Comparative analysis of lightweight Deep Learning architectures for Soybean pod count from proximal sensing data for mobile and embedded vision applications

Jithin Mathew (jithin.mathew@ndsu.edu)[1], Zhao Zhang (zhaozhangcau@cau.edu.cn)[2*], Paulo Flores (paulo.flores@ndsu.edu)[1], John Stenger (john.stenger@ndsu.edu)[1], Carrie Miranda (carrie.miranda@ndsu.edu)[3], Anup Das (anup.das@ndsu.edu)[1]

[1] Department of Agricultural and Biosystems Engineering, North Dakota State University, Fargo, ND, USA

[2] Key Lab of Modern Precision Agriculture System Integration Research, Ministry of Education of China, China Agricultural University, Beijing. 100083, China

[3] Department of Plant Sciences, North Dakota State University, Fargo, ND, USA

A paper from the Proceedings of the
15th International Conference on Precision Agriculture
June 26-29, 2022
Minneapolis, Minnesota, United States

Abstract.

Crop yield prediction is an important aspect of farming and food-production. On-farm soybean (Glycine max L.(Merr.)) yield estimation enables farmers or crop producers to make some key management decisions regarding storage, transport, and sales of the product. Yield prediction prior to harvest can be of great values for soybean breeders since it can help to make decisions on what genotypes should be advanced or discarded in their breeding programs. Existing soybean yield estimation approaches, such as satellite imagery-based yield estimation, has various limitations such as a lack of real-time and on-farm decision making capabilities. Moreover, due to high-end computation and expertise requirements, present technology that provides on-farm decision making is not practicable for end-users. The objective of this study was to apply transfer-learning to train current Machine Learning (ML) frameworks, compare them, and suggest the optimum architecture for small-end devices like smart-phones or embedded systems for real-time soybean pod counting, which then could be used to estimate yields. This study is aimed to aid soybean breeders to estimate yield from infield still images or real-time video data collected using smart phone sensors. To enhance the dataset and generalize the trained model and improve predictions, various data augmentation techniques were applied to the image dataset. Toward this goal, we train a variety of streamlined light-weight Deep Learning (DL) based object detection frameworks to compare and find the best architecture by testing and evaluating the model using COCO-evaluation metrics. We use transfer-learning to train existing state-of-the-art

The authors are solely responsible for the content of this paper, which is not a refereed publication. Citation of this work should state that it is from the Proceedings of the 15th International Conference on Precision Agriculture. EXAMPLE: Last Name, A. B. & Coauthor, C. D. (2018). Title of paper. In Proceedings of the 15th International Conference on Precision Agriculture (unpaginated, online). Monticello, IL: International Society of Precision Agriculture.

DL models (YOLOv5-small, YOLOv3-tiny, EfficientDet-Lite, SSDLite Mobilenet v2) and compare their performance to identify the architecture taking in consideration the tradeoff between speed and accuracy for mobile and embedded systems. Our research shows that YOLOv5-small exhibit the best performance in terms of accuracy with an mAP@0.5 of 87.9 and mAP@0.5:0.95 of 43.7%, outperforming other light-weight architectures by a large margin. This study makes a significant contribution towards choosing the best light-weight DL architecture for soybean pod count strategy for in-field yield estimation that will aid soybean breeder and farmer alike.

Keywords.
Plant Phenotyping, Deep learning, embedded devices, object-detection, high-throughput pod count.

INTRODUCTION

Yield is the quantitative measurement of crop [1]. In precision agriculture, crop yield estimation is an essential component that can provide support for agricultural decision-making and management [35]. Soybean yield estimation can be performed using three main methods: counting the pods per plants (PN), the number of seeds per pod (SPP), and finally by measuring the seed size [2]. Predicting yield before harvest can help estimating the net return of the crops, particularly if additional investments are to be made, and more importantly, considering the environmental factors, and the yield output of the genotype, farmer can plan for the upcoming seasons. For soybean breeders it is important to predict the performance of new varieties to further maximize its yield capacity [3], which is done via estimating the yield by measuring phenotypic features like PN, SPP, or seed size data to develop high-yielding cultivars [2].

Soybean is one of the most economically important crops in the world with an average production of 3378.5 Kg /ha in 2020 alone [4]. Therefore, it is important to understand the underlying genetic mechanism behind the agronomic traits of a crop that will help breeders to improve its yield potential for the upcoming seasons [6]. During crop development, agronomic and seed quality attributes are improved by crossing phenotypically superior soybean cultivars and selecting improved offspring in each cycle [17]. Through the combination of selective breeding and genetic engineering, soybean production per hectare increased by twofold from 1961 to 2020. Similarly, during the last 60 years, the total hectares of grown soybeans have expanded by 32.8% [4].

Crop yield is affected by several variables, including weather availability, soil types, seed variety, fertilizers, etc. In more general terms, the yield production will depend on the crop genotype (G) and its interaction with the external environment (E), which in turn produces the variation in the yield [7] [3] [8]. Primarily, soybean yield predictions are generated by statistical analysis or machine learning application on spectral data, remote sensing vegetation indices, soil properties, weather data, among other layers of information [1][9]. Recent trends show an increase in the application of digital image processing and Machine Learning for crop yield estimation [8]. In addition, a wide variety of imagery techniques have been employed to acquire these digital images, such as remote sensing (satellite [34], unmanned aerial vehicles (UAVs)) [10], proximal sensing (robots [13], smartphone cameras [2]). While majority of the research has focused on RGB images (red, green, and blue), some researchers have also explored the use of multispectral sensors to estimate soybean yield [11].

Previous research has focused on yield estimation based on satellite imaging, which provides estimates for a broader region (county, state, country), but this does not satisfy an in-filed real-time yield estimate for individual farmers and breeders, and there have been very few studies carried out in this space [9]. Because pod counting has a higher correlation with yield, it is now being used to estimate soybean yield more often than ever before. It is very common that a soybean breeder will have field trials with thousands of small plots to test a variety of genotypes in their breeding program. As one can image, manually counting the pods from individual plots on that kind scale would be expensive, labor intensive, and the results would prone to inaccuracies due to individuals' fatigue [12] [13]. Therefore, it's more necessary than ever to investigate a DL based, scalable, high-throughput pod counting technique that can be applied in real-time on low-end devices for in-field applications of high-throughput phenotypic research.

The use of computer vision coupled with machine learning or deep learning have the potential to play a big role in the development of solutions for in-field real time soybean pods assessment to support the development pre-harvest soybean yield predictions [1][12]. The recent advancement in ML and DL that are associated with computer vision (CV) has enabled vision (image) based yield estimation more practical and efficient. In contrast to soybean pod counting, where the pod and background are in a noisy environment, current CV yield estimation applications are widely seen in occlusion and clutter free background environments [13]. To overcome background and noise related hurdles, one can use multiple view angles and image fusion techniques to improve the accuracy and performance of the DL architecture deployed [13]. Most of the CV studies have demonstrated that larger architectures lead to superior performance, but they are resource intensive, making their deployment on mobile devices with limited computing capabilities impracticable. The task of finding a light-weight architecture that offers a tradeoff between speed and accuracy for edge devices is ever more challenging due to limited research carried out in this subject matter [14]. Therefore, this study was carried out to investigate the potential of lightweight mobile architectures for DL based object detection for soybean pod counting to support the development of better soybean yield predictions. The objective of this study was to compare the performance of existing light-weight architectures for pod count and to determine the best architecture that provides good performance despite the challenges faced by DL architectures during the feature extraction step, such as clutter and background noises and smaller size of the target object.

MATERIALS AND METHODS

Data acquisition and preprocessing

Data was collected from one of the NDSU soybean breeding program's field trials, which was in Casselton, ND, USA. A Google Pixel 4a smartphone was used for data collection since the overall goal of the project is to deploy the best performing trained neural network on a smartphone towards the end goal. A total of 7002 images were acquired, which then were used to generate additional augmented images (3000 images), resulting in a dataset with 10,002 images. The images were split into training and test sets using a 90:10 ratio, resulting in 8968 images being used for training and 997 images being used for testing. During augmentation, photometric and geometric transformation were applied. For photometric augmentation, we applied methods like brightness, contrast, gamma, smoothness, and gaussian noise arbitrarily. For geometric transformations, the images were subjected to random cropping, rotation, flipping, and zooming. Each image was then labelled using LabelImg software [36] under YOLO labelling format and Pascal VOC format [33] to meet each architecture requirements. After first manual labelling process, we use pseudo-labelling technique [32] to expand the dataset and label all 10,002 images.

Machine learning framework for soybean pod count

Since overall goal of the project is to develop a solution for real-time yield estimation in the field, training and implementing an object detection algorithm in a device such as a laptop or a desktop computer would create some difficulties for field implementation. Hosting the model on a cloud computing platform, on the other hand, can solve this problem, but one would still need a reliable and fast internet connection for data transfer between the server and the user, which is typically limited in rural locations throughout the world. The best method for entirely overcoming those issues is to host the DL model on a mobile or embedded device. Toward that end, the focus of this study was to train smaller versions of current state-of-the-art (SOTA) object detection frameworks, compare their performance in terms of accuracy, and deploy the best performing model on a smartphone with edge Tensor Processing Unit (TPU).

YOLOv3-tiny

The YOLO [20] working concept unifies the architecture and, as a result, performs feature extraction and bounding box/class prediction in a single stretch, while treating the task as a regression problem [37]. While YOLO isn't particularly good at accurately localizing target objects, it generates fewer mistakes in terms of background false positives than Fast R-CNN, a predecessor to the SOTA object detection framework Faster- RCNN [23]. K-means clustering is used to estimate the bounding box beforehand, whereas to extract features from three scales, a technique like feature pyramid networks (FPN) [22] is utilized. For feature extraction, Darknet-53, a larger backbone with 53 convolutional layers (an upgrade from Darknet-19 utilized in YOLOv2 [24]) is employed. Data augmentation also plays a critical role in the performance of YOLOv3 during training. The network produces a 3-d tensor encoding with bounding box, objectness, and class prediction as outputs. YOLOv3 tiny [21], a lightweight architecture of YOLOv3, which is based on Glenn Jocher's Pytorch implementation of YOLOv3 [25] was used on this study.

YOLOv5-small

Like the previous versions of YOLO, single-shot object detection architectures, YOLOv5 follows the same working principles of its predecessors while having major improvements overall. The model size was drastically reduced, enabling it to be deployed on small-end devices while the speed and accuracy of the model was increased [28]. Self-adversarial training (SAT) and mosaic data augmentation [26, 27] approaches are used during training to improve the model's robustness, allowing it to detect target objects out of context from previously unseen data [27]. YOLOv5 takes into consideration the size variation of the target objects (i.e., identifies small, medium, and big target objects), leading to the development of three distinct sizes (18x18, 36x36, and 72x72) of feature maps, allowing for multi-scale prediction [28]. After careful consideration, YOLOv5-small, a smaller variant of YOLOv5 (v6.1-61-gbc3ed95) was chosen for this study for its optimum size, speed, and accuracy.

*EfficientDet-Lite*The YOLO

EfficientDet a family of object detection models based on Weighted Bi-directional Feature Pyramid Network (BiFPN) (improved Path Aggregation Network (PANet) [28] developed by Google, employs EfficientNet (ImageNet-based pre-trained model) [18] as its backbone with the goal of improving speed and accuracy. Branching out from this architecture, EfficientNet-Lite [29] was developed to optimize the performance of this architecture for mobile devices. To achieve this goal, the following changes were made using Tensorflow Model Optimization Toolkit: 1) removal of the Squeeze-and-Excitation (SE) building block, 2) post-training quantization for edge TPU, 3) substitution of SWISH activation with Relu6, and minor improvements to model backbone and head layer to improve efficiency [18,19].

*SSDLite Mobilenet v2*The YOLO

SSDLite, a lightweight architecture variant derived from SSD, is an excellent mobile one-shot detection architecture head. In combination with efficient backbone architectures like MobileNet V2 or MobileNet V3, SSDLite achieves SOTA performance on mobile and embedded devices [14]. Single Shot MutliBox Detector (SSD) was introduced as a single-shot-detector with improved speed and accuracy on low-resolution images targeting embedded systems. It provides a reasonable performance with a fair tradeoff between speed and accuracy [15].

MobileNetV2 is a neural Network architecture tailored to fit mobile computational constraints. This architecture is designed to minimize the computational memory requirement while trying to retain a reasonable accuracy. The major improvement of this architecture is attributed to its inverted residual layer with linear bottleneck, which inputs a low-dimension representation of image which is scaled up and passed on to the convolution layer. Later, the extracted features from the convolutional layer are projected back on to the low-resolution representation using a linear convolution, thereby reducing the computational requirements [16].

Model Evaluation

We evaluate the performance of each detection architecture using mean Average Precision (mAP) with an IoU threshold of 0.5. The average precision (AP) takes a predefined confidence threshold into consideration and computes the number of positive detections in each image that falls above the confidence threshold. Therefore, the predicted bounding boxes that has overlapping over ground truth bounding box with threshold below the predefined confidence threshold is considered negative. In a multi-class detection problem, the mean Average Precision (mAP), which is the average of Average Precision (AP) of all the class [33], can be computed using equation:

$$mAP = \frac{1}{C}\sum_{i=1}^{C} AP_i$$

COCO evaluation [31] reports object detection performance with two metrics, mAP50 and mAP50:95. For our evaluation, due to single class (soybean pod), both mAP and AP are the same. To measure the overlapping between the predicted bounding box and the ground truth, intersection over union is used. IOU represents the intersected area between the predicted bounding box Bp and the ground-truth bounding box Bgt divided by the area of their union [33]. IOU can be expressed as:

$$OU = \frac{area\ of\ overlap}{area\ of\ union}$$

Model training

We used NVIDIA RTX A5000 GPU to train YOLO 3, 5, SSDLite-MobileNet V2 and EfficientDet-lite0. All the models were trained for 300 epochs with an input size of 640x640 pixels. The batch size for single-shot detector were set at 64 during the training. Both YOLO architectures were trained using Pytorch 1.11.0+cu113 with default training hyperparameters. EfficientDet-lite0 was trained with maximum instances per image set to 2500, maximum detections set to 5000, at a batch size of 16. SSDLite MobileNet V2 was trained with an Intersection over Union IoU threshold of 60%, maximum detections and maximum number of bounding boxes set to 1000, at a batch size of 4 due to GPU constrictions. The model was trained for 50,000 steps, after which the inference is performed on test data. After training, the models were evaluated using Microsoft COCO-evaluation metrics [31] such as mAP@50 and mAP@0.5:0.95.

RESULT AND DISCUSSION

Results showed that YOLO v5 model outperformed other models in terms of average precision (Table 1) at 50% and 50-95% IoU thresholds. The results are based on the test performed on 997 images used for test dataset in this study.

Table 1: Coco-evaluation metrics of the model on test data.

DL Architecture	AP@50	AP@50:95
YOLO v3 tiny	61.8%	35.3%
YOLO v5 small	**87.9%**	**43.7%**
SSDLite MobileNet v2	2.2%	0%
EffecientDet0 Lite	1%	0%

After training for 300 epochs, the YOLOv5 showed superior performance by 26.1% improvement in AP@50 and 8% increase in AP@50:95 scores in comparison to its predecessor YOLOv3. This improvement can be attributed to its improved backbone network, which incorporates Cross Stage Partial Network (CSPNet) [30], resulting in significantly better performance at extracting features from the data. In addition, incorporation of SAT and MOSAIC augmentation techniques enabled the network to learn better from a more challenging training scenario. These techniques also improved the resilience of the model to changing background conditions and noise in the data.

Figure 1 shows the output of YOLOv5 on random images from test dataset where the soybean pods and the background color show some similarity. Based on the output, the model performs reasonably well on the data provided despite some false positives and false negatives. Evaluating the confusion matrix from Figure 2 , it is more evident that the model still struggles to classify the soybean pods from the background where 35% of the time it falsely classified the background as soybean pod due to similarity and noise. Some of next steps that will be taken to improve the model's performance are: 1) to train the model at a higher images size (e.g., 1024) which will improve the performance, at the cost of speed; and 2) additional data augmentation techniques that will alter the background of target object during training, which can be used to overcome background related issues.

Fig 1. This output shows the performance of YOLOv5 on different sample images randomly picked for soybean test dataset. While the pods were detected with reasonable accuracy, some background noises (debris fallen on ground) are also falsely classified as pods. Overall, the performance of the model is much efficient compared to manual counting.

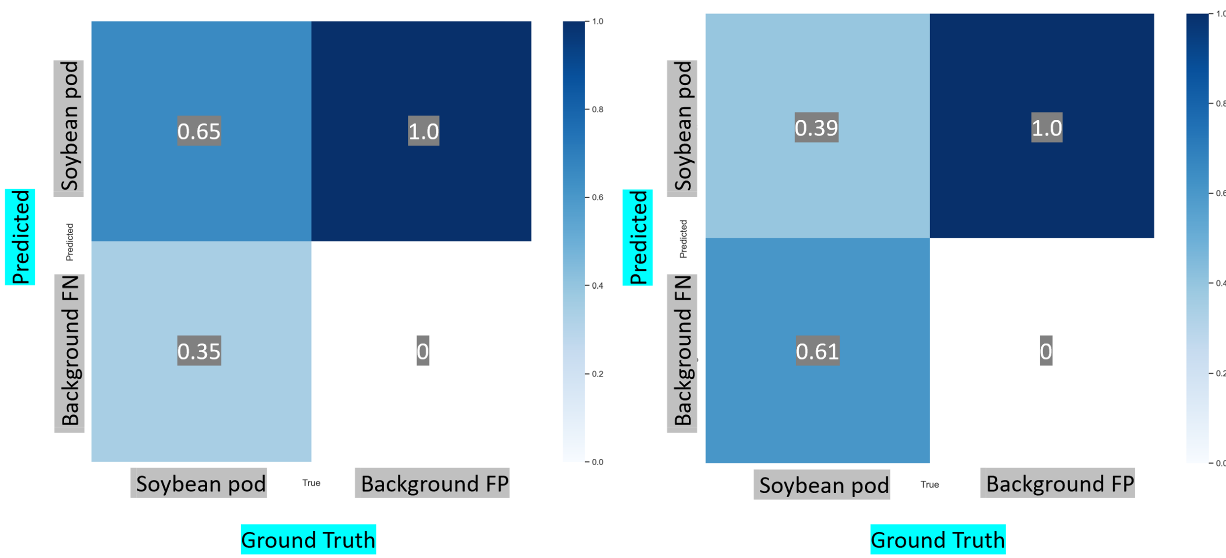

Figure 2: Confusion Matrix YOLOv5 (left) YOLOv3 (right) after model inference on soybean dataset.

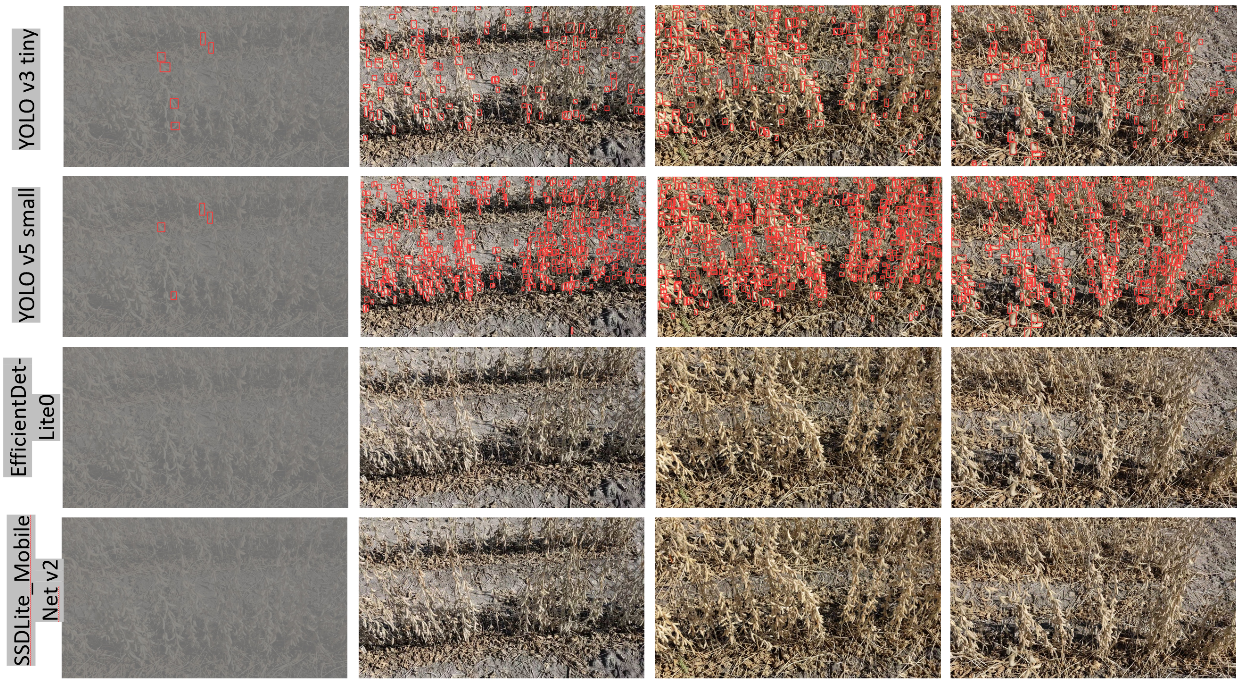

Figure 3: Performance comparison of light-weight DL based object detection architectures on four random soybean plants test images.

The performance of the SSDLite MobileNet v2 and EffecientDet0 Lite models was extremely poor on our dataset. This could imply that these models lack the ability to extract and learn from the features, especially when the target size of the object is very small, such as a soybean pod. A high similarity between the pod color and the background could be another limiting factor preventing these networks from performing well. To improve the detection performance, multiple viewpoint methods, as described by [13], can be introduced in the smartphone and fed into the DL algorithm for prediction, after which the detected bounding boxes can be combined and filtered to finalize and count the number of pods per row or plot.

Based on the results found on this study, it can be concluded that suggest that YOLOv5 small, without any prior modification to its neural network, performs better than other light-weight architectures when targeting to detect small objects such soybean pods still on the plants under field condition.

CONCLUSION

The main goal of this study was to find a suitable DL based object detection model that can be used in the future for estimation of soybean yield using pod count strategy. To achieve this goal, four different DL models, each with its own uniqueness in accuracy and speed, were trained and evaluated,

Based on the coco-evaluation metrics, YOLOv5 small showed the best performance among all the four models compared on this study. Not only did that model performed well, but it was also able to detect pods accurately under varying conditions induced by data augmentation techniques. When the model was deployed on an edge TPU device, it was evident that YOLOv5 is a good fit for detecting small objects like soybean pod and has the potential for in-field yield estimation. As we continue research on this subject matter, future studies will focus on aspects to improve YOLOv5 small's performance (train the model with larger image datasets and use of additional augmentation techniques) and to solve the occlusion and background issues for soybean yield estimation.

References

[1] Wei, Marcelo Chan Fu, and José Paulo Molin. "Soybean yield estimation and its components: A linear regression approach." Agriculture 10, no. 8 (2020): 348.

[2] Li, Yue, Jingdun Jia, Li Zhang, Abdul Mateen Khattak, Shi Sun, Wanlin Gao, and Minjuan Wang. "Soybean seed counting based on pod image using two-column convolution neural network." Ieee Access 7 (2019): 64177-64185.

[3] Khaki, Saeed, and Lizhi Wang. "Crop yield prediction using deep neural networks." Frontiers in plant science 10 (2019): 621.

[4] https://www.fao.org/faostat/en/#data/QCL

[5] Dietz, Nicholas, Rachel Combs-Giroir, Grace Cooper, Minviluz Stacey, Carrie Miranda, and Kristin Bilyeu. "5." BMC plant biology 21, no. 1 (2021): 1-13.

[6] Miranda, Carrie, Andrew Scaboo, Elroy Cober, Nicholas Denwar, and Kristin Bilyeu. "6." BMC Plant Biology 20, no. 1 (2020): 1-13.

[7] Oikonomidis, Alexandros, Cagatay Catal, and Ayalew Kassahun. "Deep learning for crop yield prediction: a systematic literature review." New Zealand Journal of Crop and Horticultural Science (2022): 1-26.

[8] Lu, Wei, Rongting Du, Pengshuai Niu, Guangnan Xing, Hui Luo, Yiming Deng, and Lei Shu. "Soybean Yield Preharvest Prediction Based on Bean Pods and Leaves Image Recognition Using Deep Learning Neural Network Combined With GRNN." Frontiers in Plant Science 12 (2022): 791256.

[9] Engen, M., Sandø, E., Sjølander, B. L. O., Arenberg, S., Gupta, R., & Goodwin, M. (2021). Farm-Scale Crop Yield Prediction from Multi-Temporal Data Using Deep Hybrid Neural Networks. Agronomy, 11(12), 2576

[10] Khaki, Saeed, Hieu Pham, and Lizhi Wang. "Yieldnet: A convolutional neural network for simultaneous corn and soybean yield prediction based on remote sensing data." bioRxiv (2021): 2020-12.

[11] Eugenio, Fernando Coelho, Mara Grohs, Luan Peroni Venancio, Mateus Schuh, Eduardo Leonel Bottega, Régis Ruoso, Cristine Schons, Caroline Lorenci Mallmann, Tiago Luis Badin, and Pablo Fernandes. "Estimation of soybean yield from machine learning techniques and multispectral RPAS imagery." Remote Sensing Applications: Society and Environment 20 (2020): 100397.

[12] McGuire, Michael, Chinmay Soman, Brian Diers, and Girish Chowdhary. "High Throughput Soybean Pod-Counting with In-Field Robotic Data Collection and Machine-Vision Based Data Analysis." arXiv preprint arXiv:2105.10568 (2021).

[13] Riera, Luis G., Matthew E. Carroll, Zhisheng Zhang, Johnathon M. Shook, Sambuddha Ghosal, Tianshuang Gao, Arti Singh et al. "13." Plant Phenomics 2021 (2021).

[14] Xiong, Yunyang, Hanxiao Liu, Suyog Gupta, Berkin Akin, Gabriel Bender, Yongzhe Wang, Pieter-Jan Kindermans, Mingxing Tan, Vikas Singh, and Bo Chen. "14." In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3825-3834. 2021.

[15] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "Ssd: Single shot multibox detector." In European conference on computer vision, pp. 21-37. Springer, Cham, 2016.

[16] Sandler, Mark, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. "Mobilenetv2: Inverted residuals and linear bottlenecks." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510-4520. 2018.

[17] Bruce, Robert W., Christopher M. Grainger, Alexandra Ficht, Mehrzad Eskandari, and Istvan Rajcan. "Trends in soybean trait improvement over generations of selective breeding." Crop Science 59, no. 5 (2019): 1870-1879.

[18] Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." In International conference on machine learning, pp. 6105-6114. PMLR, 2019.

[19] Nguyen, Huy-Hung, Duong Nguyen-Ngoc Tran, and Jae Wook Jeon. "Towards real-time vehicle detection on edge devices with nvidia jetson tx2." In 2020 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), pp. 1-4. IEEE, 2020.

[20] Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788. 2016.

[21] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).

[22] Lin, Tsung-Yi, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. "Feature pyramid networks for object detection." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2117-2125. 2017.

[23] Jiao, Licheng, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng, and Rong Qu. "A survey of deep learning-based object detection." IEEE access 7 (2019): 128837-128868.

[24] Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7263-7271. 2017.

[25] Jocher, Glenn. 2019. “yolov3.” https://github.com/ultralytics/yolov3.

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

[27] Yap, Moi Hoon, Ryo Hachiuma, Azadeh Alavi, Raphael Brüngel, Bill Cassidy, Manu Goyal, Hongtao Zhu et al. "Deep learning in diabetic foot ulcers detection: a comprehensive evaluation." Computers in Biology and Medicine 135 (2021): 104596.

[28] Xu, Renjie, Haifeng Lin, Kangjie Lu, Lin Cao, and Yunfei Liu. "A forest fire detection system based on ensemble learning." Forests 12, no. 2 (2021): 217.

[29] https://blog.tensorflow.org/2020/03/higher-accuracy-on-vision-models-with-efficientnet-lite.html

[30] Wang, Chien-Yao, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. "CSPNet: A new backbone that can enhance learning capability of CNN." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, pp. 390-391. 2020.

[31] Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. "Microsoft coco: Common objects in context." In European conference on computer vision, pp. 740-755. Springer, Cham, 2014.

[32] Chen, Qilei, Ping Liu, Jing Ni, Yu Cao, Benyuan Liu, and Honggang Zhang. "Pseudo-labeling for small lesion detection on diabetic retinopathy images." In 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1-8. IEEE, 2020.

[33] Padilla, Rafael, Wesley L. Passos, Thadeu LB Dias, Sergio L. Netto, and Eduardo AB da Silva. "A comparative analysis of object detection metrics with a companion open-source toolkit." Electronics 10, no. 3 (2021): 279.

[34] Ferencz, Cs, P. Bognar, János Lichtenberger, Daniel Hamar, Gy Tarcsai, Gábor Timár, Gábor Molnár et al. "Crop yield estimation by satellite remote sensing." International Journal of Remote Sensing 25, no. 20 (2004): 4113-4149.

[35] Yang, Siqi, Ling Hu, Haobo Wu, Huazhong Ren, Hongbo Qiao, Peijun Li, and Wenjie Fan. "Integration of crop growth model and random forest for winter wheat yield estimation from UAV hyperspectral imagery." IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 14 (2021): 6253-6269.

[36] Tzutalin. LabelImg. Git code (2015). https://github.com/tzutalin/labelImg

[37] Mathew, Jithin Jose, Yu Zhang, Paulo Flores, Cannayen Igathinathane, and Zhao Zhang. "Development and Test of an RGB-D Camera-based Rock Detection System and Path Optimization Algorithm in an Indoor Environment." In 2021 ASABE Annual International Virtual Meeting, p. 1. American Society of Agricultural and Biological Engineers, 2021.