247	87%	3556	**95%**	4158	*89%*
Block9	2050	771	38%	930	45%	2644	71%	2033	**99%**	1764	*86%*
Block10	7150	2775	39%	3106	43%	5602	78%	5889	*82%*	6000	**85%**
Block11	2300	1735	75%	803	35%	2423	**95%**	2153	*94%*	1775	77%
Block12	6600	1803	27%	2607	40%	6051	*92%*	5088	77%	6050	**92%**
Block13	13100	8914	68%	4572	35%	15016	85%	14090	**92%**	14364	*90%*

Table 1. Comparison to state-of-the-art counting methods on the constructed dataset. The proposed method outperforms automated visual counting methods based on the relative precision metric. Best results are in bold and second best are in italic

Method	MRP	MAE	MSE
YOLO [24]	69.34%	1887	2526
Faster R-CNN [26]	50.88%	3443	4108
CSRNET [16]	78.20%	1717	*2463*
MCCNN [33]	*82.29%*	*1670*	3008
MoundCount	**87.43%**	**608**	**703**

Table 2. Quantitative comparison results of our proposed method with the state of the art counting methods in the constructed dataset.

	Block A		**Block B**	
Ground Truth	8600		7150	
	Count	RP	Count	RP
Participant1	8257	96.1%	6292	88.08%
Participant2	8373	97.26%	6312	88.28%

Table 3. Comparison between HLP performance of participants on two different blocks.

VGG Image Annotator tool and collected JSON files from both participants. The count results and precision for each participant in the two blocks are presented in Table 3. Based on these results, we can see that the human error varied between blocks A and B, with rates of 3.99% and 11.94%, respectively. One possible explanation for this performance degradation between blocks is the challenging environmental factors that were present during visual recognition, such as occlusion due to debris, trees, and water flow. These factors may have affected the participants' ability to accurately identify mounds in images. Additionally, there may have been variations in the block properties, lighting, and image quality between the two blocks, contributing to the differences in performance.

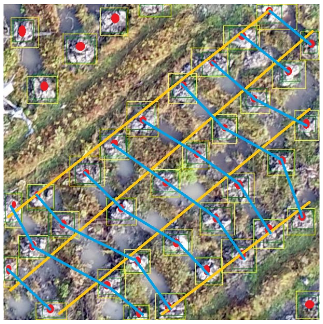


Figure 4. Example of patch showing a regular pattern. Red points are centers of mounds. Vertical orange lines and blue lines shows a regular prepared zone.

In general, mechanical preparation by mounding exhibits a regular pattern across all the blocks. Each block is comprised of parallel zones separated by excavator tracks, as depicted in Figure 4. Within each zone, four vertical lines of yellow-orange color denote the preparation lines for planting. These preparation lines contain four mounds, marked by horizontal blue lines. Figure 5 shows two patches from blocks A and B, respectively, demonstrating this regular pattern. In these patches, the mounds and the lines of four mounds are clearly visible, facilitating the visual recognition process. As a result, both participants were able to provide similar and complete annotations for these patches. It should be emphasized that this regular pattern of mounding is not present in all patches due to the challenging factors discussed previously.

In some situations, visual identification can be particularly challenging. The invisibility of mounds is the primary cause of such irregularities. Identifying occluded or destroyed mounds in images from both participants is often

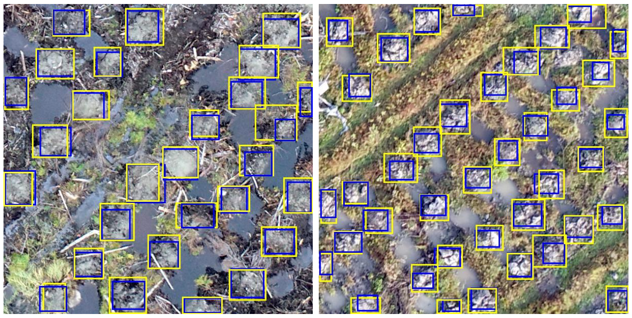


Figure 5. Qualitative results for visual recognition in patches with regular patterns. Yellow and blue rectangles are respectively annotations of participants 1 and 2. Left, a regular patch example from block A, and right, is a regular patch example from block B

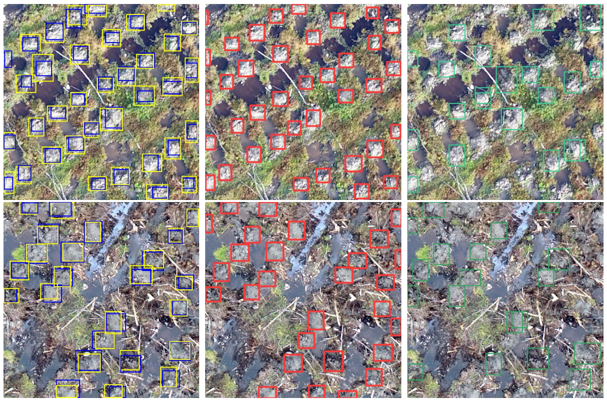


Figure 6. Qualitative comparison between human annotations and visual detection using two different methods. The first column corresponds to human annotations, the second column to YOLO detections, and the third to Faster R-CNN detections.

a difficult task due to the presence of debris. Additionally, water over slopes may cause the erosion of mounds and the alteration of the visual texture of blocks. Furthermore, mechanical preparation of certain regions by machinery operators could be impossible in some cases, due to difficulties in accessing some areas. It is worth noting that despite the presence of invisible, occluded, and destroyed mounds, planting operations are carried out by maintaining a level of regularity even in irregular regions.

5. Conclusion

In this paper, we proposed a novel method for mound detection and counting on UAV images. Our method estimates the number of planting micro-sites through a combination of local object detection and global count estimation. Firstly, a local count by detection is performed by using a visual object detector. A global count estimation using global features is then performed to provide a final count. We conducted extensive experiments including, a comparison with state-of the art methods and an evaluation of human-level performance in recognizing mounds. We showed that our proposed method significantly outperforms both detection and density-based methods in the counting task.

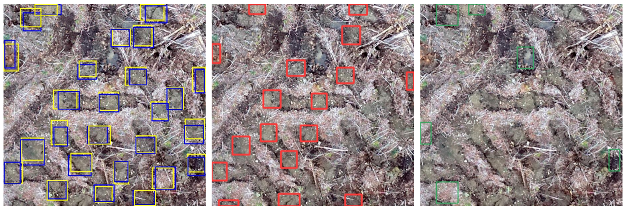


Figure 7. Patch example from block A to show missed detections due to visual detection model performance. Left, human annotations, center, YOLO detections, right, Faster R-CNN detections.

In future work, we aim to further improve the performance of visual counting by incorporating other types of UAV images. In particular, we recently obtained promising results by using thermal infrared (IR) imaging. In fact, thermal IR allows to efficiently exploit the difference in temperature between a given mound and surrounding background region. This is because mounds are warmer, being formed of mineral soil and bare of vegetation, while surrounding background regions between mounds are colder, given the presence of organic matter.

Acknowledgment

This work was supported by research grants from the Natural Sciences and Engineering Research Council of Canada (NSERC) [Discovery Grants RGPIN-2020-04937 and RGPIN-6656-2017].

References

[1] Lokesh Boominathan, Srinivas S S Kruthiventi, and R. Venkatesh Babu. Crowdnet: A deep convolutional network for dense crowd counting. In *Proceedings of the 24th ACM International Conference on Multimedia*, MM '16, page 640–644, New York, NY, USA, 2016. Association for Computing Machinery. 3

[2] Wassim Bouachir, Koffi Eddy Ihou, H. Gueziri, N. Bouguila, and N. Bélanger. Computer vision system for automatic counting of planting microsites using uav imagery. *IEEE Access*, 7:82491–82500, 2019. 3

[3] Pablo Chamoso, William Raveane, Victor Parra, and Angélica González. Uavs applied to the counting and monitoring of animals. In Carlos Ramos, Paulo Novais, Céline Ehrwein Nihan, and Juan M. Corchado Rodríguez, editors, *Ambient Intelligence - Software and Applications*, pages 71–80, Cham, 2014. Springer International Publishing. 2

[4] Antoni B. Chan, Zhang-Sheng John Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting peo-

ple without people models or tracking. *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2008. 2

[5] Ke Chen, Chen Change Loy, Shaogang Gong, and Tony Xiang. Feature mining for localised crowd counting. In *Proceedings of the British Machine Vision Conference*, pages 21.1–21.11. BMVA Press, 2012. 2

[6] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649, 2012. 3, 5

[7] M. Enzweiler and D. Gavrila. Monocular pedestrian detection: Survey and experiments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:2179–2195, 2009. 2

[8] Min Fu, P. Xu, Xudong Li, Qihe Liu, Mao Ye, and Ce Zhu. Fast crowd density estimation with convolutional neural networks. *Eng. Appl. Artif. Intell.*, 43:81–88, 2015. 3

[9] J. Hwang and Hwang-Soo Lee. Adaptive image interpolation based on local gradient features. *IEEE Signal Process. Lett.*, 11:359–362, 2004. 2

[10] H. Idrees, Imran Saleemi, C. Seibert, and M. Shah. Multi-source multi-scale counting in extremely dense crowd images. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2547–2554, 2013. 2

[11] J. Ilao and M. Cordel. Crowd estimation using region-specific hog with svm. *2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–5, 2018. 2

[12] Naveed Ilyas, Ashfaq Ahmad, and Kiseon Kim. Casa-crowd: A context-aware scale aggregation cnn-based crowd counting technique. *IEEE Access*, 7:182050–182059, 2019. 3

[13] H Jemaa, W Bouachir, B Leblon, and N Bouguila. Computer vision system for detecting orchard trees from uav images. *ISPRS-International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43:661–668, 2022. 2

[14] A. Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84 – 90, 2012. 3, 4

[15] Victor Lempitsky and Andrew Zisserman. Learning to count objects in images. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010. 3

[16] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. pages 1091–1100, 2018. 5, 6, 7

[17] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017. 4

[18] A. Marana, L. 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)*, pages 354–361, 1998. 2

[19] K. Mikolajczyk, A. Zisserman, and C. Schmid. Shape recognition with edge-based features. In *Proceedings of the British Machine Vision Conference*, pages 79.1–79.10. BMVA Press, 2003. doi:10.5244/C.17.79. 2

[20] Majid Nikougoftar Nategh, Ahmed Zgaren, Wassim Bouachir, and Nizar Bouguila. Automatic counting of mounds on uav images: combining instance segmentation and patch-level correction. In *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 375–381, 2022. 3

[21] Daniel Oñoro-Rubio and Roberto J. López-Sastre. Towards perspective-free object counting with deep learning. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 615–629, Cham, 2016. Springer International Publishing. 3

[22] N. Paragios and Visvanathan 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*, 1:I–I, 2001. 2

[23] V. Pham, Tatsuo Kozakaya, Osamu Yamaguchi, and R. Okada. Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3253–3261, 2015. 2, 3

[24] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *ArXiv*, abs/1804.02767, 2018. 3, 5, 6, 7

[25] C. Regazzoni and A. Tesei. Distributed data fusion for real-time crowding estimation. *Signal Process.*, 53:47–63, 1996. 2

[26] Shaoqing Ren, Kaiming He, Ross B. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015. 5, 6, 7

[27] Olga Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Zhiheng Huang, A. Karpathy, A. Khosla, Michael S. Bernstein, A. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015. 5

[28] Biyun Sheng, Chunhua Shen, Guosheng Lin, J. Li, Wankou Yang, and Changyin Sun. Crowd counting via weighted vlad on a dense attribute feature map. *IEEE Transactions on Circuits and Systems for Video Technology*, 28:1788–1797, 2018. 3

[29] Mihran Tüceryan and Anil K. Jain. Texture analysis. In *Handbook of Pattern Recognition and Computer Vision*, 1993. 2

[30] Elad Walach and Lior Wolf. Learning to count with cnn boosting. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 660–676, Cham, 2016. Springer International Publishing. 3

[31] Chuan Wang, Hua Zhang, Liang Yang, Si Liu, and Xiaochun Cao. Deep people counting in extremely dense crowds. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, page 1299–1302, New York, NY, USA, 2015. Association for Computing Machinery. 3

[32] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional

neural networks. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 833–841, 2015. 3

[33] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 589–597, 2016. 3, 6, 7

[34] Bingyin Zhou, M. Lu, and Yonggang Wang. Counting people using gradient boosted trees. *2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*, pages 391–395, 2016. 2