2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Seed Kernel Counting using Domain Randomization and Object Tracking Neural Networks

Venkat Margapuri
Department of Computer Science
Villanova University
Villanova, PA
vmargapu@villanova.edu

Prapti Thapaliya
Department of Computer Science
Villanova University
Villanova, PA
pthapali@villanova.edu

Mitchell Neilsen
Department of Computer Science
Kansas State University
Manhattan, KS
neilsen@ksu.edu

***Abstract*—High-throughput phenotyping (HTP) of seeds is the comprehensive assessment of complex seed traits and the measurement of parameters that form more complex traits [1]. The key aspect of seed phenotyping is cereal yield estimation. While mechanized seed kernel counters are available in the market currently, they are often priced high and sometimes outside the range of small scale seed production firms' affordability. The development of object tracking neural network models such as You Only Look Once (YOLO) enables computer scientists to design algorithms that can estimate cereal yield inexpensively. The key bottleneck with neural network models is that they require a plethora of labelled training data before they can be put to task. We demonstrate that the use of synthetic imagery serves as a feasible substitute to train neural networks for object tracking. Furthermore, we propose a seed kernel counter that uses a low-cost mechanical hopper, trained YOLOv8 neural network model, and object tracking algorithms on StrongSORT and ByteTrack to estimate cereal yield from videos. The experiment yields a seed kernel count with an accuracy of 95.2% and 93.2% for Soy and Wheat respectively using the StrongSORT algorithm, and an accuray of 96.8% and 92.4% for Soy and Wheat respectively using the ByteTrack algorithm.**

***Index Terms*—YOLOv8, Artificial Intelligence, Domain Randomization, Object Tracking, Seed Counter**

I. Introduction

The advent of technology in agriculture commenced over a century ago, and several studies have been conducted since the 1990s to improve production efficiency [2]. High-throughput Phenotyping (HTP) of seeds is the comprehensive assessment of complex seed traits and the measurement of parameters that form more complex traits [1]. Currently, seed production firms have to use expensive mechanized seed counting machinery to pack seed kernels by count. This paper demonstrates leveraging videos of seed kernels rolling down a platform to estimate seed kernel count using low-cost hardware components (described in section II) and the object tracking neural network model, You Only Look Once (YOLO). Supervised neural network models require a plethora of labelled information to train for tasks. However, labeled training data is not always readily available for entities such as seed kernels. We demonstrate that the use of synthetic image datasets, generated following the principles of Domain Randomization [3]–[5], is a feasible alternative to train neural network models.

II. Related Work

Neilsen et al. [6] proposed an image processing algorithm to conduct seed kernel counting from videos. It is based on tracking each of the seed kernels as they flow down a backlit platform. However, the image processing algorithm is highly sensitive to the video's frame rate. GridFree [7] is a Python package for image analysis of interactive grain counting and measurement. It uses K-Means to and principal component analysis (PCA) on both raw image channels and their color indices. It exhibits great performance on multiple crop types. Parico et al. [8] performed real-time pear fruit detection and counting using YOLOv4 models and Deep SORT algorithm. The region-of-interest (ROI) line technique was used to estimate the number of pear fruits detected by the neural network model. Wu et. al. [9] performed detection of Camellia oleifera fruit in complex scenes by using YOLOv7 and data augmentation. The experiment yielded a Mean Average Precision, Precision, Recall, F1 Score, and average detection time of 96.03%, 94.76%, 95.54%, 95.15%, and 0.025 seconds per image respectively.

III. Hardware Components

We propose a low-cost setup for the capture of seed kernel videos for algorithmic analysis using YOLOv8. Fig. 1 shows the seed kernel image capture setup designed for the experiment. The mechanical hopper delivers seeds at a constant rate. The mobile phone is placed on a 3-D printed stand to ensure that the camera is always held orthogonal to the surface to eliminate any skew that may result during the capture of the video. The 3-D printed platform at the bottom channels the seed kernels ensuring that the seed kernels remain in the field of view of the camera as they roll down the lightbox. The mobile phone used for image capture is a Google Pixel 2 XL mobile phone whose default capture frame rate is 60 fps. Fig. 2 shows a frame of the wheat seed kernel video captured using the proposed setup in Fig. 1.

IV. Domain Randomization and Image Datasets

Domain Randomization (DR) trains neural network models on a small sample of images containing simulated objects that

Fig. 1. Mechanical hopper delivering seed kernels

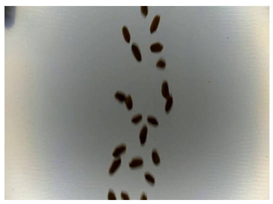

Fig. 2. Wheat seed kernels flowing down the light box

translate closely to real-world objects. Fig. IV shows soy seeds being captured by the proposed image capture setup. Images of 25 seed kernels of soy and wheat are captured using the setup shown in Fig. 3. Using the synthetic image generator developed as part of a previous work [3] synthetic images containing seed kernels of soy and wheat are developed. The synthetic images allow for about 25% overlap at the maximum to account for clustered seed kernels as the frames of the video are processed. Datasets are created for the seed types of soy and wheat, wherein each dataset consists of 200 images of size 320x320x3 with each image containing between 25 and 50 seed kernels overlaid on a light background, as shown in Fig. 4. The generator outputs annotation files that contain location coordinates pertinent to each seed kernel in the image in the TXT format for YOLOv8 to consume and process during training. An additional 35 synthetic images containing 30 seed kernels of each seed kernel type are generated for testing on YOLOv8.

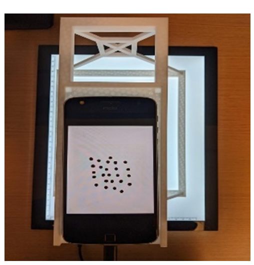

Fig. 3. Image capture of soy seed kernels

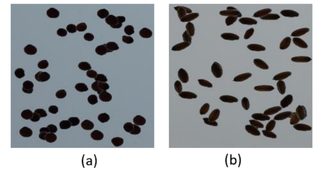

Fig. 4. Image capture of soy seed kernels

V. YOLOv8 and Object Tracking Algorithms

The YOLO model is a single-shot detector [10]–[12] that uses a fully convolutional neural network as the backbone to process the input image. The YOLOv8 [13] model was released in January 2023 by Ultralytics. It comprises a convolutional neural network divided into two parts: backbone and head. The backbone, CSPDarknet53 [14], consists of 53 convolutional layers. The head consists of multiple convolutional layers followed by a series of fully connected layers. Object tracking requires that the object be detected in every frame across the video. Several object tracking algorithms [17]–[19] have been proposed over the years. This paper considers two object tracking algorithms for experimentation, namely, StrongSORT and ByteTrack.

A. StrongSORT

The StrongSORT [17] algorithm is an improvement over the DeepSORT [18] algorithm. It is a two-branch framework consisting of an Appearance branch and a Motion branch. The Appearance branch identifies the features of each of the objects detected in a given frame. BoT [20] is leveraged as the feature extractor by the StrongSORT algorithm. The appearance state for the i^{th} tracklet within frame t, e_i^t, is updated as the exponential moving average (EMA) given by $e_i^t = \alpha e_i^{t-1}$ + (1-α)f_i^t where f_i^t is the appearance embedding of the current matched detection and α = 0.9, is a variable momentum term. The Motion branch leverages Kalman Filter [23] to predict the position of the object in the frame based on a constant velocity model. The StrongSORT algorithm uses the NSA Kalman Filter algorithm borrowed from the GIAO tracker [24].

B. ByteTrack

ByteTrack [21] algorithm leverages bounding boxes at all confidence levels agnostic of any threshold and attempts to identify all objects in a frame. It uses Tracklets queue to store all the objects (and bounding boxes) that have been detected by the object detector (YOLOv8). The bounding boxes are separated into high score (D^{high}) and low score (D^{low}) based on threshold. Each of the objects in the Tracklets queue is tracked across each frame of the video using Kalman Filter [23]. Firstly, the position of each of the objects in the tracklets queue is predicted in the subsequent frame. The predictions are matched with the actual detections made by the object detector using Motion Similarity score which is computed with Intersection over Union (IoU) between the predicted and actual bounding boxes. Initially, tracklet matching is done between the predicted and high score (D^{high}) bounding boxes. The tracklets that do not match with any of the high score bounding boxes are matched with low score (D^{low}) bounding boxes. Any tracklet that is not matched is preserved for a predefined number of frames to test for rebirth in case of occlusion. Finally, the tracklet is removed from the queue if a match is never found.

VI. Experiment

The development of the Seed Counter using YOLOv8 and object tracking algorithms involves three steps: Seed detection, Seed Tracking, and Seed Counting.

A. Seed Detection

The YOLOv8 model is trained on the image dataset (described in section IV) using transfer learning. 80% of the image dataset is used for training and 20% is used for validation. The test dataset consists of 100 images, 50 of soy and 50 of wheat, each containing 20-30 seed kernels. Model weights from the YOLOv8 model pre-trained on the COCO image dataset are leveraged as provided by Uralytics. The hyperparameters used to train the model are as shown in Table I. The results of seed kernel detection on the test dataset are evaluated using the metrics of Precision, Recall, and Average Precision (PR). The metrics are briefly described below and the obtained results are presented in Table II.

TABLE I
Hyperparameters for YOLOv8

Hyperparameter	Value
Learning Rate	0.001
Batch Size	32
Input Image Size	320x320x3 px
Bounding Box Confidence Threshold	0.4
Non-Maximum Suppresion Threshold	0.4
Intersection-over-Union Threshold	0.5
Activation Function	LeakyReLU
Filters in Each Layer	64
Dropout	yes
Pretrained Model Weights	yolov8x

Precision: Precision is given by *true positives/ (true positives + false positives).*
Recall: Recall is given by *true positives/(true positives + false negatives).*
Average Precision: The Average Precision is computed for an 50% overlap between ground truth and predicted bounding boxes for the purposes of this paper, given by AP_{50}.
Note: Average Precision is only reported for the validation data set but not test data set because the images in the test set do not have ground truth bounding boxes plotted around them.
From the results in table II, high recall scores of 91% and 90% on soy and wheat respectively for the test set indicates that the model albeit being trained on synthetic images detects real seed kernels well. The precision scores of 93% and 92% indicate that the model classifies the seed kernels correctly on most instances. The reason for high precision might be due to the clear morphometric distinction between soy and wheat.

TABLE II
Evaluation Results

Seed Kernel	Validation Set			Test Set	
	Precision	Recall	AP_{50}	Precision	Recall
Soy	98%	92%	92%	93%	91%
Wheat	97%	93%	89%	92%	90%

TABLE III
Unique ID Count on Video of Wheat Seed Kernels by StrongSORT

Frame Rate	Seed Kernel Count	Unique IDs
30	250	306
60	250	381
120	250	533

B. Seed Tracking

The Seed Tracking phase applies StrongSORT and ByteTrack algorithms on videos with 250 seed kernels of each seed type captured at three different frame rates, 30, 60, and 120. Both algorithms are applied using the detection weights obtained in the Seed Detection phase as input. The algorithms apply a unique ID to each seed kernel detected in the video and track them throughout the video. The seed kernels in the video are clustered in parts, occluded, and prone to sudden deviations in trajectory, as shown in Fig. 2. These issues lead to the risk of object tracking algorithms assigning different unique IDs to the same seed kernel in different frames of the video, leading to a discrepancy. The number of unique IDs generated by the algorithms on each of the videos captured for the wheat seeds are shown in Tables III and IV respectively. The results show that either of the algorithms consistently overcount the number of seed kernels in the video.

C. Seed Counting

Seed Counting uses a region of interest (RoI) established at a common location across each frame in the video. Any seed kernel that crosses the RoI is accounted to be one seed kernel. The total number of seed kernels is given by the total number of tracks that cross the RoI. Tables V and VI show the results obtained by applying the StrongSORT and ByteTrack algorithms on each of the videos. The performance of the algorithms improves as the frame rate increases. The key issue faced by the algorithms is sudden changes in trajectory due to the seed kernels touching one another and deviating from their present trajectory. This phenomenon affects the object tracking algorithms's ability to predict the location of the seed kernel accurately in subsequent frames.

TABLE IV
Unique ID Count on Video of Wheat Seed Kernels by ByteTrack

Frame Rate	Seed Kernel Count	Unique IDs
30	250	322
60	250	406
120	250	592

TABLE V
RESULTS OF SOY AND WHEAT KERNEL COUNT USING STRONGSORT

Seed Type	Frame Rate	Actual Count	YOLOv8 Count	Accuracy
Soy	30	250	238	95.2
Soy	60	250	214	85.6
Soy	120	250	188	75.2
Wheat	30	250	233	93.2
Wheat	60	250	207	82.8
Wheat	120	250	166	66.4

TABLE VI
RESULTS OF SOY AND WHEAT KERNEL COUNT USING BYTETRACK

Seed Type	Frame Rate	Actual Count	YOLOv8 Count	Accuracy
Soy	30	250	242	96.8
Soy	60	250	211	84.4
Soy	120	250	194	77.6
Wheat	30	250	231	92.4
Wheat	60	250	209	83.6
Wheat	120	250	171	68.4

From the results in tables V and VI, the seed count is most accurate on videos captured at a frame rate of 30 and least on videos captured at a frame rate of 120 for either seed type, and object tracking algorithm demonstrating that a lower frame that captures a higher level of detail positively influences the performance of object tracking algorithms. However, both algorithms undercount the number of seed kernels agnostic of frame rate due to the clustering of seed kernels.

VII. PITFALLS, FUTURE WORK, AND CONCLUSION

The key pitfall of the experiment is that the videos used for the experiment consist of seed kernels that are clustered. As a result, the object tracking algorithms failed to track each seed kernel accurately. In further experiments, the video capture mechanism will be altered to ensure that the videos do not contain clustered (or occluded) seed kernels. Overall, the experiment demonstrates the feasibility of synthetic images to train object tracking neural network models, and their application in seed kernel counting aimed at the seed production industry. As the results are encouraging, future work will involve the development of a mobile application (Android/iOS) and a robust video capture mechanism.

REFERENCES

[1] Li, Lei and Zhang, Qin and Huang, Danfeng. A review of imaging techniques for plant phenotyping, Sensors, vol. 14, pp. 20078–20111, 2014

[2] Santos, Luís and Santos, Filipe N and Oliveira, Paulo Moura and Shinde, Pranjali. Deep learning applications in agriculture: A short review, Robot 2019: Fourth Iberian Robotics Conference: Advances in Robotics, Volume 1, pp. 139–151, 2020, Springer

[3] Margapuri, Venkat and Neilsen, Mitchell. Seed phenotyping on neural networks using domain randomization and transfer learning, 2021 ASABE Annual International Virtual Meeting, pp. 1, 2021, American Society of Agricultural and Biological Engineers

[4] Tobin, Josh and Fong, Rachel and Ray, Alex and Schneider, Jonas and Zaremba, Wojciech and Abbeel, Pieter. Domain randomization for transferring deep neural networks from simulation to the real world, 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp. 23–30, 2017, IEEE

[5] Zakharov, Sergey and Kehl, Wadim and Ilic, Slobodan. Deceptionnet: Network-driven domain randomization, Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 532–541, 2019

[6] Neilsen, Mitchell L and Courtney, Chaney and Amaravadi, Siddharth and Xiong, Zhiqiang and Poland, Jesse and Rife, Trevor. A dynamic, real-time algorithm for seed counting, Proc. Of the 26th International Conference on Software Engineering and Data Engineering, 2017

[7] Hu, Yang and Zhang, Zhiwu, GridFree: a Python package of image-analysis for interactive grain counting and measuring. Plant physiology, volume=186, pp. 2239–2252, 2021, Oxford University Press

[8] Real time pear fruit detection and counting using YOLOv4 models and deep SORT, Parico, Addie Ira Borja and Ahamed, Tofael. Sensors, volume=21, pp. 4803, 2021

[9] Wu, Delin and Jiang, Shan and Zhao, Enlong and Liu, Yilin and Zhu, Hongchun and Wang, Weiwei and Wang, Rongyan, Detection of Camellia oleifera fruit in complex scenes by using YOLOv7 and data augmentation, Applied Sciences, volume=12, pp. 11318, 2022, MDPI

[10] Liu, Wei and Anguelov, Dragomir and Erhan, Dumitru and Szegedy, Christian and Reed, Scott and Fu, Cheng-Yang and Berg, Alexander C. Ssd: Single shot multibox detector, Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, pp. 21–37, 2016

[11] Fu, Cheng-Yang and Liu, Wei and Ranga, Ananth and Tyagi, Ambrish and Berg, Alexander C. Dssd: Deconvolutional single shot detector, arXiv preprint arXiv:1701.06659, 2017

[12] Magalhães, Sandro Augusto and Castro, Luís and Moreira, Germano and Dos Santos, Filipe Neves and Cunha, Mário and Dias, Jorge and Moreira, António Paulo. Evaluating the single-shot multibox detector and YOLO deep learning models for the detection of tomatoes in a greenhouse, Sensors, volume 21, pp. 3569, 2021

[13] Terven, Juan and Cordova-Esparza, Diana. A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond, arXiv preprint arXiv:2304.00501, 2023

[14] Mahasin, Marsa and Dewi, Irma Amelia. Comparison of CSPDarkNet53, CSPResNeXt-50, and EfficientNet-B0 Backbones on YOLO V4 as Object Detector, International Journal of Engineering, Science and Information Technology, volume 2, pp. 64–72, 2022

[15] Ramachandran, Prajit and Parmar, Niki and Vaswani, Ashish and Bello, Irwan and Levskaya, Anselm and Shlens, Jon. Stand-alone self-attention in vision models, Advances in neural information processing systems, volume 32, 2019

[16] Zhao, Hengshuang and Jia, Jiaya and Koltun, Vladlen. Exploring self-attention for image recognition, Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 10076–10085 2020

[17] Du, Yunhao and Zhao, Zhicheng and Song, Yang and Zhao, Yanyun and Su, Fei and Gong, Tao and Meng, Hongying. Strongsort: Make deepsort great again, IEEE Transactions on Multimedia, 2023, IEEE

[18] Dang, Tuan Linh and Nguyen, Gia Tuyen and Cao, Thang. Object tracking using improved deep SORT YOLOv3 architecture, ICIC Express Letters, volume 14, pp. 961–969, 2020

[19] Yolox: Exceeding yolo series in 2021, Ge, Zheng and Liu, Songtao and Wang, Feng and Li, Zeming and Sun, Jian, arXiv preprint arXiv:2107.08430, 2021

[20] A strong baseline and batch normalization neck for deep person re-identification. Luo, Hao and Jiang, Wei and Gu, Youzhi and Liu, Fuxu and Liao, Xingyu and Lai, Shenqi and Gu, Jianyang, IEEE Transactions on Multimedia, volume 22, pp. 2597–2609, 2019, IEEE

[21] Zhang, Yifu and Sun, Peize and Jiang, Yi and Yu, Dongdong and Weng, Fucheng and Yuan, Zehuan and Luo, Ping and Liu, Wenyu and Wang, Xinggang. Bytetrack: Multi-object tracking by associating every detection box, European Conference on Computer Vision, pp. 1–21, 2022, Springer

[22] Girshick, Ross. Fast r-cnn, Proceedings of the IEEE international conference on computer vision, pp. 1440–1448, 2015

[23] Li, Qiang and Li, Ranyang and Ji, Kaifan and Dai, Wei. Kalman filter and its application, 2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS), pp. 74–77, 2015, IEEE

[24] Du, Yunhao and Wan, Junfeng and Zhao, Yanyun and Zhang, Binyu and Tong, Zhihang and Dong, Junhao. Giaotracker: A comprehensive framework for mcmot with global information and optimizing strategies in visdrone 2021, Proceedings of the IEEE/CVF International conference on computer vision, pp. 2809–2819, 2021