Computers and Electronics in Agriculture 217 (2024) 108485

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

GrainPointNet: A deep-learning framework for non-invasive sorghum panicle grain count phenotyping

Chrisbin James [a,*], Daniel Smith [a], Weigao He [b], Shekhar S. Chandra [b], Scott C. Chapman [a,*]

[a] School of Agriculture and Food Sustainability, The University of Queensland, Brisbane, Australia
[b] School of Electrical Engineering and Computer Science, The University of Queensland, Brisbane, Australia

ARTICLE INFO

Keywords:
Sorghum
Highthroughput phenotyping
Deep learning
Point clouds

ABSTRACT

Grain count is an important trait in sorghum because it is highly correlated to the potential yield. By accurately phenotyping the number of grains per panicle, farmers and agronomists can better monitor crop development. Additionally, mapping the spatial variability of grain count can help identify areas of the field with higher or lower potential yields, allowing for targeted management strategies. This study introduces a method for predicting grain count for sorghum panicles by employing a deep learning-based regression framework for point clouds and Red Green Blue (RGB) images. The framework integrates global features derived from a point cloud model of the panicle and grain counts detected from a sequence of RGB images. The models were evaluated on a paired dataset of point cloud models and RGB images collected for 147 sorghum panicles, which included a variety of panicle structures and grain counts. The point cloud models were constructed via a proximal structure-from-motion-based photogrammetry workflow. The model uses PointNet as the backbone network for processing the point clouds and YoloV5 for detecting grains from RGB images. Following the grain detection step, a scaled dot product attention module is integrated into the network to process the grain counts obtained from the RGB image sequence. Finally, the global features for the point cloud model and the grain counts are combined to predict the total grain count for the panicle. Furthermore, the models are also evaluated on downscaled low-resolution point clouds to assess their potential to be adapted in the future for point cloud models for panicles acquired in the field. The models were able to predict grain counts for the high-resolution point cloud dataset with a mean absolute percent error of 6.5% and 6.8% for the low-resolution point cloud dataset. The results serve as a proof of concept to demonstrate the viability of using a multimodal approach based on point clouds and RGB images to estimate grain count per panicle. Additional enhancements to the model like the inclusion of a module to register the point cloud and the RGB images, and evaluating more point cloud backbone networks can help further strengthen the method.

1. Introduction

Grain count estimation is a key component of plant phenotyping and precision agriculture systems. Grain counts help explain the spatial variability of crops in fields (Chung et al., 2016) and are a key component of yield (Andrade et al., 1999; Van Oosterom and Hammer, 2008). Accurate measurement of several crop traits like head density and vegetation indices, via non-invasive remote sensing technologies, has been made possible due to the success of deep learning-based image analysis algorithms (Jiang and Li, 2020). However, grain count estimation for crops still widely relies upon threshing-based methods. The crops are either threshed manually or via a thresher (Fu et al., 2018), to separate the grains from the crop panicle/head. Following the separation of the grains, the grains need to be cleaned and segregated from the 'grain heap' produced by the threshing process. The grain heap is comprised of full grain, straw, and grain impurities, which are usually processed via a pneumatic-sorting cleaning system to segregate the grains (Uhl and Lamp, 1966). These grain cleaning systems use a combination of sieving, followed by pneumatic transport to separate the grains, which involves using an air stream to separate and sort the grains from the waste. Pneumatic segregation leverages the aerodynamic properties of materials in the grain heap, specifically the critical velocity (lift velocity), to lift the chaffy and dusty waste material, while separating the grain by moving it downward (Panasiewicz et al., 2012). Finally, after the separation and cleaning of grains, they can be processed through a grain counting system. Grain counting systems either use photoelectric/impact-based methods (Zanke et al., 2015), or image

* Corresponding authors.
E-mail address: chris.james@uq.edu.au (C. James).

https://doi.org/10.1016/j.compag.2023.108485
Received 7 March 2023; Received in revised form 25 October 2023; Accepted 28 November 2023
Available online 21 December 2023

Acronyms

CNN	Convolutional Neural Network
FPS	Farthest-Point Sampling
mlp	Multi-Layer Perceptron
RGB	Red Green Blue
t-net	transformation network
UAV	Unmanned Aerial Vehicle
VCS	View Count Sequence

processing-based algorithms for counting grains. Image processing-based grain counting algorithms can be further classified into classical image processing algorithms and deep-learning-based models. Classical image processing-based grain counting algorithms rely upon thresholding or binarization (Zhao and Li, 2009; Gong et al., 2018), while deep learning models rely on Convolutional Neural Network (CNN) (Albawi et al., 2017) – based object detection algorithms to detect grains (Wei et al., 2020; Velesaca et al., 2020).

Deep learning algorithms have been widely applied in agriculture for panicle and head detection in various crops. For example, in the case of wheat, Gong et al. (2020) modified YOLOV4 (Bochkovskiy et al., 2020) and combined it with YOLOV3's (Redmon and Farhadi, 2018) detection head to build a lightweight model for real-time wheat head detection, similarly, Khaki et al. (2022) propose a modified MobileNetV2 (Sandler et al., 2018) for real-time detection, and Fourati et al. (2021) propose a combination of Faster R-CNN (Ren et al., 2015) and EfficientDet (Tan et al., 2020) for a robust detection model. Furthermore, in addition to detection-based models, regression-based models have also been applied for maize panicle counting. Lu et al. propose TasselNet (Lu et al., 2017) and TasselNetv2 (Xiong et al., 2019), a CNN-based density map regression model, trained via dot annotated images. Zou et al. (2020) provide a comparison between detection and regression-based methods. In addition to wheat and maize, sorghum is another crop where deep learning models have been applied for panicle detection. Wei et al. (2020) and Ghosal et al. (2019) propose detection-based models, furthermore, due to the relatively simpler panicle structure of sorghum, semantic segmentation models have also been trained with panicle boundary labelled images (Lin and Guo, 2020; Malambo et al., 2019b). Due to the popularity of deep learning models in crop panicle detection, there are several publicly available open-source datasets to facilitate this area of research. The global wheat head detection (GWHD)(David et al., 2021) dataset is comprised of 4700 labelled Red Green Blue (RGB) images with 193,634 labelled wheat heads made openly available for benchmarking wheat head detection models. Ghosal et al. (2019) released a sorghum head detection dataset with 1440 images of annotated sorghum row plots, and Lu et al. (2017) released the maize tassels counting (MTC) dataset with 361 dot annotated field images. Grain detection, on the other hand, when compared to panicle detection, has garnered relatively less attention from deep learning models. This is primarily because, crops like sorghum and wheat have grains hidden within their panicles, which are occluded and cannot be observed via RGB images of their surface, furthermore, the size and the shape of the panicle must also be considered while estimating the grain count, which also cannot be accounted for by images. Nonetheless, image-based deep learning algorithms have been successfully used for counting grains on certain crops with 'simpler' panicle structures non-invasively. For example, in the case of rice panicles, Deng et al. (2021) applied Faster R-CNN (Ren et al., 2015) to detect grains, and Gong and Fan (2022) developed a CNN-based model to segment the centre regions of grains, followed by counting connected regions to estimate grain count. Corn is another example of crops where deep learning models have been used for in-situ grain detection and counting, Khaki et al. (2020) propose a CNN-based model to detect corn kernels from images and locate their centre locations. Grain count estimation for crops like sorghum and wheat, as mentioned earlier still largely rely upon invasive and destructive methods.

Some methods have explored the use of medical imaging techniques to completely scan the internal structure of the panicles to locate every individual grain. Schmidt et al. (2020) used X-ray computed tomography to fully scan 200 wheat heads grown under drought and heat-stressed conditions, and propose a pipeline to reconstruct the complete 3D structure of the head to locate grains, and estimate their shape and size. The pipeline was able to estimate the grain count with an R-squared value of 0.99 for wheat heads developed in drought-affected conditions, and R-squared value of 0.7 for heads developed in drought and heat-related conditions. Similarly, Li et al. (2020) have also applied X-ray computed tomography to scan 55 sorghum heads, belonging to five characteristically distinct botanical races, to reconstruct the entire panicle and identify individual grains. The reconstructions of the sorghum heads were able to estimate the net grain counts with an R-squared value of 0.98. Also, Hughes et al. (2017) explored the application of micro-computed tomography scans to reconstruct wheat and sorghum heads and were able to accurately estimate grain counts. Although the aforementioned methods can estimate the grain counts with high precision, the primary objective of the studies was not grain count estimation, but to explore the morphology of the grains and how different environmental factors affected grain size, volume, and count. Additionally, given the exhaustive nature of X-ray scans, retrieving the grain counts from fully reconstructed 3D models of panicles is a relatively trivial task. More importantly, X-ray scans require expensive hardware and intricate setup, furthermore, are unfeasible to be deployed in in-situ conditions on the field. Recent work by Freeman et al. (2022) proposes a pipeline to reconstruct a high-resolution 3D point cloud model of sorghum panicles. A robotic arm mounted with a stereo camera is used to image the panicle from different directions, followed by masking each grain in individual images via CenterMask (Lee and Park, 2020). A pose graph is constructed based on grain masks across images to create a high-quality point cloud. Finally, density-based spatial clustering (DBSCAN) (Ester et al., 1996) is applied to segmented grain pixels projected on the 3D point cloud to locate and count the individual grains on the surface. Although the results show that, for the authors' dataset, there is a strong linear correlation (R-squared value of 0.875) between the number of grains estimated on the surface and the true grain count (panicle grain count is estimated to be roughly 1.6 times the observed surface grain count), there is a significant difference between the grains observed on the surface and true grain count especially as the true grain count increases, as sorghum panicles have internal hidden grains that cannot be seen from an outside view. Additionally, this observed strong linear correlation might not necessarily hold up for panicles with more open and varied shapes and different grain sizes.

Here we propose a pipeline of deep learning algorithms, which are designed to estimate grain counts from images and 3D point clouds of sorghum panicles. We introduce a framework that uses a combination of YOLOV5 (Jocher et al., 2021) to detect grains from RGB images and a modified version of PointNet (Qi et al., 2017) to process point clouds of sorghum panicles.(1) PointNet is used to extract features from the point cloud model of the panicle; (2) Yolov5 is used for detecting grains from a sequence of images of the panicle to create a grain count vector; (3) The grain count vector is processed through a scaled dot product attention block; (4) Features extracted from the point cloud are combined with the grain count vector to predict the panicle grain count. Our dataset is comprised of carefully selected sorghum panicles, which represent a reasonable range of structures and sizes. We test our pipeline on high-resolution* point clouds to validate the viability of estimating grain counts from 3D surface scans. Furthermore, we tested our pipeline on synthetically downscaled, low-resolution point clouds to check the feasibility of adapting our pipeline to process point clouds captured on the field with in-situ constraints for future work.

Fig. 1. 2 views of a sorghum panicle which was deformed during transportation, the first view shows that the panicle was flattened along a side, the second view shows the panicle was folded along the centre.

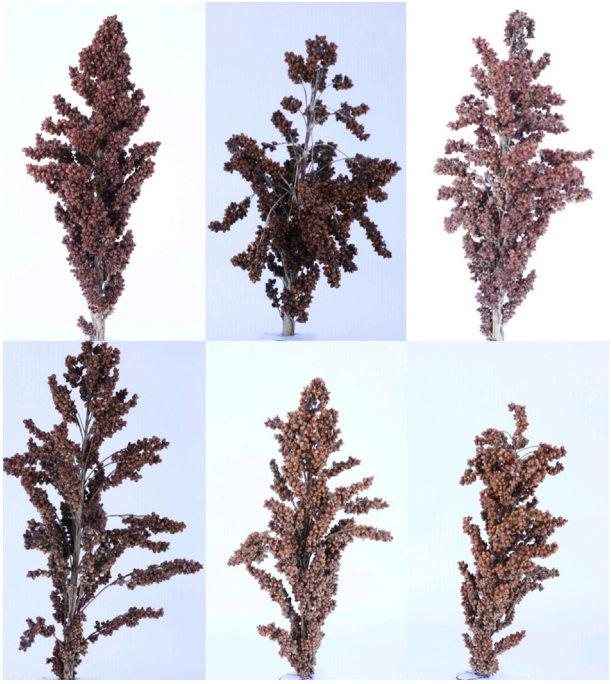


Fig. 2. Panicle shapes after transportation.

2. Materials and methods

In this section, we describe our dataset, and pipeline that uses a combination of Yolov5 (Jocher et al., 2021) and a modified PointNet (Qi et al., 2017) for grain count estimation from images and point cloud models of sorghum panicles.

2.1. Dataset

2.1.1. Panicle collection

Our dataset is comprised of two sets of sorghum panicles collected in April 2021 and April 2022 from two different breeding trials located at Allora, Queensland, Australia. In 2021, a total of 98 panicles were collected from 12 different plots, belonging to 12 different genotypes. From every plot roughly 8 different heads were collected (4 from high-density planting and 4 from low-density planting), in increasing order of size, to sample a wide range of grain counts. The heads were then transported in cloth bags to the lab to be imaged. We observed that, that transporting heads in such a manner "closes" the panicle's natural shape, especially panicles with more open structures, Fig. 1 shows the shape of a panicle that was deformed during transportation. In 2022, a total of 60 panicles were collected from 12 different plots, belonging to 12 different genotypes. From every plot, 5 different heads were collected, in increasing order of size. Additionally, we focused on collecting panicles with more "open" and varied structures. Fig. 2 shows panicles whose "open" structures were preserved after transportation (refer to section 5 in supplementary materials section).

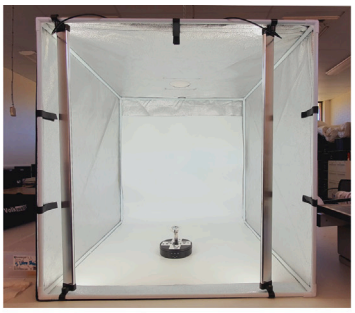
2.1.2. Image acquisition

The panicles were imaged within a cube-shaped tent, and the walls inside the tent were fitted with white sheets. Additionally, there were two white fluorescent LED strips attached vertically to the sides of the opening face of the tent. The strips were slightly angled inwards, facing a rotating dial, in which the panicles were mounted. This lighting configuration was in place to create a consistent and clear illumination condition for the panicles to be photographed. Fig. 3(a) displays the imaging tent for the panicles. The panicles were photographed while being slowly rotated, with more than 90% overlap between every consecutive image.

A Canon EOS 6D camera was used to photograph the panicles. The camera was mounted on a tripod with an adjustable mount and was placed roughly 60 cm away from the imaging tent. The images were captured with a 50 mm lens - f/9 aperture, with an ISO of 400 and a shutter speed of 1/100s. Two sets of images of the panicles were taken, where each set was comprised of images taken while the panicle completed a full rotation on the dial. The tripod mount was shifted roughly about 20 cm and the camera was realigned before taking the second set of images, this was done to image the panicle from an upper and a lower view to capture more detail for the 3D reconstruction of the panicle. On average about 75 ~ 80 images were captured for most panicles, but for panicles with more open and complex structures, about 140 images were captured by slowing down the speed of the rotating dial to ensure the panicle stayed still while imaging them. Fig. 3(b) shows the different views captured for the panicle, and Fig. 3(c) shows the different views from where the panicle was imaged.

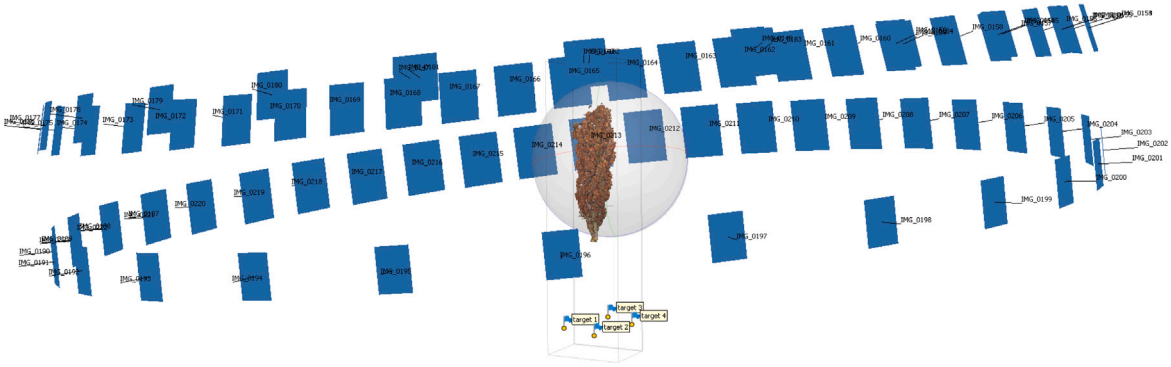
2.1.3. Grain Detection for RGB images

For detecting individual grains in images (later to be used in Agisoft for 3D reconstruction), an 'm' (medium) version of Yolov5 (Jocher et al., 2021) was trained on a dataset of manually annotated 512 × 512 "grain" patches extracted from the images (with the default Yolov5 augmentation pipeline and parameters). The dataset contained 100 annotated patches, out of which 85 patches were used for training and 15 were used for validation. The input resolution for the model was 512 × 512 and was trained for 500 epochs with the default parameters and augmentation pipeline. Evaluating the model on the validation set, the model had a mean average precision of 0.94 (at 0.65 intersection over union threshold). Fig. 4(a) shows a sample of the training images for the model, and Fig. 4(b) shows the precision–recall curve for the model on the validation set. The native resolution of each image was 5472 × 3648. For detecting grains in images, each image was divided into cut into 512 × 512 patches with 15% vertical and horizontal overlap before being processed by the model. After running inference on the individual patches, the detections were combined across the patches for the entire image. Fig. 5(a) shows how the image was divided into patches, and 5(b) shows the combined grain detections for an entire



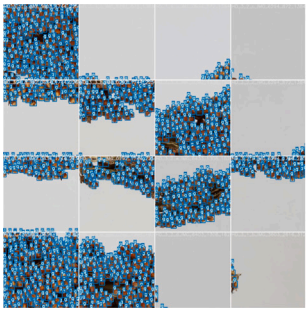
(a) Imaging tent for panicles

(b) Images captured from different angles

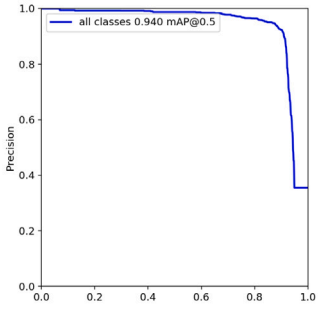


(c) Camera views for imaging the panicle

Fig. 3. Rotating dial for mounting panicles while imaging.



(a) Training set images



(b) Precision-Recall curve for validation set

Fig. 4. Yolov5 — 2D grain detection model.

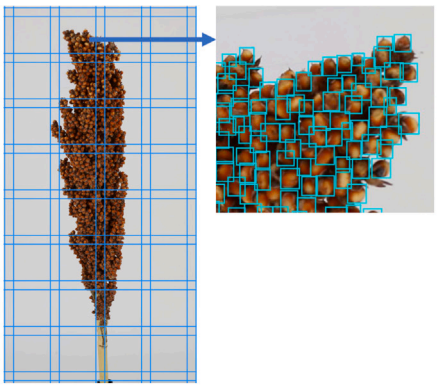
image. Fig. 5(c) shows the variation in the observable number of grains in images from different views/directions for a subset of our dataset. Although there appears to be a moderate positive linear correlation between grain counts observed in images and total grain count, the is a significant variation between observed counts from different views, and more importantly, the degree of variation in observable counts between different panicles is also dependent on the shape of individual panicles.

2.1.4. High resolution — point cloud reconstruction

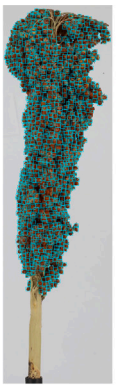
For constructing the high-resolution* point cloud of the panicle, before processing the images through Agisoft metashape (Agisoft, 2020), the YoloV5-grain detection model is used to detect grains in every image. Followed by, colouring the centre location of every predicted grain bounding box with a fluorescent green circle (RGB: 102, 255, 0) to highlight the centre location of all the grains. Fig. 6 shows the images used for 3D reconstruction with highlighted grain centres (Please refer to section 5, in the supplementary section for more details about the photogrammetry pipeline).

Upon visual inspection of the point cloud outputs, the panicles were reconstructed with high quality as shown in Fig. 7. Close-up inspection of the dense cloud model depicted in Fig. 7(a) shows that all the grains can be identified individually, additionally, all the grain centres are highlighted in green. Fig. 7(b) further demonstrates the quality of the dense cloud reconstruction, where the texture of the tape, the patterns on the marker stickers, and the heart-shaped symbol on the circular disc are clearly visible.

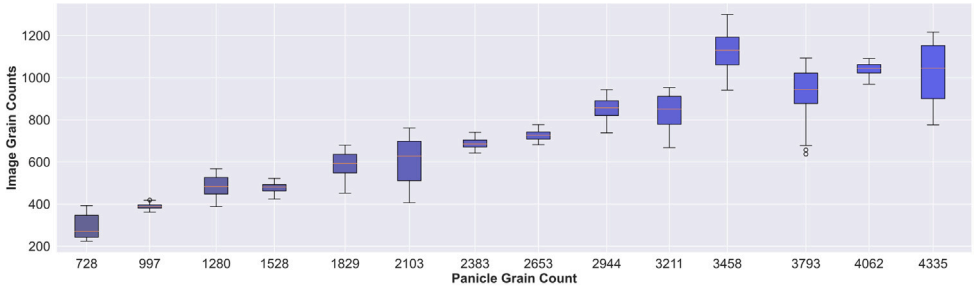
The final dense cloud point counts approximately range between one to nine million and is clearly a lot of points to be directly processed. This is the reason why the center regions of the grains were highlighted before constructing the dense cloud models, the centre locations of the grains can be segmented by simply thresholding the point cloud to separate green points. The segmented set of points represents individual grain centres as small clusters of separated points, and collectively are a good descriptor of the entire shape of the panicle. Finally, iterative Farthest-Point Sampling (FPS) is applied to segmented points to 'sample' 10000 points to get the final point cloud model. FPS ensures better



(a) Predicting on image patches



(b) Combined predictions for entire image



(c) Variation in the observable number of grains on a panicle from different views

Fig. 5. Grain detection for images.

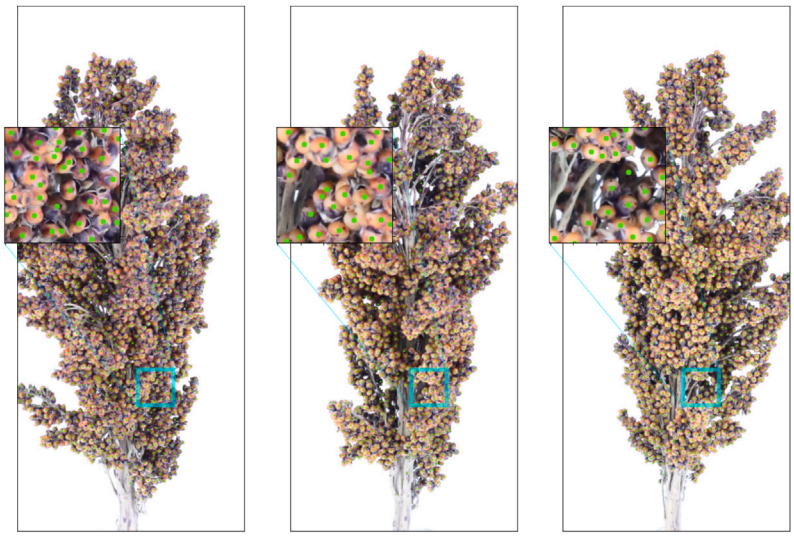


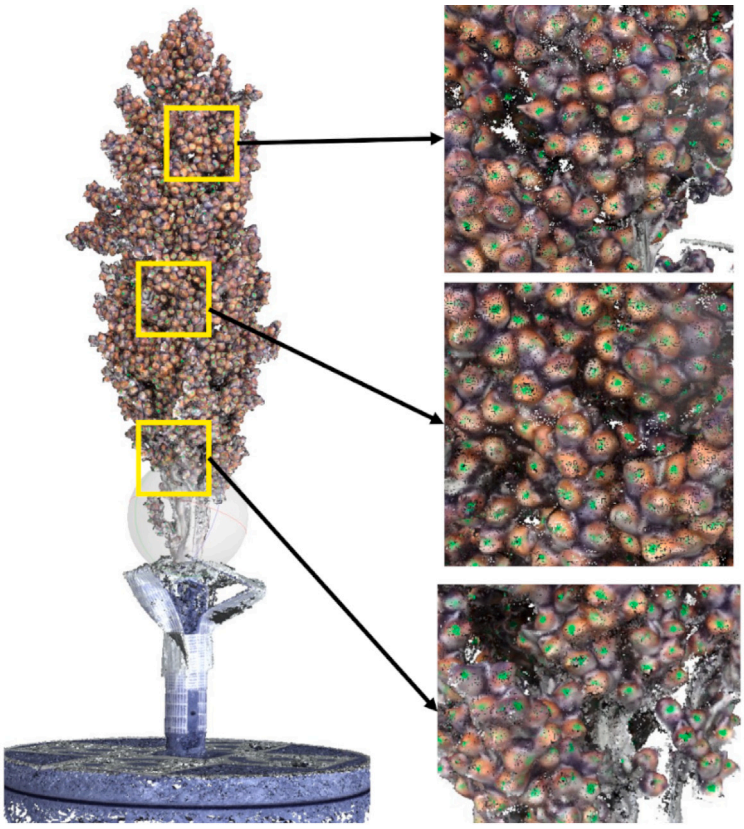
Fig. 6. Images used for 3D reconstruction with highlighted grain centres.

coverage of the entire point set when compared to random sampling, given the same number of centroids.

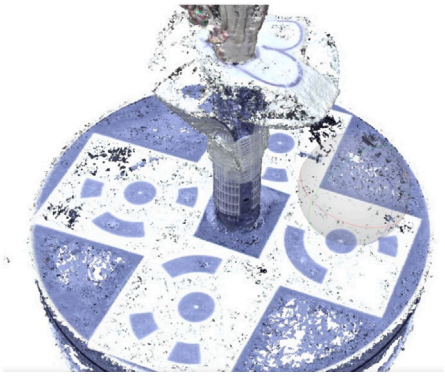
Fig. 8 shows how the final point cloud is derived from the dense cloud model, Fig. 8(b) shows the segmented grain centre points after thresholding the dense cloud model, and Fig. 8(c) shows the FPS sampled point cloud model from the segmented point cloud, zoomed in regions in Fig. 8(c) show small clusters of points belonging to different grains. For large panicles, with high grain count and high surface grain density, the final point cloud models have larger volumes(volume encompassing the point cloud model) and low-density clusters, point clouds with a 'loose' spread of points. The final point cloud models of small panicles with small grain counts, on the other hand, have smaller volumes and clearly spread high-density clusters, a 'tight' spread of points. Fig. 9 shows comparisons between a high grain count panicle and a low grain count panicle, Fig. 9(a) shows the final point cloud model of a panicle with 4234 grains, Fig. 9(b) shows the final point cloud model of a panicle with 838 grains.

2.1.5. Low resolution — point cloud construction

Although there are various imaging solutions available to create detailed 3D reconstructions of panicles in labs and 'controlled' environments, high-quality 3D reconstruction of panicles in field situations with in-situ constraints is very difficult. Especially, when trying to



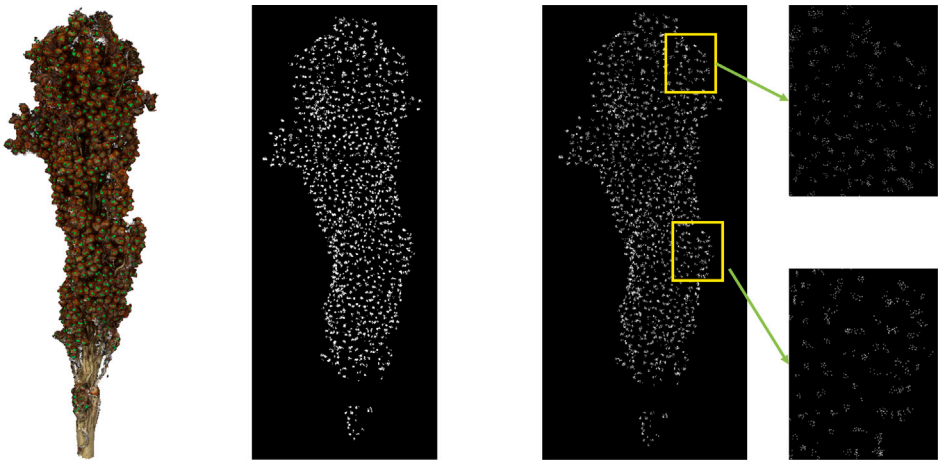
(a) Dense cloud model from 3D panicle reconstruction - Individual grains are clearly visible, and the grain centres are highlighted in green



(b) Zoomed in view of imaging base - The imaging base is reconstructed with high quality, all patterns are clearly visible on the markers, tape, and the plastic ring shown figure 26b

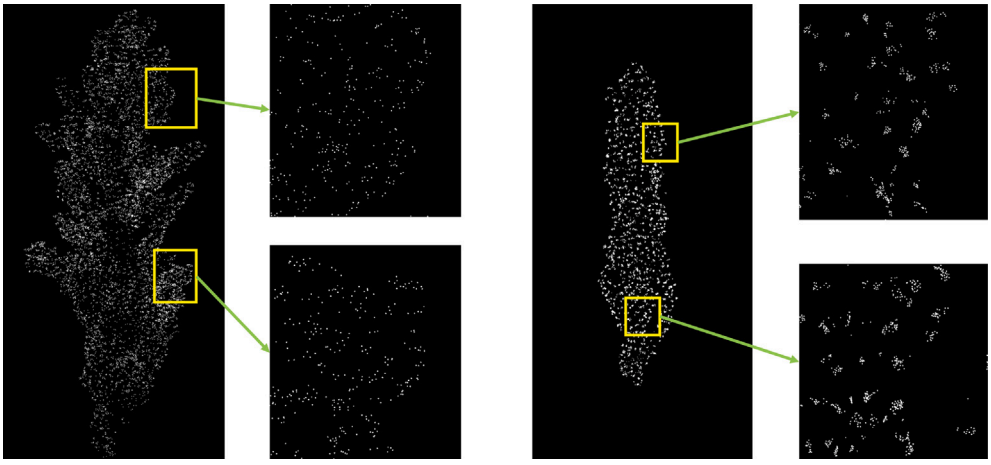
Fig. 7. 3D Dense cloud model.

account for tiny objects like grains. Nonetheless, advancements in portable imaging and LiDAR sensors, in combination with the increasing popularity of drone-based imaging, have facilitated the 3D reconstruction of canopies and panicles. Liu et al. (2021) propose a Unmanned Aerial Vehicle (UAV)-based image acquisition pipeline for the 3D reconstruction of maize and soybean canopies. Chang et al. (2021) propose a UAV-based sorghum panicle reconstruction and segmentation pipeline. Although the reconstruction quality and resolution of these methods are far inferior when compared to imaging techniques for isolated and controlled environments, UAV and portable sensor-based imaging solutions are the only current viable approaches for high-throughput 3D phenotyping techniques. We simulate (relatively) low-resolution 3D point cloud models of panicles in our dataset. Following the creation of the dense cloud model in Agisoft, 3D mesh models from the dense cloud outputs are constructed, with the face count option set to 'low'. The exported meshes, even with the low setting, on average the exported meshes had 120000 faces, which still contains considerably more detail, especially when compared to the quality 3D outputs that portable LiDAR scanners can produce. The outputs are triangular meshes stored as wavefront objects (.OBJ files). Fig. 10 compares three mesh models, Fig. 10(a) shows a low-resolution mesh output interpolated from the dense cloud from our Agisoft photogrammetry pipeline, Fig. 10(b) shows the same mesh decimated to 5000 faces to simulate a low-resolution mesh which would be obtained from a portable LiDAR scanner, Fig. 10(c) shows a 3D mesh of a different panicle imaged via Scaniverse (AI, 2022), on the 3rd gen 11-inch iPad pro (equipped with a LiDAR scanner). Upon visual inspection of the decimated mesh model and the model scanned from the iPad,



(a) Dense cloud model of panicle (b) Segmented grain centre points (c) Farthest point sampled point cloud (10000 points)

Fig. 8. Grain centre point cloud model.



(a) High grain count panicle point cloud model - 4234 grains (b) Low grain count panicle point cloud model - 838 grains

Fig. 9. FPS point clouds for a high grain count and low grain count panicle.

both meshes are relatively similar in quality and level of detail. After exporting the 'low' quality mesh in agisoft, all the meshes are decimated via Python script using the blender (Community, 2018) 3.1 API, with the number of target faces set to 5000. Finally, iterative FPS is applied to the surface of the decimated mesh to sample points to create the 'low' resolution point cloud model. Fig. 11 shows how the low-resolution point cloud is created. Fig. 11(a) shows the dense cloud model created in Agisoft, Fig. 11(b) shows the mesh model constructed from the dense cloud (without texture), Fig. 11(c) shows the mesh decimated to 5000 faces, Fig. 11(d) shows the point cloud sampled from the mesh surface via FPS.

2.2. GrainPointNet

2.2.1. PointNet

Fig. 12(a) shows the architecture of our proposed framework GrainPointNet to predict grain counts. The network was implemented via the PyTorch (Paszke et al., 2019) library. The original PointNet (Qi et al., 2017) is used to process the point cloud inputs to extract 'global' features from the point cloud. As described in Section 2.1, both the low-resolution and high-resolution datasets have 10000 points for every panicle point cloud. PointNet is comprised of blocks that apply a sequence of 1D convolutional operations (a shared dense network) to all the points in the input point cloud to derive feature maps. In each block, before applying the shared dense network, the input points/features are processed through transformation network (t-net). The t-net module 'realigns' the input features by multiplying them with a learnable transformation matrix. The transformation matrices are derived from the input features by processing them through a 'mini-PointNet' network, which applies the same order of 1D convolution operations and linear layers to the input features, followed by projecting them to a $n \times n$ transformation matrix (a 3×3 matrix for the input transform, and a 64×64 matrix for the feature transform block). Fig. 12(b) shows an expanded diagram of the t-net module. Finally, after processing the points through the shared dense layers, a max-pooling operation is applied over the collective feature maps for each set of points to extract the final global feature block. The max-pool layer acts as a symmetric set function, which makes the network invariant to the permutation (order) of the points.

2.2.2. VCS — View count sequence

In addition to the point cloud input to the networks, a vector of numbers is also passed as a separate input. This vector is comprised

(a) Mesh exported from Agisoft (b) Decimated mesh-5000 faces (c) Mesh scanned on iPad pro

Fig. 10. Comparing mesh models from our photogrammetry pipeline and iPad's LiDAR scanner.

of the number of observable grain counts from different 'views' of the panicle. A view is a photograph of the panicle, from the set of images used for the 3D reconstruction of the panicle, as described in 2.1.2, the panicle is imaged from all directions. So the set of original images can be considered as a set of cameras placed around the panicle in a circular arrangement. A View Count Sequence (VCS) is a set of observable grain counts from (consecutively ordered) equispaced cameras, sampled from a set of cameras surrounding the panicle. For the high-resolution dataset, the length of the VCS is 10 (i.e. 10 equally spaced cameras around the panicle). For the low-resolution dataset, the length of the VCS is 3. Fig. 13 shows how images are sampled to construct a view sequence, Fig. 13(a) shows the images selected from a circular set of cameras, Fig. 13(b) shows images in a VCS of length 10 and Fig. 13(c) shows images in a VCS of length 3.

The VCS is transformed and 'combined' (described in Section 2.2.4) with features extracted from the point cloud of the panicle via PointNet (Qi et al., 2017) before predicting the final grain count of the panicle. The primary objective for using VCS along with point clouds is to test the viability of a framework where features from 2D images of the panicles (high res and relatively cheap to acquire) are combined independently with 3D models of panicles (low res and time-consuming to collect), to calibrate the grain count prediction from 3D models. In order to, account for grain density and distribution along the surface of the panicle. In our experiment, we simply use the number of observable grain counts as the feature for each image, but our proposed framework can be upgraded in the future by using deep learning models to process images for extracting 'task-specific' features and including a step to register and align point cloud models with the 2D images.

2.2.3. Self attention-based VCS rescaling

The VCS is not directly fed into the network, the sequence is processed via the scaled dot product attention mechanism proposed by Vaswani et al. (2017). The motivation for using the scaled-dot-product attention is to extract permutation invariant features from the VCS. Since the input VCS vector, is highly subjective, due to the nature of its acquisition and definition. A set of equispaced cameras around a subject is determined by the position of the first camera, therefore, the VCS can be treated as a circular queue, where the first camera location will determine the order of the VCS. The attention-based rescaling module works as follows (besides the self-attention module, we also apply an augmentation strategy to the VCS described in Section 2.3.2). Firstly, the VCS is fed into a Multi-Layer Perceptron (mlp) as shown in 12(a), and the sequence is projected into a tensor twice its original length. For the high-resolution dataset, with a VCS of length 10, the mlp is comprised of linear layers with dimensions, 10-20-40-20 (as shown in the figure). For the low-resolution dataset, with a view count sequence of length 3, the mlp is comprised of linear layers with dimensions 3-6-12-6. Finally, the output of the mlp is split into equal tensors, where the first tensor is the query input and the second tensor is the key input for the scaled dot product attention, and the original count sequence is the value input. The query, key, and value inputs are treated as sequences of length 10 (and 3 for the low-resolution dataset) with an embedding dimension of size 1. The output of the scaled dot product attention with this query, key, and value combination is essentially a rescaled count sequence. Which is then 'combined' with the global features extracted from the point cloud model.

2.2.4. VCS — PointNet Feature Fusion

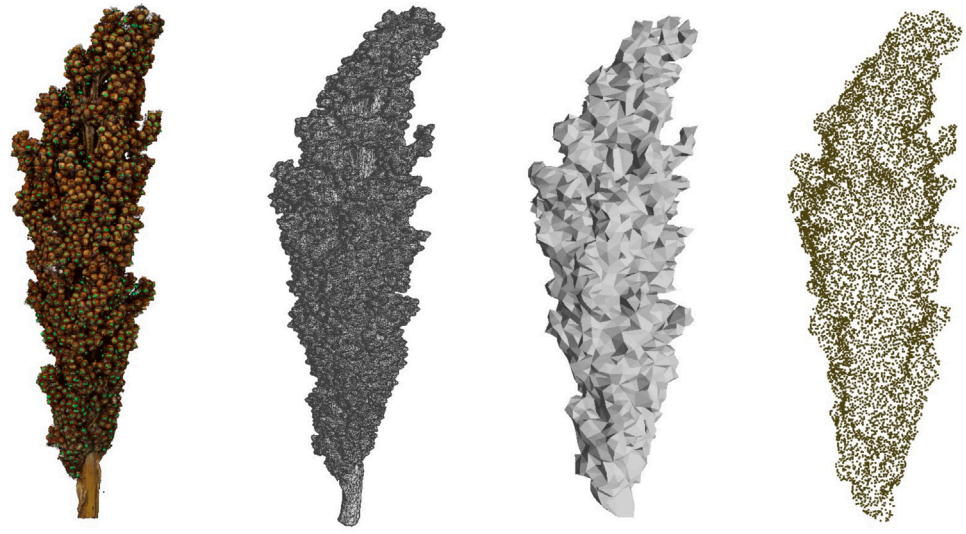
After extracting the global point cloud feature tensor and rescaling the VCS, a copy of the global feature tensor is fed into an mlp which projects it to a tensor of length (length of view count sequence) * 128. For the high-resolution dataset, the mlp is comprised of linear layers with dimensions 1024-1152-1280. For the low-resolution dataset, the mlp is comprised of linear layers with dimensions 1024-704-384. Followed by processing the feature tensor via the mlp, the resultant tensor is reshaped into a matrix of shape (length of VCS) $\times$ 128, which is then multiplied with the rescaled VCS tensor. This results in a final tensor with a length of 128. The global feature tensor is also processed by a separate mlp as proposed in the original PointNet architecture for classification, but, the mlp has been adopted for regression by replacing the last layer. In the original model, the final mlp is comprised of linear layers with dimensions 512-256-k (where k is the number of classes), for our regression task we have replaced the last layer with 2 layers, an mlp with linear layers 512-256-128-1 (dropout layers with prob 0.3 and 0.2 after the 512 and the 256 linear layers). The fused point cloud-view count sequence tensor is treated as a residual feature and is finally combined with the 128-length tensor through element-wise addition before the final count prediction as shown in Fig. 12(a). The smooth-L1 loss (introduced by Girshick, 2015) is used as the loss function for training the model (with the default parameters in PyTorch). For each element in the batch, the loss l_n is computed as described in Eq. (1) and is reduced as the average value for the batch. where x_n is the predicted value, y_n is the ground truth, and *beta* is the delta for Huber (Huber loss), as *beta* tends to zero smooth-L1 loss converges to L1 loss.

$$l_n = \begin{cases} 0.5(x_n - y_n)^2/beta, & \text{if } |x_n - y_n| < beta \\ |x_n - y_n| - 0.5 * beta, & \text{otherwise} \end{cases} \quad (1)$$

2.3. Model training

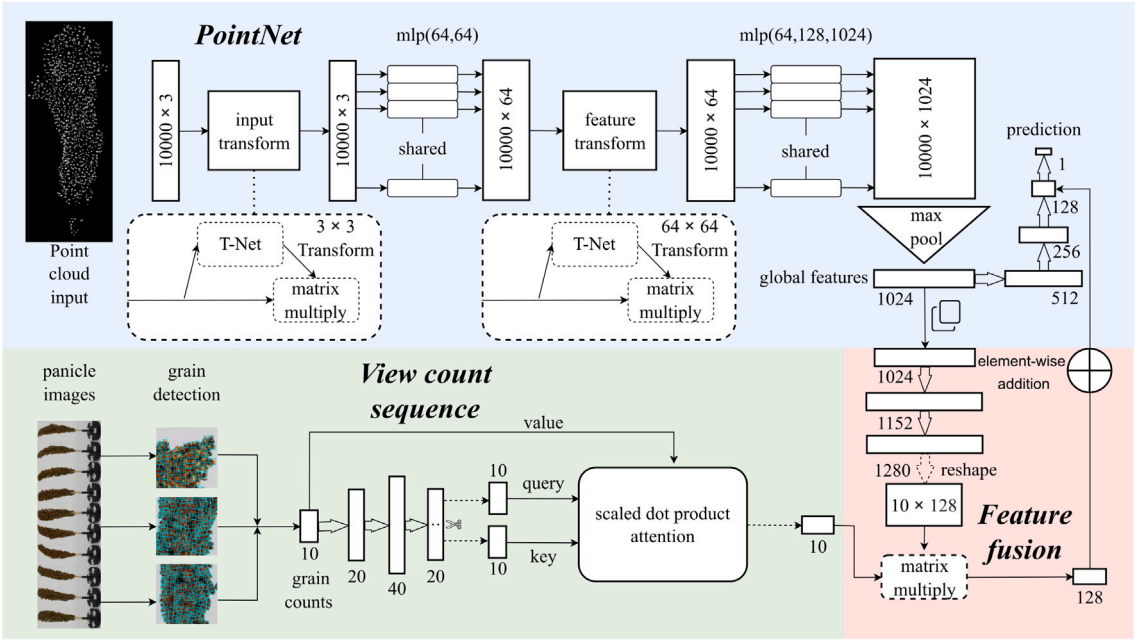
2.3.1. Dataset split

Our dataset is comprised of a collection of 147 sorghum panicles and the grain counts for the panicles roughly range between 700 to 4400. Fig. 14(a) shows a histogram of the grain count distribution in the dataset. Although the grain counts for the majority of the panicles in our dataset fall between the range of 2000 to 3500, our dataset also contains panicles with grain counts falling in ranges 3500–4400 and 1000–2000 grains to help account for variability across different ranges of counts. It is however worth noting that, we only had 4 panicles with grain counts of fewer than 1000 grains in our dataset. For model training and evaluating model performance, we split the dataset with a stratified 5-fold cross-validation configuration. For each dataset in our cross-validation split, 80% of the dataset was included in the training set and 20% of the dataset was selected for the validation set. The panicles were grouped into 5 1000 grain count bins (from 0–1000 to 4000-), followed by, uniformly sampling roughly the same number of panicles from each group for each validation set to create the cross-validation split. Fig. 14(b) shows the box plot for the grain counts for the validation sets for the cross-validation split.

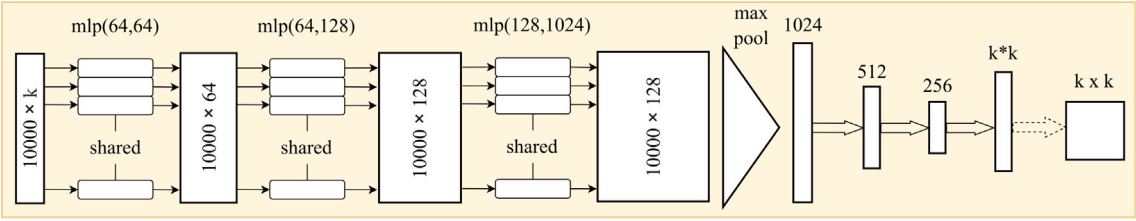


(a) Dense cloud (b) Mesh (c) Decimated Mesh (d) FPS Point cloud

Fig. 11. Creating low-resolution point cloud models.



(a) GrainPointNet Framework

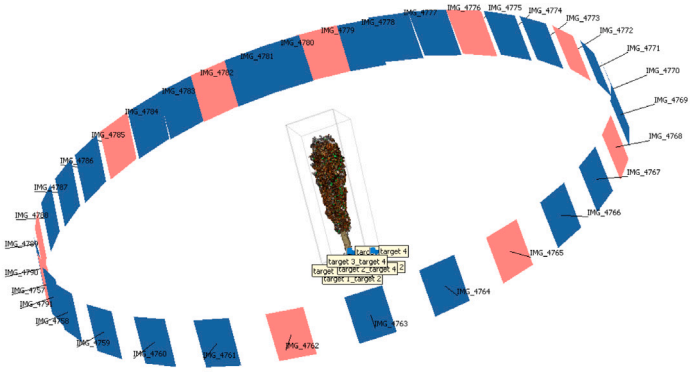


(b) t-net module for PointNet(Qi et al. 2017)

Fig. 12. Grain count prediction network.

2.3.2. Data augmentation

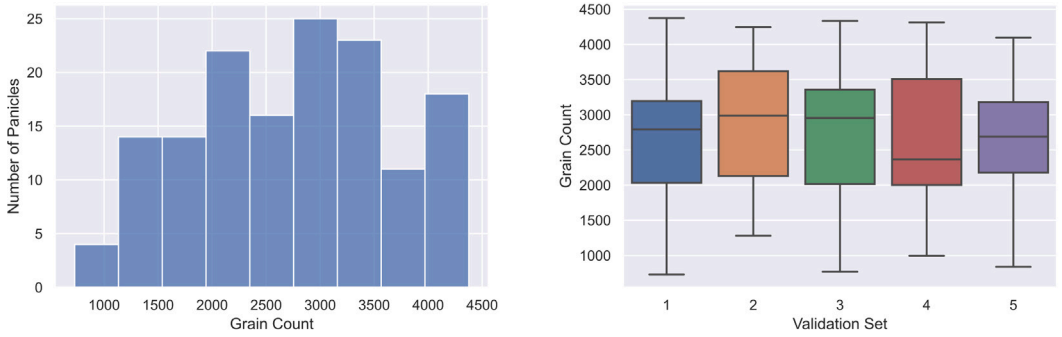
As suggested in the original PointNet (Qi et al., 2017) paper the learned representation of the point clouds should be invariant to rigid transformations like rotation and translation. Therefore, while training our model we apply the following sequence of transforms to the point clouds. Firstly, we translate the point cloud to the origin by subtracting the mean along each axis from the respective coordinates for all points in the cloud point cloud. Followed by, randomly rotating the point



(a) Images selected for a view sequence of length 10

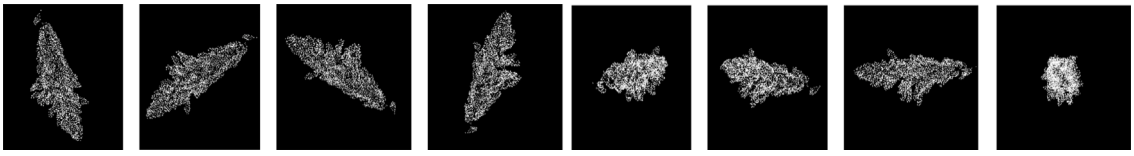
(b) 10 image view sequence (c) 3 image view sequence

Fig. 13. VCS for a sample panicle.



(a) Grain count histogram (b) Cross-validation split grain count box plots

Fig. 14. Dataset grain count distribution.



(a) Point cloud model rotated along XY-axis (b) Point cloud model rotated along XZ-axis

Fig. 15. Point cloud model rotation augmentation.

cloud either along the XY or XZ axis, where the angle for rotation is uniformly sampled between 0 - 2π radians. Finally, as the model should be invariant to input permutation, the input order of the points is randomly shuffled. Fig. 15 shows the different rotation configurations applied to a panicle point cloud model, along the XY (Fig. 15(a)) and XZ axis (Fig. 15(b)) at 72, 144, 216, and 288 degrees.

In addition to augmenting the point cloud, the VCS is also augmented. The VCS is considered a circular queue (since it is comprised of a set of views encircling the panicle), so during training, the VCS is also rotated randomly about a position. Furthermore, while imaging the panicle, if a complete circular set of views is comprised of 70 images, there are 70 possible view sequences to sample from. Therefore, a single panicle point cloud model is trained with a combination of multiple view sequence counts for the same ground truth (grain count).

3. Results

3.1. Training log

As described in Section 2.3.1, a five-stratified cross-validation split was used to divide the dataset to train the network. For each validation set split, the model was trained for 100000 epochs. The Adam (Kingma and Ba, 2014) optimizer with default parameters for the learning rate, $\beta 1$, and $\beta 2$ (0.001, 0.9, and 0.999) was used to train the network.

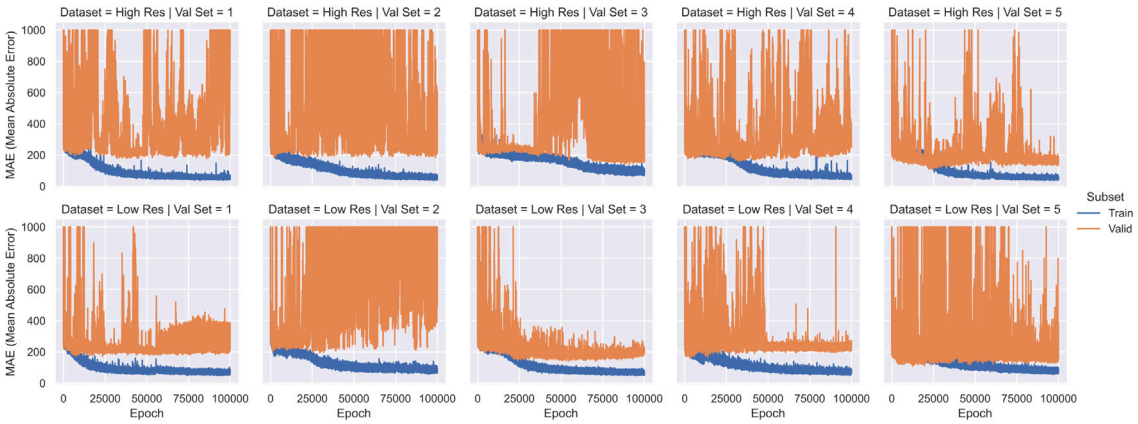


Fig. 16. Training and Validation loss curves.

Table 1
Model hyperparameters.

Hyperparameter	Value
# of Epochs	100 000
Optimizer	Adam
Learning Rate	0.001
$\beta 1$, $\beta 2$	(0.9, 0.999)
Batch Size	8
Point cloud size	10 000
VCS length (High-Resolution Dataset)	10
VCS length (Low-Resolution Dataset)	3

Table 1 gives a summary of the model hyperparameters for both the high-resolution and low-resolution datasets.

Fig. 16 shows the training and validation loss (Mean Absolute Error) for all the cross-validation subset models trained for both the low-resolution and high-resolution datasets (losses above 1000 have been cropped for better visualization). From the training loss curves, it is evident that the validation set loss is much more 'volatile' when compared to the training loss. This can be primarily attributed to the relatively small size of the datasets and the high learning rate. The error of the model for the validation set fluctuates frequently throughout training, increasing abruptly after improving, due to the interaction between the high learning rate and the diversity of samples in the validation set. But overall, the validation loss does decrease over time and the best model checkpoint achieves error rates comparable to those of the training set. It is also worth noting that the loss values for both sets of models trained on the low-resolution dataset and the high-resolution dataset are very comparable. When running inference on the validation set(during model training), no rotation augmentation is applied to the point cloud model, and the same VCS is used for each panicle data point. The best-performing model, model iteration with the lowest Mean Absolute Error is selected for each subset. Mean Absolute Error is described in Eq. (2), where x_i is the predicted value, where y_i is the ground truth, and N is the size of the dataset.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|x_i - y_i| \tag{2}$$

3.2. Test time augmentation

While running inference on the validation set (for testing), the predicted grain count was taken as the average prediction over a combination of point cloud inputs with different rotation configurations and different possible VCSs that could be sampled from the available set of images for each panicle (as described in Section 2.3.2). For each panicle, along with the default point cloud position, the point cloud would be rotated along the XY-axis and the XZ-axis at 72, 144, 216, and 288 degrees as shown in Fig. 15 (Test time augmentation was only applied to the point clouds and the VCS for the grain count prediction network, not for the detection model on the RGB images).

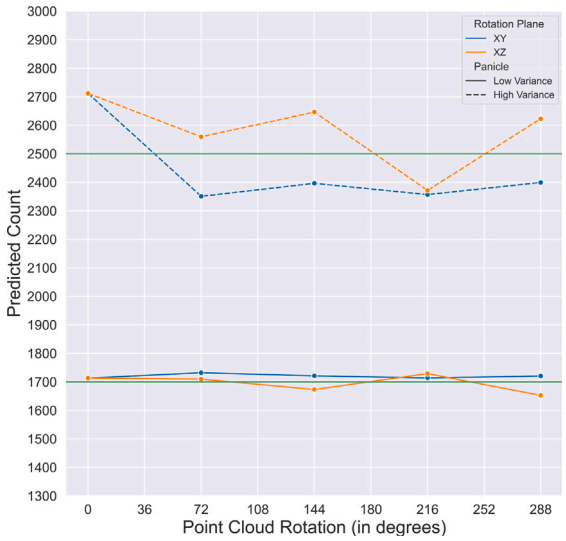


Fig. 17. High-Resolution dataset models: Test time augmentation — Rotation (Green line = ground truth reference).

Fig. 17 shows how the grain count prediction varies when rotating the point cloud model(trained on the high-resolution dataset) for two sample panicles when using the same VCS input. The model's prediction for one panicle (ground truth count — 2500) varies significantly with rotation along both axes, as shown in the figure. In contrast, the model's prediction for the second panicle (ground truth count — 1700) is relatively more consistent and invariant to rotation.

Fig. 18 shows how the grain count prediction varies when using different VCSs (trained on the high-resolution dataset) for a sample panicle when using the same point cloud model input. As mentioned in Section 2.1.2, the panicles were photographed to form a set of circular views, additionally, they were photographed from two different angles (high and low). The VCS as described in Section 2.2.2 is essentially a sequence of grain counts observable from a set of equispaced views/images encircling a panicle. VCS can be sampled from either the 'upper view' set of images or from the 'lower view' set of images. Although there is no considerable variation in prediction when using different VCSs, in certain cases, there is variation when using VCSs belonging to different viewing angles. This is due to the fact that depending on the shape of the panicle and the angle at which the

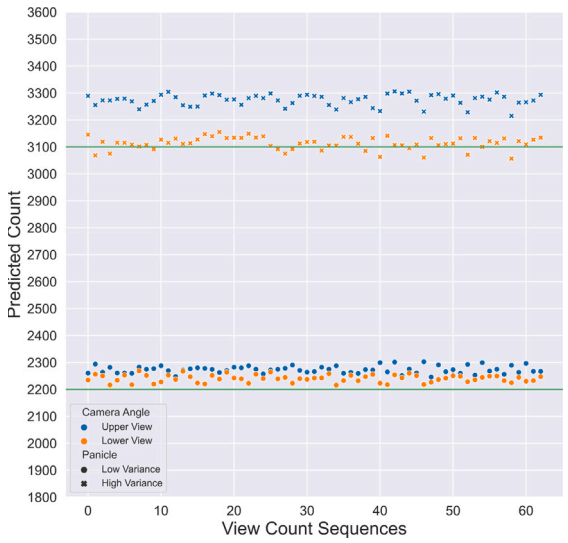


Fig. 18. High-Resolution dataset models: Multiple View Count Sequences (Green line = ground truth reference).

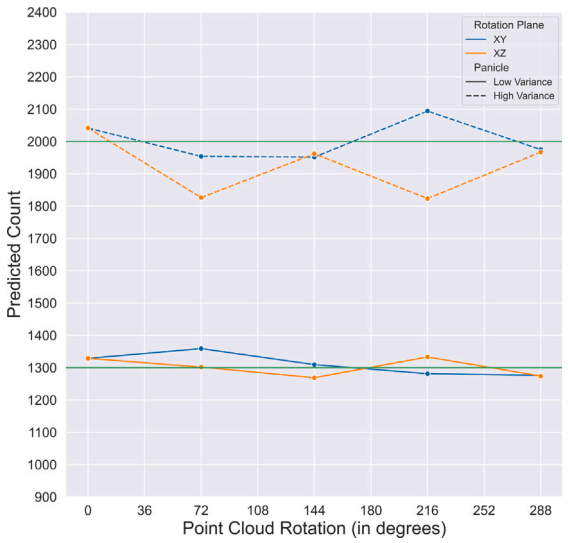


Fig. 19. Low-Resolution dataset models: Test time augmentation — Rotation (Green line = ground truth reference).

panicle is being photographed, there will be a difference in the number of observable grains on the surface, which will impact model prediction for the grain count. Fig. 18 shows the variance in predicted grain count for two different panicles when using different VCSs. The x-axis of the figures is the different VCSs, which are sampled iteratively from the circular view sets. Where each image is treated as the first image in the sequence and the rest of the images are selected based on the length of the view sequence, which are all equally spaced from the first image. The model prediction for one panicle (ground truth count — 2200) is fairly consistent when using different VCSs from different viewing angles. On the other hand, the model prediction for the second panicle (ground truth count — 3100), the grain count prediction is fairly consistent between VCSs from same the viewing angle but varies between different viewing angles.

For the models trained on the low-resolution dataset, similar patterns to the high-resolution dataset, variance in prediction with rotation augmentation can be observed for certain panicles. Fig. 19 shows how the grain count prediction varies when rotating the point cloud model for sample panicles using the same VCS input. For the panicle with a lower grain count of 1300, applying rotation to the point cloud does not significantly affect grain count prediction. However, similar to the high-resolution dataset models, for the panicle with a higher grain count of 2000, rotating the point cloud model results in more variation in the predicted grain count.

Fig. 20 shows how the grain count prediction varies when using different VCSs (for models trained on the low-resolution dataset) for a sample panicle when using the same point cloud model input. Similar to the high-resolution dataset, for certain panicles, there is not much variation in predicted grain count when using different VCSs sampled from different perspective image sets as shown in the figure, but for cases with high variation in predicted grain count, there is a considerable difference, even between VCSs sampled within the same perspective image set and across different perspective image sets. This variation can be attributed to the fact that, for the low-resolution dataset, we only use 3 images for the VCS (unlike 10 images for the high-resolution dataset) which can lead to higher variance in observable grain counts depending on which angle the panicle is imaged from. Therefore, there is a higher variance for predicted grain counts for the low-resolution dataset models.

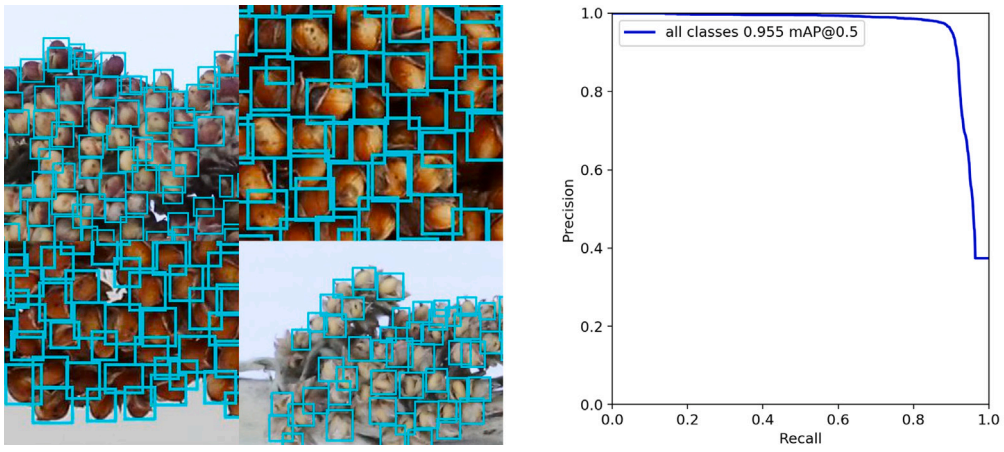
Fig. 20. Low-Resolution dataset models: Multiple View Count Sequences (Green line = ground truth reference).

3.3. Detection model performance

The yolov5 detection model trained for the grain detection from the RGB images was also evaluated on a separate test set comprised of 50 512 × 512 patches sampled randomly from new images which were manually annotated. Fig. 21(a) shows the detection results and Fig. 21(b) shows the mean average precision of the model(evaluated 0.65 intersection over union threshold). The model achieved a consistent mean average precision of 0.955, matching its performance on the validation set.

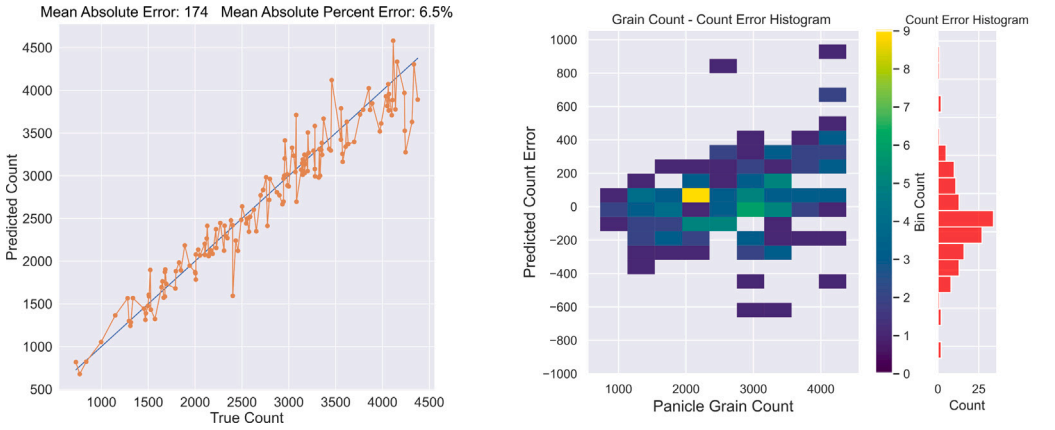
3.4. Cross validation results

Fig. 22 shows the combined results for the cross-validation datasets for the models trained on the high-resolution dataset, and the mean absolute error metric is used to evaluate model performance (as described in Eq. (2)). The best iterations of the models were selected as described in Section 3.1, and test time augmentation was applied to the point cloud models as described in the previous section. The models achieve



(a) Patch Test Set Detections (b) Precision-Recall curve for test set

Fig. 21. Yolov5 Detection model — Test set performance.



(a) Cross Validation Results (b) Prediction Errors - Bland-Altman 2D Histogram

Fig. 22. High-Resolution Dataset Model Results.

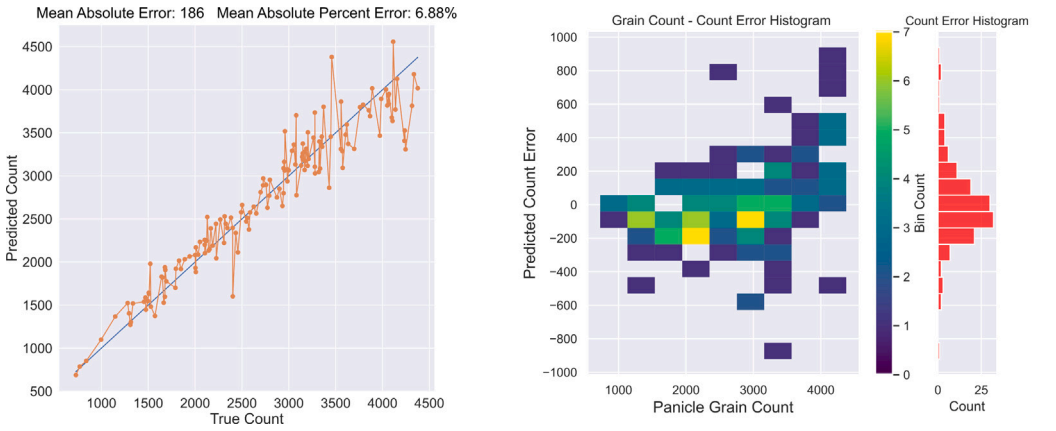
a mean absolute error of 147 grains and a mean absolute percent error of 6.50%, as shown in Fig. 22(a), which depicts the model predictions plotted against the ground-truth grain counts. Fig. 22(b) shows a 2D histogram of the ground truth panicle counts and count errors (and a histogram of the count errors on the right in red), we can observe from the figure that the majority of the errors fall within the range of −200 to 200 grains for the dataset. Furthermore, as the ground truth grain count increases, the variance in prediction errors for the model also increases, and the errors become positively skewed for panicles with higher grain counts.

Fig. 23 shows the combined results for the cross-validation datasets for the models trained on the low-resolution dataset. The models achieve a mean absolute error of 186 grains and a mean absolute percent error of 6.89% for the dataset as shown in Fig. 23(a), which is very comparable to the results achieved by the models trained and evaluated on the high-resolution dataset. Fig. 23(b) shows a 2D histogram of the ground truth panicle counts and count errors, we observe a similar trend as the high-resolution dataset models, the majority of the errors fall within the range of −200 to 200 grains, an increase in the range and variance of errors as the panicle count increases, more positive errors for higher grain count panicles.

3.5. Image count regression models and the ablation experiments on attention module

To examine the benefit of using 3D and RGB images instead of only RGB images, we compare the performance of linear regression models only using the grain counts from RGB image(s) of VCSs to GrainPointNet models. Additionally, we also examine the contribution of the attention-rescaling module, by comparing the performance of GrainPointNet with and without the module. Table 2 shows the mean absolute count errors and r-squared values of image count regression models and GrainPointNet models. For constructing the datasets for the image count regression models, a random VCS of equivalent length was sampled from the list of available VCSs (for the single image regression model we randomly select a single image from the available set of images) for each panicle and the same cross-validation split approach as the GrainPointNet models were used to evaluate model performance. Table 2 clearly demonstrates the limitations of the single image count model with the highest mean absolute count error of 335 grains and the lowest r-squared score of 0.69. This is to be expected as there is considerable variation in observed grain count when observing a sorghum panicle from different views (as seen in 5(c)), it is evident that a single image model is insufficient to capture this variability effectively. However, when regression models are applied to 3-VCS images, a substantial improvement in performance is observed. The mean absolute count error decreases to 264 grains, and the r-squared value increases to 0.829. Further increasing the image count by using a 10-VCS yields only marginal performance gains, with minimal differences observed.

When using a 3-VCS model with low-resolution point cloud models (without the attention rescaling module), leads to a further reduction



(a) Cross Validation Results (b) Prediction Errors - Bland-Altman 2D Histogram

Fig. 23. Low-Resolution Dataset Model Results.

Table 2
Model performance comparison with image count based regression models, and GrainPointNet without attention rescaling module.

Model	Mean absolute error	R-Squared score
Single image count regression model	335.50	0.690
3 VCS image count regression model	263.71	0.829
10 VCS image count regression model	261.77	0.841
3 VCS-Low-Res point cloud (Without attention module)	224.79	0.841
3 VCS-Low-Res point cloud (With attention module)	186.30	0.916
10 VCS-High-Res point cloud (Without attention module)	173.36	0.929
10 VCS-High-Res point cloud (With attention module)	174.29	0.928

in the mean absolute count error to 225 grains. However, the r-squared value remains the same (0.841) as the 10 view-count-sequence regression model. In contrast, when the attention-rescaling module is incorporated into the aforementioned model, there is a significant enhancement in performance. The mean absolute count error is further minimized to 186 grains, while the r-squared value notably increases to 0.916 for the same dataset. When using a 10-VCS with high-resolution point cloud models, we observe another small improvement in performance. Notably, the inclusion of the attention-rescaling module does not contribute to any discernible performance enhancement, when used with the high-resolution dataset. The mean absolute count errors (173 grains vs. 174 grains) and the r-squared values (0.929 vs. 0.928) are virtually identical.

4. Discussion

Upon comparing the cross-validation results for the model trained on the high-resolution dataset and the low-resolution dataset, it is evident that there is no significant decrease in grain count prediction accuracy, the mean absolute error slightly increases from 174 to 186 (mean absolute percent error increases from 6.50% to 6.89%). The high-resolution dataset point clouds provide a more accurate outline of the panicle shape and grain distribution across the surface and use 10 RGB images when compared to the low-resolution dataset set, which uses panicle point clouds derived from synthetically downscaled and low polygon meshes and 3 RGB images. From our experiments, we have demonstrated that a framework that 'independently' combines features from RGB images with low-resolution 3D point cloud models is a viable approach to reliably phenotype individual sorghum panicle grain counts. Furthermore, results from our comparative experiments demonstrate the advantages of using a multi-modal approach for predicting grain count instead of only using RGB images. Additionally, our proposed self-attention-based VCS rescaling module proves effective in processing grain counts from RGB images for low-resolution datasets and helped significantly improve model performance. For the high-resolution dataset, we have not observed any performance improvements with the VCS rescaling module, but it is worth noting that the high-resolution dataset uses grain counts from 10 images, as opposed to only 3 images in the low-resolution dataset. Therefore, reprocessing the counts through the VCS rescaling module may not contribute to an increase in performance, as the grain counts from 10 images likely provide sufficient information to account for the surface grain count and density for the panicle, additionally, capturing 10 images for a single panicle is only viable in a lab or an idealized environment.

As the acquisition of LiDAR data is becoming more ubiquitous and affordable, along with the availability of multimodal sensors (aerial and ground-based RGB, multi-spectral, and LiDAR devices), this warrants researching methodologies that focus on multimodal data fusion to phenotype architectural traits non-invasively on a stand scale. With the help of portable photogrammetry-capable devices equipped with LiDAR sensors, it is possible to obtain a complete mesh or point cloud model of a sorghum panicle. Furthermore, as suggested by Liu et al. (2021), low-altitude UAV flights offer a practical and scalable approach for 3D canopy reconstruction, which could be employed to acquire panicle models in field environments (we have found the panicle models acquired via this method to be comparable to our low-resolution dataset, although the resolution of the surface is lower quality — see supplementary section 5) to validate our methodology. Furthermore, the use of TLS (terrestrial laser scanners) and LiDAR SLAM-based methods can also be considered for the acquisition of a 3D point cloud model of the canopy in the field through proximal measurements. Malambo et al. (2019a) proposed a density-based clustering method for sorghum panicle segmentation, allowing for panicle length and width measurement using RGB point cloud data collected from a TLS. Additionally, Li et al. (2023) introduced a framework utilizing a mobile helmet-based laser scanning system for real-time LiDAR SLAM-based 3D mapping of forest areas. However, when employing LiDAR-based acquisition methods, factors such as LiDAR frequency, duration of scanning, speed, and other inconsistencies of LiDAR SLAM (such as

cumulative slam drift) and point cloud sampling should be considered. Eventually, being able to phenotype grain count per panicle via a scalable high-throughput method will be of enormous benefit to breeders and growers, as grain number is one of the most important traits in sorghum. Typically, there is a strong correlation between grain number and grain yield, Van Oosterom and Hammer (2008) also found grain number was strongly associated with the crop growth rate per unit area, furthermore, Prasad et al. (2021) also found grain number to be a good criterion for selection for yield under high-temperature conditions.

The framework we presented is intended to be a proof of concept, instead of a completed methodology, and can be improved upon in multiple ways. Firstly, the feature extraction modules we used for the point cloud model and the RGB images, can be replaced by newer models. PointNet (Qi et al., 2017) is one of the seminal works in deep learning for natively processing point cloud data, the model can be replaced with more recent works like Dynamic graph CNNs, Wang et al. (2019), and Point Transformer Zhao et al. (2021) which leverage newer deep learning architectures for possibly better feature extraction. Secondly, we use the observable grain count number from the RGB images directly as features, including an additional module that can extract and use shape information and grain density can further improve the performance of the model. Finally, a registration step to align the point cloud model and RGB image can be integrated into our framework for optimizing feature fusion between point clouds and RGB images. In our future work, we aim to extend our current dataset by collecting and imaging more diverse panicles, collecting more LiDAR and point cloud data from aerial and ground-based sensors to extend and validate our methodology infield, and testing different deep learning models to optimize our framework.

5. Conclusion

This study demonstrates a novel method for predicting grain counts in sorghum panicles using a combination of point clouds and RGB images. A framework to process point clouds and a vector (grain counts from a set of images), GrainPointNet, was developed. PointNet (Qi et al., 2017) is used to extract features from the point cloud model, YoloV5 (Jocher et al., 2021) is used for detecting grains from RGB images, and a scaled dot production block is used to process the grain counts from images, finally, the features are combined to predict the panicle grain count. The results show that the proposed method can accurately predict grain counts with a mean absolute percent error of 6.5% for high-resolution point clouds and 6.8% for low-resolution point clouds. This approach offers a non-invasive and efficient way to phenotype individual sorghum panicles for grain count, which can be extended for use in field-based studies.

CRediT authorship contribution statement

Chrisbin James: Conceptualization, Methodology, Data curation, Software, Formal analysis, Investigation, Writing – original draft, Visualization. **Daniel Smith:** Conceptualization, Data curation, Investigation, Writing – review & editing, Project administration. **Weigao He:** Conceptualization, Data curation, Investigation. **Shekhar S. Chandra:** Conceptualization, Methodology, Supervision, Writing – review & editing, Project administration. **Scott C. Chapman:** Conceptualization, Methodology, Supervision, Writing – review & editing, Resources, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Chrisbin James reports financial support was provided by Grains Research and Development Corporation. Daniel Smith reports financial support was provided by Grains Research and Development Corporation.

Data availability

Data will be made available on request.

Acknowledgements

From the University of Queensland, we would like to express our thanks to Mr. Alexander Chamanmáh for helping design and set up the 3D modelling protocol, Mr. Naveed Saeed for supervising the induction to use the threshing and seed-cleaning equipment, and Mr. Lleyton Cave for assisting us with the sampling and transport of the sorghum panicles.

This project was partially funded by the Grains Research and Development Corporation (GRDC) of Australia projects UOQ2003-011RTX 'Innovations in plant testing in Australia' and UOQ2002-08RTX 'High-throughput feature extraction from imagery to map spatial variability'.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.compag.2023.108485.

References

Agisoft, LLC, 2020. Metashape python reference. Release 1, 1–199.

AI, Toolbox, 2022. Scaniverse. URL: https://apps.apple.com/us/app/scaniverse-3d-scanner/id1541433223.

Albawi, Saad, Mohammed, Tareqa Abed, Al-Zawi, Saad, 2017. Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology. ICET, IEEE, pp. 1–6.

Andrade, Fernando H, Vega, Claudia, Uhart, Sergio, Cirilo, Alfredo, Cantarero, Marcelo, Valentinuz, Oscar, 1999. Kernel number determination in maize. Crop Sci. 39 (2), 453–459.

Bochkovskiy, Alexey, Wang, Chien-Yao, Liao, Hong-Yuan Mark, 2020. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

Chang, Anjin, Jung, Jinha, Yeom, Junho, Landivar, Juan, 2021. 3D characterization of sorghum panicles using a 3D point cloud derived from UAV imagery. Remote Sens. 13 (2), 282.

Chung, Sun-Ok, Choi, Moon-Chan, Lee, Kyu-Ho, Kim, Yong-Joo, Hong, Soon-Jung, Li, Minzan, 2016. Sensing technologies for grain crop yield monitoring systems: A review. J. Biosyst. Eng. 41 (4), 408–417.

Community, Blender Online, 2018. Blender - A 3D Modelling and Rendering Package. Blender Foundation, Stichting Blender Foundation, Amsterdam, URL: http://www.blender.org.

David, Etienne, Serouart, Mario, Smith, Daniel, Madec, Simon, Velumani, Kaaviya, Liu, Shouyang, Wang, Xu, Pinto, Francisco, Shafiee, Shahameh, Tahir, Izzat SA, et al., 2021. Global wheat head detection 2021: an improved dataset for benchmarking wheat head detection methods. Plant Phenomics 2021.

Deng, Ruoling, Tao, Ming, Huang, Xunan, Bangura, Kemoh, Jiang, Qian, Jiang, Yu, Qi, Long, 2021. Automated counting grains on the rice panicle based on deep learning method. Sensors 21 (1), 281.

Ester, Martin, Kriegel, Hans-Peter, Sander, Jörg, Xu, Xiaowei, et al., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd. Vol. 96. No. 34. pp. 226–231.

Fourati, Fares, Mseddi, Wided Souidene, Attia, Rabah, 2021. Wheat head detection using deep, semi-supervised and ensemble learning. Can. J. Remote Sens. 47 (2), 198–208.

Freeman, Harry, Schneider, Eric, Kim, Chung Hee, Lee, Moonyoung, Kantor, George, 2022. 3D reconstruction-based seed counting of sorghum panicles for agricultural inspection. arXiv preprint arXiv:2211.07748.

Fu, Jun, Chen, Zhi, Han, Lujia, Ren, Luquan, 2018. Review of grain threshing theory and technology. Int. J. Agricult. Biol. Eng. 11 (3), 12–20.

Ghosal, Sambuddha, Zheng, Bangyou, Chapman, Scott C, Potgieter, Andries B, Jordan, David R, Wang, Xuemin, Singh, Asheesh K, Singh, Arti, Hirafuji, Masayuki, Ninomiya, Seishi, et al., 2019. A weakly supervised deep learning framework for Sorghum head detection and counting. Plant Phenomics 2019.

Girshick, Ross, 2015. Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1440–1448.

Gong, Bo, Ergu, Daji, Cai, Ying, Ma, Bo, 2020. Real-time detection for wheat head applying deep neural network. Sensors 21 (1), 191.

Gong, Liang, Fan, Shengzhe, 2022. A CNN-based method for counting grains within a panicle. Machines 10 (1), 30.

Gong, Liang, Lin, Ke, Wang, Tao, Liu, Chengliang, Yuan, Zheng, Zhang, Dabing, Hong, Jun, 2018. Image-based on-panicle rice [Oryza sativa L.] grain counting with a prior edge wavelet correction model. Agronomy 8 (6), 91.

Hughes, Aoife, Askew, Karen, Scotson, Callum P, Williams, Kevin, Sauze, Colin, Corke, Fiona, Doonan, John H, Nibau, Candida, 2017. Non-destructive, high-content analysis of wheat grain traits using X-ray micro computed tomography. Plant Methods 13 (1), 1–16.

Jiang, Yu, Li, Changying, 2020. Convolutional neural networks for image-based high-throughput plant phenotyping: a review. Plant Phenomics 2020.

Jocher, Glenn, Stoken, Alex, Borovec, Jirka, NanoCode012, Chaurasia, Ayush, TaoXie, Changyu, Liu, V, Abhiram, Laughing, Tkianai, yxNONG, Hogan, Adam, Lorenzomammana, AlexWang1900, Hajek, Jan, Diaconu, Laurentiu, Marc, Kwon, Yonghye, Oleg, Wanghaoyang0106, Defretin, Yann, Lohia, Aditya, Ml5ah, Milanko, Ben, Fineran, Benjamin, Khromov, Daniel, Yiwei, Ding, Doug, Durgesh, Ingham, Francisco, 2021. ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations. http://dx.doi.org/10.5281/zenodo.4679653, Zenodo.

Khaki, Saeed, Pham, Hieu, Han, Ye, Kuhl, Andy, Kent, Wade, Wang, Lizhi, 2020. Convolutional neural networks for image-based corn kernel detection and counting. Sensors 20 (9), 2721.

Khaki, Saeed, Safaei, Nima, Pham, Hieu, Wang, Lizhi, 2022. Wheatnet: A lightweight convolutional neural network for high-throughput image-based wheat head detection and counting. Neurocomputing 489, 78–89.

Kingma, Diederik P., Ba, Jimmy, 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Lee, Youngwan, Park, Jongyoul, 2020. Centermask: Real-time anchor-free instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13906–13915.

Li, Mao, Shao, Mon-Ray, Zeng, Dan, Ju, Tao, Kellogg, Elizabeth A, Topp, Christopher N, 2020. Comprehensive 3D phenotyping reveals continuous morphological variation across genetically diverse Sorghum inflorescences. New Phytol. 226 (6), 1873–1885.

Li, Jianping, Yang, Bisheng, Yang, Yandi, Zhao, Xin, Liao, Youqi, Zhu, Ningning, Dai, Wenxia, Liu, Rundong, Chen, Ruibo, Dong, Zhen, 2023. Real-time automated forest field inventory using a compact low-cost helmet-based laser scanning system. Int. J. Appl. Earth Obs. Geoinf. 118, 103299.

Lin, Zhe, Guo, Wenxuan, 2020. Sorghum panicle detection and counting using unmanned aerial system images and deep learning. Front. Plant Sci. 11, 534853.

Liu, Fusang, Hu, Pengcheng, Zheng, Bangyou, Duan, Tao, Zhu, Binglin, Guo, Yan, 2021. A field-based high-throughput method for acquiring canopy architecture using unmanned aerial vehicle images. Agricult. Forest Meteorol. 296, 108231.

Lu, Hao, Cao, Zhiguo, Xiao, Yang, Zhuang, Bohan, Shen, Chunhua, 2017. TasselNet: Counting maize tassels in the wild via local counts regression network. Plant Methods 13 (1), 1–17.

Malambo, L., Popescu, S.C., Horne, D.W., Pugh, N.A., Rooney, W.L., 2019a. Automated detection and measurement of individual Sorghum panicles using density-based clustering of terrestrial lidar data. ISPRS J. Photogramm. Remote Sens. 149, 1–13.

Malambo, Lonesome, Popescu, Sorin, Ku, Nian-Wei, Rooney, William, Zhou, Tan, Moore, Samuel, 2019b. A deep learning semantic segmentation-based approach for field-level Sorghum panicle counting. Remote Sens. 11 (24), 2939.

Panasiewicz, Marian, Sobczak, Paweł, Mazur, Jacek, Zawiślak, Kazimierz, Andrejko, Dariusz, 2012. The technique and analysis of the process of separation and cleaning grain materials. J. Food Eng. 109 (3), 603–608.

Paszke, Adam, Gross, Sam, Massa, Francisco, Lerer, Adam, Bradbury, James, Chanan, Gregory, Killeen, Trevor, Lin, Zeming, Gimelshein, Natalia, Antiga, Luca, Desmaison, Alban, Kopf, Andreas, Yang, Edward, DeVito, Zachary, Raison, Martin, Tejani, Alykhan, Chilamkurthy, Sasank, Steiner, Benoit, Fang, Lu, Bai, Junjie, Chintala, Soumith, 2019. PyTorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems. Vol. 32. Curran Associates, Inc, pp. 8024–8035, URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Prasad, VB Rajendra, Govindaraj, Mahalingam, Djanaguiraman, Maduraimuthu, Djalovic, Ivica, Shailani, Anjali, Rawat, Nishtha, Singla-Pareek, Sneh Lata, Pareek, Ashwani, Prasad, PV Vara, 2021. Drought and high temperature stress in Sorghum: Physiological, genetic, and molecular insights and breeding approaches. Int. J. Mol. Sci. 22 (18), 9826.

Qi, Charles R., Su, Hao, Mo, Kaichun, Guibas, Leonidas J., 2017. Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660.

Redmon, Joseph, Farhadi, Ali, 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

Ren, Shaoqing, He, Kaiming, Girshick, Ross, Sun, Jian, 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems. Vol. 28.

Sandler, Mark, Howard, Andrew, Zhu, Menglong, Zhmoginov, Andrey, Chen, Liang-Chieh, 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4510–4520.

Schmidt, Jessica, Claussen, Joelle, Wörlein, Norbert, Eggert, Anja, Fleury, Delphine, Garnett, Trevor, Gerth, Stefan, 2020. Drought and heat stress tolerance screening in wheat using computed tomography. Plant Methods 16 (1), 1–12.

Tan, Mingxing, Pang, Ruoming, Le, Quoc V., 2020. Efficientdet: Scalable and efficient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10781–10790.

Uhl, J.B., Lamp, B.J., 1966. Pneumatic separation of grain and straw mixtures. Trans. ASAE 9 (2), 244–246.

Van Oosterom, E.J., Hammer, G.L., 2008. Determination of grain number in Sorghum. Field Crops Res. 108 (3), 259–268.

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N, Kaiser, Łukasz, Polosukhin, Illia, 2017. Attention is all you need. In: Advances in Neural Information Processing Systems. Vol. 30.

Velesaca, Henry O, Mira, Raúl, Suárez, Patricia L, Larrea, Christian X, Sappa, Angel D, 2020. Deep learning based corn kernel classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 66–67.

Wang, Yue, Sun, Yongbin, Liu, Ziwei, Sarma, Sanjay E, Bronstein, Michael M, Solomon, Justin M, 2019. Dynamic graph CNN for learning on point clouds. ACM Trans. Graph. (tog) 38 (5), 1–12.

Wei, Wu, Yang, Tian-le, Rui, Li, Chen, Chen, Tao, Liu, Kai, Zhou, Sun, Cheng-ming, Li, Chun-yan, Zhu, Xin-kai, Guo, Wen-shan, 2020. Detection and enumeration of wheat grains based on a deep learning method under various scenarios and scales. J. Integrat. Agricult. 19 (8), 1998–2008.

Xiong, Haipeng, Cao, Zhiguo, Lu, Hao, Madec, Simon, Liu, Liang, Shen, Chunhua, 2019. TasselNetv2: in-field counting of wheat spikes with context-augmented local regression networks. Plant Methods 15 (1), 1–14.

Zanke, Christine D, Ling, Jie, Plieske, Jörg, Kollers, Sonja, Ebmeyer, Erhard, Korzun, Viktor, Argillier, Odile, Stiewe, Gunther, Hinze, Maike, Neumann, Felix, et al., 2015. Analysis of main effect QTL for thousand grain weight in European winter wheat (Triticum aestivum L.) by genome-wide association mapping. Front. Plant Sci. 6, 644.

Zhao, Hengshuang, Jiang, Li, Jia, Jiaya, Torr, Philip HS, Koltun, Vladlen, 2021. Point transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16259–16268.

Zhao, Ping, Li, Yongkui, 2009. Grain counting method based on image processing. In: 2009 International Conference on Information Engineering and Computer Science. IEEE, pp. 1–3.

Zou, Hongwei, Lu, Hao, Li, Yanan, Liu, Liang, Cao, Zhiguo, 2020. Maize tassels detection: a benchmark of the state of the art. Plant Methods 16 (1), 1–15.