AI-AUGMENTED MONITORING AND MANAGEMENT BY IMAGE ANALYSIS FOR OBJECT DETECTION AND COUNTING

An Undergraduate Research Scholars Thesis

by

HUAXING REN

Submitted to the LAUNCH: Undergraduate Research office at
Texas A&M University
in partial fulfillment of requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by
Faculty Research Advisor: Dr. Xiaoning Qian

May 2022

Major: Electrical Engineering

RESEARCH COMPLIANCE CERTIFICATION

Research activities involving the use of human subjects, vertebrate animals, and/or biohazards must be reviewed and approved by the appropriate Texas A&M University regulatory research committee (i.e., IRB, IACUC, IBC) before the activity can commence. This requirement applies to activities conducted at Texas A&M and to activities conducted at non-Texas A&M facilities or institutions. In both cases, students are responsible for working with the relevant Texas A&M research compliance program to ensure and document that all Texas A&M compliance obligations are met before the study begins.

I, Huaxing Ren, certify that all research compliance requirements related to this Undergraduate Research Scholars thesis have been addressed with my Research Faculty Advisor prior to the collection of any data used in this final thesis submission.

This project did not require approval from the Texas A&M University Research Compliance & Biosafety office.

TABLE OF CONTENTS

Page

ABSTRACT

AI-augmented Monitoring and Management by Image Analysis for Object Detection and Counting

Huaxing Ren
Department of Electrical & Computer Engineering
Texas A&M University

Research Faculty Advisor: Dr. Xiaoning Qian
Department of Electrical & Computer Engineering
Texas A&M University

Counting the number of objects from images has become an increasingly important topic in different applications, such as crowd counting, cell microscopy image analyses in biomedical imaging, and horticulture monitoring and prediction. Many studies have been working on automatic object counting with Convolutional Neural Networks (CNNs). This research is aimed to shed more light on the applications of deep learning models to count objects in images in different places, such as growing fields, classrooms, streets, etc. We will study how CNN predicts the numbers of objects and measure the accuracy of trained models with different training parameters by using evaluation metrics, mAP, and RMSE. The performance of object detection and counting using a CNN, YOLOv5, will be analyzed. The model will be trained on the Global Wheat Head Detection 2021 dataset for crop counting and COCO dataset for counting of labeled objects. The performance of the optimized model on crowd counting will be tested with pictures taken on the Texas A&M University campus.

DEDICATION

To my friends, families, instructors, and peers who supported me throughout the research process.

ACKNOWLEDGEMENTS

Contributors

I would like to thank my faculty advisor, Dr. Qian, and my mentor, Yucheng Wang for their guidance and support throughout the course of this research.

Thanks also go to my friends and colleagues and the department faculty and staff for making my time at Texas A&M University a great experience.

Finally, thanks to my family for their encouragement, patience and love.

Funding Sources

Undergraduate research was supported by Department of Electrical & Computer Engineering at Texas A&M University and an additional research fellowship from Research Experiences for Undergraduates (REU) program supported by the National Science Foundation.

NOMENCLATURE

CNN	Convolutional Neural Network
COCO	Common Objects in Context Dataset
YOLOv5	You Only Look Once Version 5
GWHD_2021	Global Wheat Head Detection 2021 Dataset
RMSE	Root Mean Square Error
mAP	Mean Average Precision
HPRC	High Performance Research Computing

1. INTRODUCTION

1.1 Background

Object number estimation is important for people living in modern society to make decisions. Counting objects of interest with quantified uncertainty can address many real-world decision-making challenges, including smart and sustainable agriculture, road safety in particular with autonomous driving vehicles, as well as disaster management, for example, maintaining social distances in public space under the current COVID situation. While past efforts in manual surveying are time-consuming and result in a high error rate, AI-augmented analyses of remote images can lead to automatically tracking and monitoring object numbers with machine learning. This application of AI and machine learning has the potential for usage in crowd and traffic counting of smart cities, cell microscopy image analysis in biomedical imaging [1], and agricultural monitoring [2].

1.2 Plant-density Monitoring

Plant-density monitoring plays an important role in making decisions regarding irrigating, applying fertilizers, and cultivating to manage crop productivity and sustainability. Since 2018, there have already been some studies [3] with machine learning to train prediction models with agricultural datasets based on UAV images to achieve visual recognition of plants. Recent studies reported limited performances due to varying background appearances with indistinguishable color features. For example, plants can have similar colors or be occluded due to imaging angles. Also, due to the variant distances between cameras and objects, the object sizes and geometries vary. In the same images, the object closer to the camera occupies more pixels than the objects far away from the camera. The challenge also happens when counting other kinds of objects on other large-scale occasions, such as crowds. Lastly, it is also desired to quantify the uncertainty of objects' counting to help corresponding decision making and improve the techniques of density monitoring.

1.3 Crowd Monitoring

Crowd management is critical to dealing with the problem of crowd crushes [4], resource management and distribution, traffic control [5], etc. During the COVID-19 pandemic, detecting crowds can help maintain the social distance in public spaces such as classrooms, concerts, and churches. Traditional methods can be tedious and inaccurate. For example, the hardware-based crowd counting method utilized sensors to count people and registers to record the number. The accuracy of this method relies on the capability of sensors and the speed of people's movements. [5] Also, sensors cannot detect multiple objects when they are overlapping in front of sensors. In contrast, artificial intelligence and machine learning have been used to detect people with video feeds nowadays with the ability to count objects in a wide range in real-time. It can detect objects by detecting features with either an entire or partial image of an object.

1.4 Previous Works

When it comes to solving the problem of counting objects in images or videos, there are two main kinds of techniques, object detection and regression (segmentation regression and density regression [6, 7]). In the application of agriculture management and crowd monitoring, many researchers also utilized density map estimation to count a number of objects by integrating the possible number of objects of each pixel in images [8]. For the object detection method, there are existing research works that were using different CNNs, such as DarkNet and previous versions of YOLO algorithms [9]. But both of them show limitations when it comes to the high-dense crowd problem.

1.5 Convolutional Neural Network

Convolutional Neural Networks (CNNs) is a type of deep learning models widely used in image processing and computer vision. It consists of many different kinds of layers, such as convolutional, pooling, and fully connected layers. CNN-based approaches can learn the robust object characteristics such as the edges, texture, and parts of objects [10].

1.5.1 Convolutional Layer

Convolutional layers slide the convolutional image filters on each image with a certain length of stride. Sometimes zero padding can be needed to make sure that pixels at the edges are included in the filtering operation.

1.5.2 Pooling Layer

There are two main types of pooling layers, max pooling and average pooling. They are used to get feature maps and reduce the image dimensions. Reducing image dimensions can decrease the number of parameters of the neural network, and then reduce the model complexity.

1.6 Research Objectives

Our work is expected to understand the underlying architecture of (CNNs) and improve the current method to overcome the challenges of object counting using different images. In this project, we use the YOLOv5 object detection model as our baseline model. YOLOv5 will be trained first and predict the location of objects. The datasets we will utilize to train the models are GWHD_2021 [11] and COCO [12]. We will evaluate the performance of the YOLOv5 model on object detection and counting accuracy, and analyze its deficiency with the corresponding evaluation metrics, RMSE, and mAP.

2. METHODS

2.1 Datasets

There are two datasets used for this project, GWHD_2021 [11] and COCO [12]. GWHD_2021 dataset contains 6500 RGB images taken from 16 institutions in 12 countries with 275,000 wheat heads [11]. These are images of 1024*1024 pixels with bounding boxes of wheat heads as labels with the goal of detecting and counting the wheat heads as the object category of interest. Figure 2.1 displays some examples from this dataset. The entire dataset was split into a model set and a test set with the ratio of 9:1. The model set was split further into a training set and a validation set with a ratio of 9:1. The authors of GWHD_2021 also provided the corresponding python script program in the scikit-learn framework. COCO dataset has 117,265 JPG images with bounding boxes as labels and 80 object categories, such as persons, bicycles and cars [12]. Figure 2.2 shows some image examples from COCO.

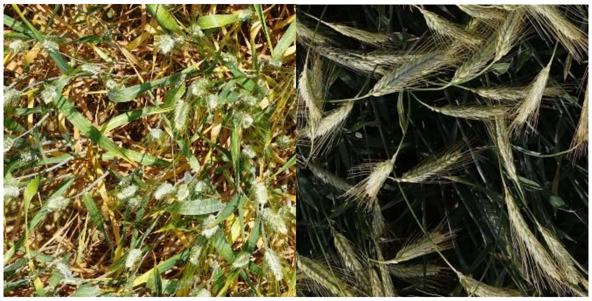


Figure 2.1: Examples of GWHD_2021 dataset with different backgrounds, densities, and colors. [11]

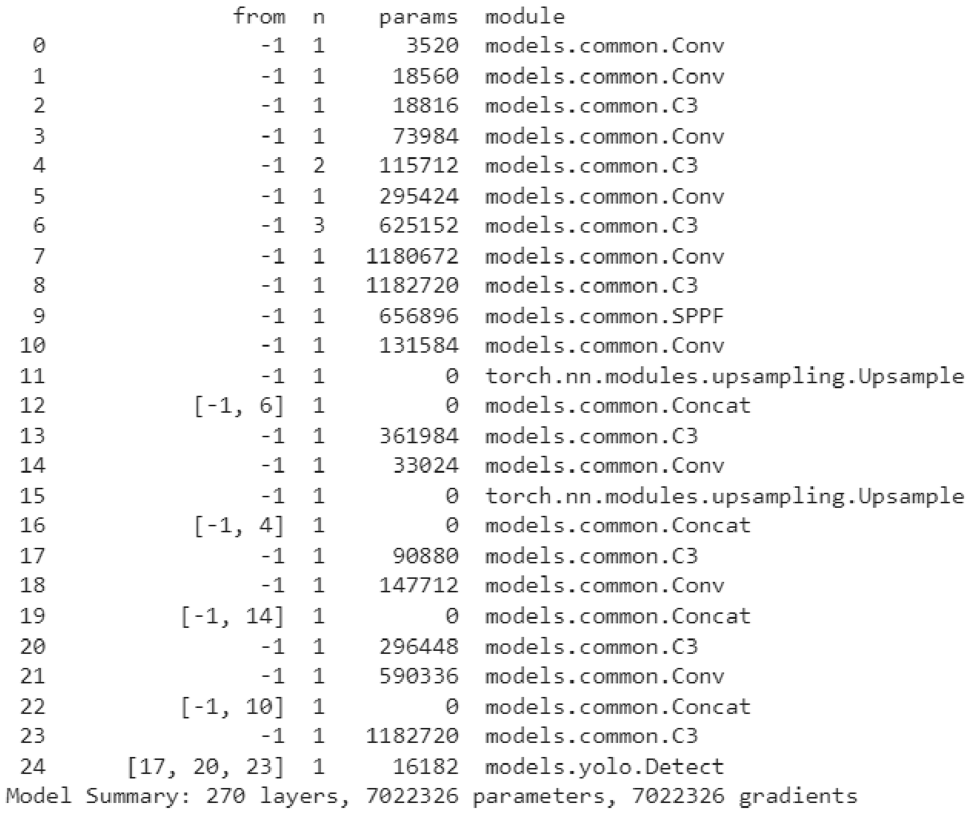
Figure 2.2: Examples of COCO dataset on different occasions. (Left) Fields. (Middle left) Streets. (Middle right) Crowds. (Right) Rooms. [12]

2.2 YOLOv5

2.2.1 Model Architecture

YOLOv5 is a state-of-the-art object detection model built in the PyTorch framework. In this project, we trained the YOLOv5 model and used it to make predictions on images. This model is based on CNNs in terms of its architecture. It has a CNN backbone for pre-training to extract features, a "neck" of CNN layers for getting feature pyramids and feature fusion, and a "head" of the output layer for finding and classifying objects. Using CSPDarknet as the backbone can minimize the model parameter size while still ensuring the accuracy and speed of training the model. The feature pyramids obtained by the neck, PANet, have a bottom-up structure, which enables the model to propagate low-level features. Finally, the YOLO output layer, as the head of the model, can extract different sizes of features and realize a multi-scale prediction.

The details of the YOLOv5 architecture are shown in Figure 2.3. Each column in the figure means the index of modules, the input channels, the number of modules, and the number of parameters. The value "-1" as input channels means that the input channel is from the last module. The CSPDarknet is from the 0th module to the 9th module. The PANet is from the 10th to the 23rd module. The yolo layer is the 24th module that makes predictions with anchors. Within these modules, the SPPE block can separate the significant features.



```
                 from  n    params  module
  0                -1  1      3520  models.common.Conv
  1                -1  1     18560  models.common.Conv
  2                -1  1     18816  models.common.C3
  3                -1  1     73984  models.common.Conv
  4                -1  2    115712  models.common.C3
  5                -1  1    295424  models.common.Conv
  6                -1  3    625152  models.common.C3
  7                -1  1   1180672  models.common.Conv
  8                -1  1   1182720  models.common.C3
  9                -1  1    656896  models.common.SPPF
 10                -1  1    131584  models.common.Conv
 11                -1  1         0  torch.nn.modules.upsampling.Upsample
 12           [-1, 6]  1         0  models.common.Concat
 13                -1  1    361984  models.common.C3
 14                -1  1     33024  models.common.Conv
 15                -1  1         0  torch.nn.modules.upsampling.Upsample
 16           [-1, 4]  1         0  models.common.Concat
 17                -1  1     90880  models.common.C3
 18                -1  1    147712  models.common.Conv
 19          [-1, 14]  1         0  models.common.Concat
 20                -1  1    296448  models.common.C3
 21                -1  1    590336  models.common.Conv
 22          [-1, 10]  1         0  models.common.Concat
 23                -1  1   1182720  models.common.C3
 24      [17, 20, 23]  1     16182  models.yolo.Detect
Model Summary: 270 layers, 7022326 parameters, 7022326 gradients
```

Figure 2.3: YOLOv5 architecture.

For the training process, a loss function called Binary Cross-Entropy (BCELoss) with Logits is used to calculate the class probability for each detection to measure the prediction precision of each training sample. When having more than one object of interest, this loss function is extended with a Sigmoid layer to combine the loss for each single object class.

2.2.2 Data Augmentation

Data Augmentation is a technique that enlarges the dataset during training by making changes to original images. Some examples of changes could be flips, translations, rotation and crops. In YOLOv5, methods called mosaic (Figure 2.4) and random perspective (Figure 2.5) have been used for data augmentation. Mosaic is to randomly select four images from the dataset, cut them into pieces, and then crop these pieces together to form a new image. New labels of bounding boxes are generated corresponding to the cropped image. In addition to the mosaic method, YOLOv5 increases the number of data by randomly distorting images and changing their perspective.

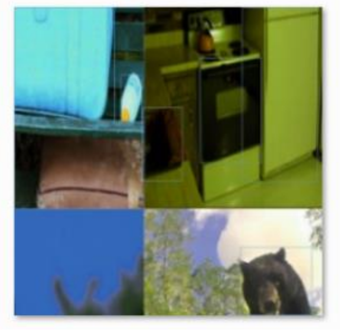


Figure 2.4: Mosaic examples with images from COCO dataset.

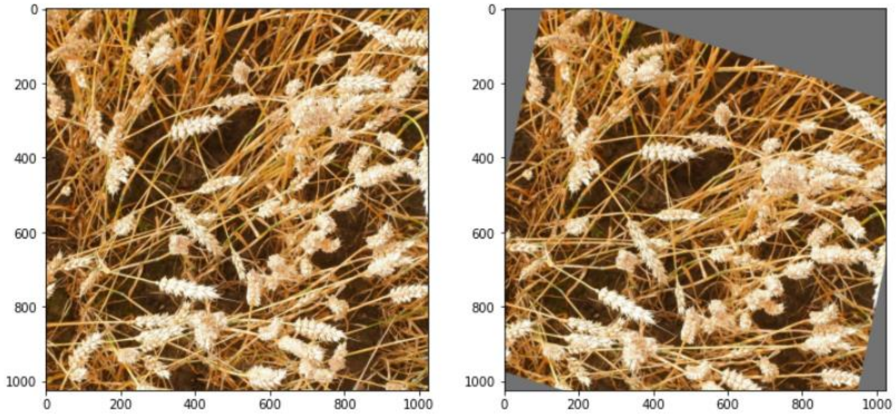


Figure 2.5: Random Perspective example with images from GWHD2021 dataset. The scale is 1024 1024 pixels. (Left) The original image. (Right) The image after doing the operation.*

2.3 Metrics and Uncertainty Quantification

In this study, we utilized two metrics to evaluate the object detection and counting performance of the YOLOv5 model, mAP, and RMSE.

The evaluation metric mAP is to measure the accuracy of object detection. The mAP of each image is calculated by the area of precision-recall (PR) curve. We estimated precision and recall based on the values of the four terms: true positive, false positive, false negative and true negative. We distinguish if a prediction is a true positive or a false positive with an IoU threshold of 0.5. Here are the corresponding definitions:

- IoU: The overlap area between the predicted bounding box and the ground truth bounding box (the real bounding on the target object). The formula of IoU is shown in equation 2.1.

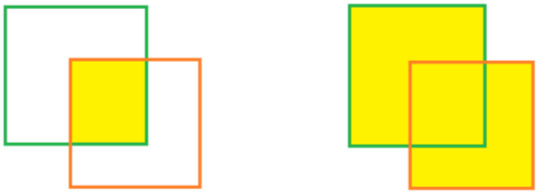


Figure 2.6: IoU. Green boxes mean bounding boxes of ground truth. Red boxes are the prediction results. (Left) The area of overlap. (Right) The area of union.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{2.1}$$

- True positive: The total number of predictions that is correct while the IoU is greater than the threshold, e.g. 0.5.
- False positive: The total number of incorrect predictions when it detects non-existing objects, or correct predictions while the IoU is less than the threshold.
- False negative: The number of ground truth bounding boxes that are not detected.
- Recall: The ratio of making the correct positive predictions among all ground-truth targets.

$$Recall = \frac{all\ true\ positive}{all\ ground\ truth} \tag{2.2}$$

- Precision: The ratio of making the correct positive predictions among all predictions.

$$Precision = \frac{all\ true\ positive}{all\ predictions} \tag{2.3}$$

- AP (average precision) is the area under the precision-recall curve with a certain threshold. mAP is the average AP for all classes. In our study, we first use 0.5 IoU as a threshold. Then we calculated mAP for many IoU thresholds from 0.5 to 0.95.

The other metric that we utilized to evaluate the prediction accuracy is RMSE. This metric is based on measuring the accuracy of counting the number of objects. In Equation 2.4 of calculating RMSE, A represents the number of images in the training set. N_{I_a} is the ground truth of the number of objects according to labels. ${N^{est}}_{I_a}$ is the counting results of the trained model.

$$RMSE = \sqrt{\frac{1}{A}\sum_{a=1}^{A}\left|N_{I_a} - {N^{est}}_{I_a}\right|^{2}} \tag{2.4}$$

3. RESULTS

3.1 Training Process

The YOLOv5 model was trained with the clusters in the High Performance Research Computing (HPRC) of Texas A&M University. We trained YOLO with GWHD_2021 and visualized the loss during the training process with up to 94 epochs. The objectiveness loss and box detection loss are shown in Figures 3.1 and 3.2. The visualization helped us to make sure the neural network model is working properly when we were testing the model.

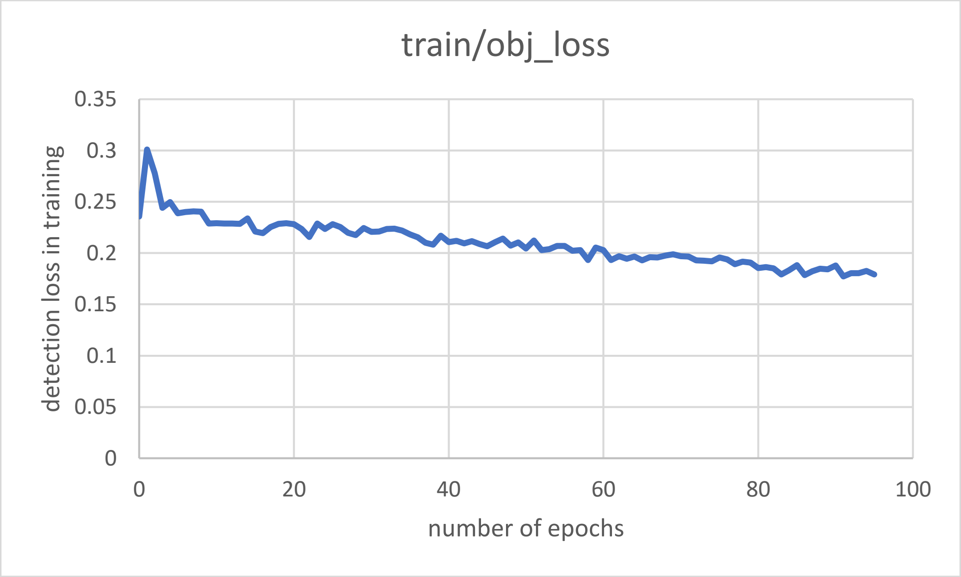


Figure 3.1: Objectiveness Loss of training process.

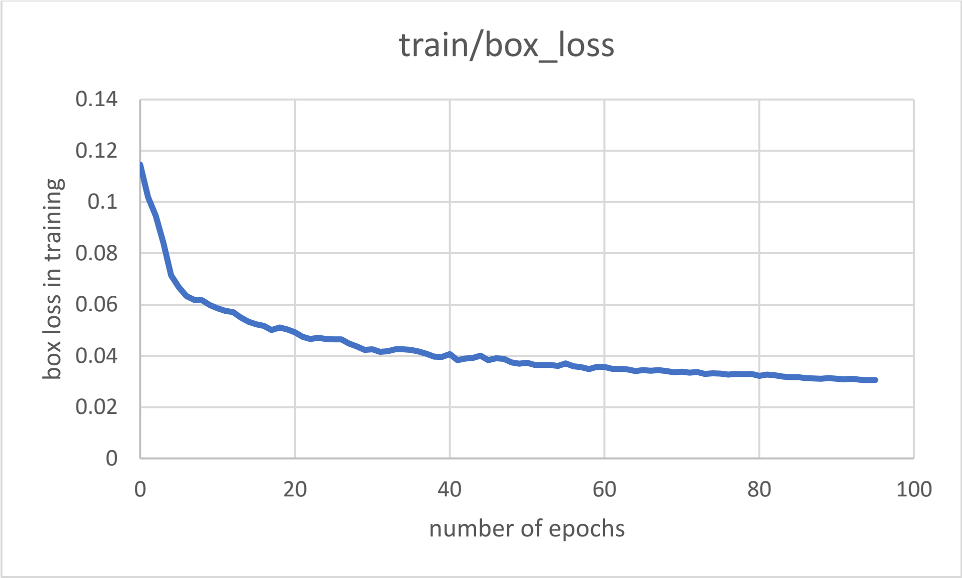


Figure 3.2: Objectiveness Loss of training process.

3.2 Performance Analysis on GWHD_2021

The original YOLOv5 model was first trained with GWHD_2021 dataset. For the training process, we chose 32 as the batch size to reduce the training time. We trained the model with different sizes of epochs, 9, 50, 100, and 150. Here is an example of the detection results with a trained model in Figure 3.3. The floating-point number on each bounding box means the class probability.

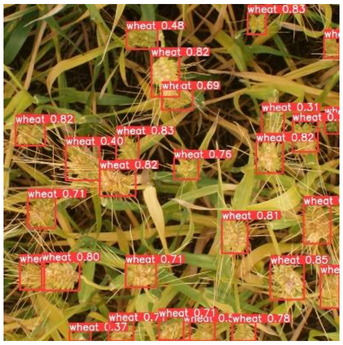


Figure 3.3: Detection result of 50-epoch trained YOLOv5 model.

There are two metrics used for evaluating the performance of the training process, mAP of validation loss and RMSE. The evaluation results shown below are all based on the validation subset of GWHD_2021 dataset.

The mAP is to evaluate the accuracy of localizing the detected object based on Intersection over Union (IoU). With mAP, we can find the trained model with the highest accuracy and avoid overfitting. The TensorFlow framework is utilized to visualize the loss of the training process. We first evaluated the trained model with IoU threshold of 0.5 for the epochs from 0 to 150 as shown in Figure 3.4. We defined 0.5 to 0.95 as the IOU threshold to separate the prediction of true positive and false positive. Figure 3.5 shows that the mAP reached a high value at epoch 50. It indicates that the trained model with 50 epochs is the most precise one among all trained models with 0.938 of mAP 0.5 and 0.525 of mAP 0.5: 0.9.

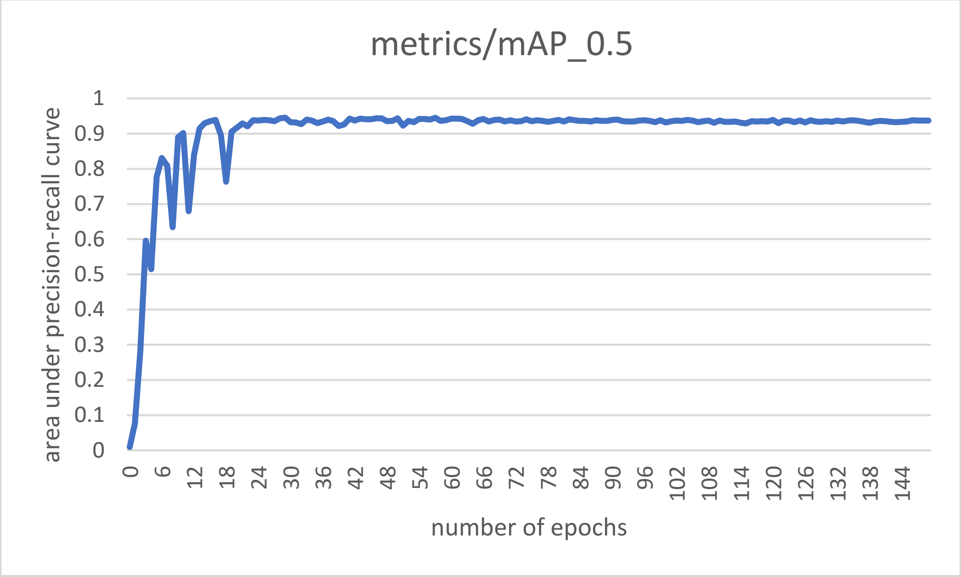


Figure 3.4: mAP-based evaluation result with 0.5 IoU threshold.

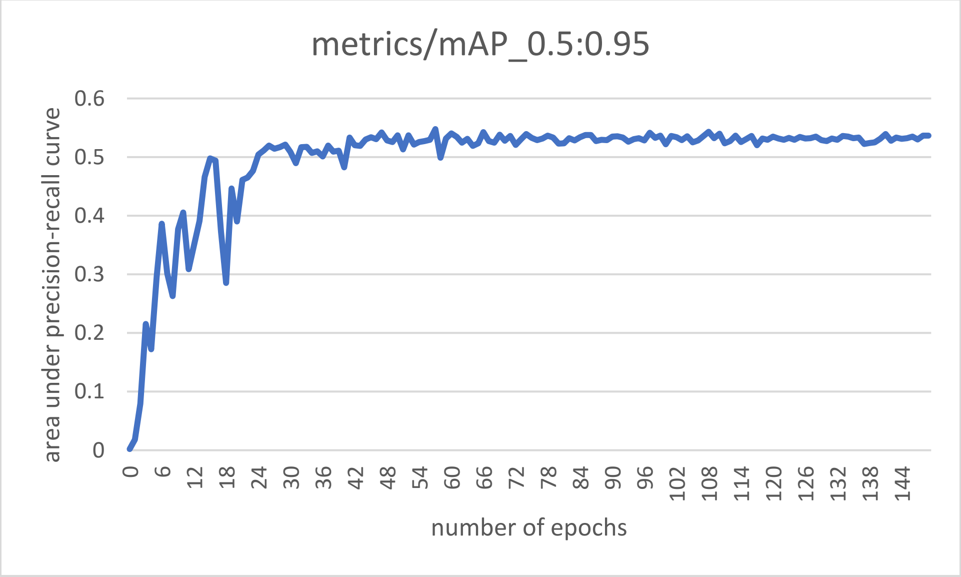


Figure 3.5: mAP-based evaluation result with 0.5:0.05:0.95 threshold.

In addition to mAP, we calculated the box loss and objectiveness loss in validation process as shown in Figures 3.6 and 3.7. The box loss is the regression loss for output position of bounding boxes, i.e. position on x-axis and y-axis and width and height, showing how well the bounding boxes cover the target object. The objectiveness loss relates to the probability that there is a target object in the bounding boxes.

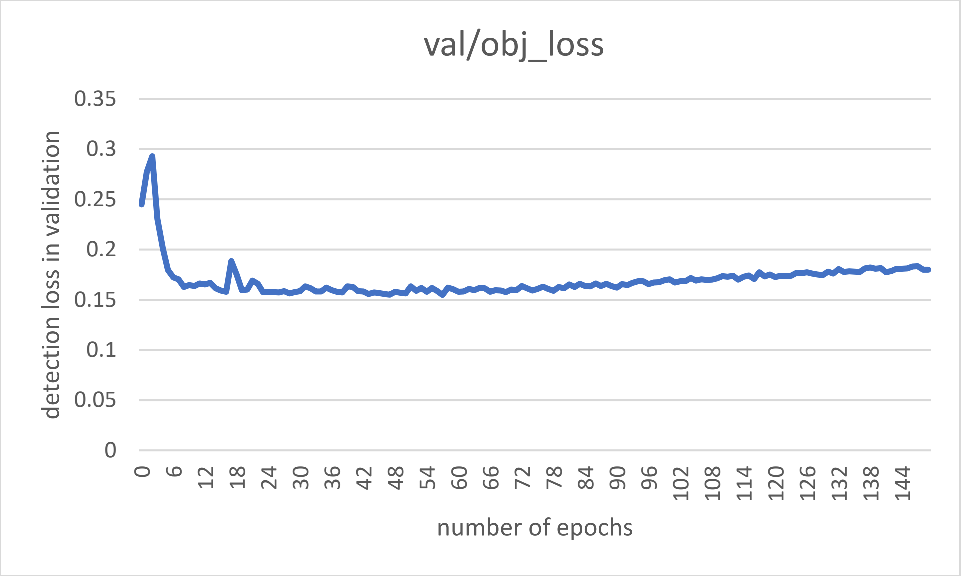


Figure 3.6: Objectiveness Loss.

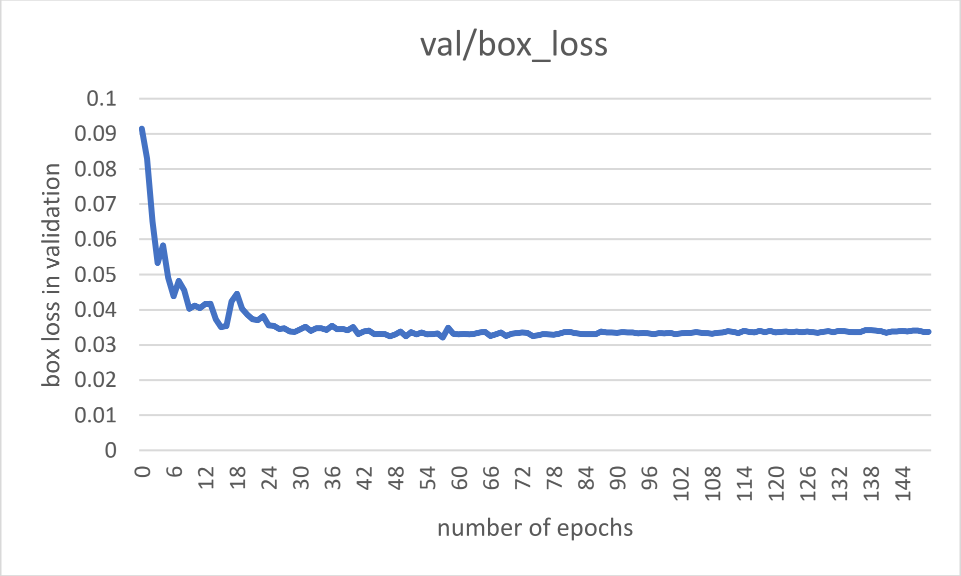


Figure 3.7: Box Loss.

Since the mAP results showed the trained models with 50 epochs and 100 epochs have close performance, we used RMSE to evaluate their performance in counting the number of wheat heads as shown in Table 3.1. After comparison, the trained YOLOv5 model with 50 epochs has the best performance. We also compared the performance of our proposed solution with the existing method based on the density map for the purpose of object counting, which is called Domain-Adversarial Neural Network [13]. The researchers used ACID dataset to train the neural network and adapted it to the GWHD dataset in parallel. As we can see, our method reduced the RMSE by 20.3. The accuracy has been approved by about 56.7%.

Table 3.1: RMSE-based evaluation result.

Models	RMSE
Yolov5 trained with 50 epochs	15.5
Yolov5 trained with 100 epochs	17.2
Domain-Adversarial Neural Network [13]	35.8

3.3 Performance on Customized Crowd Dataset

We trained YOLOv5 model with COCO dataset in order to see its capability of detecting and counting object on different occasions. We tested the trained model with the picture with different levels of crowds on campus of Texas A&M University. Figure 3.8 shows some examples of prediction results. It can detect and classify objects precisely due to the good performance of the feature extraction layers in YOLOv5 architecture. However, it performed less accurately when the resolution of images was not high enough. This is partially due to the limitation of COCO dataset that focuses on classifying objects and does not contain images with high-dense objects. Also, the counting-of-objects feature requires further improvement in terms of accuracy. This is why we plan to use a counting regression branch to combine with YOLOv5 object-detection-based method for further improvement in the future.

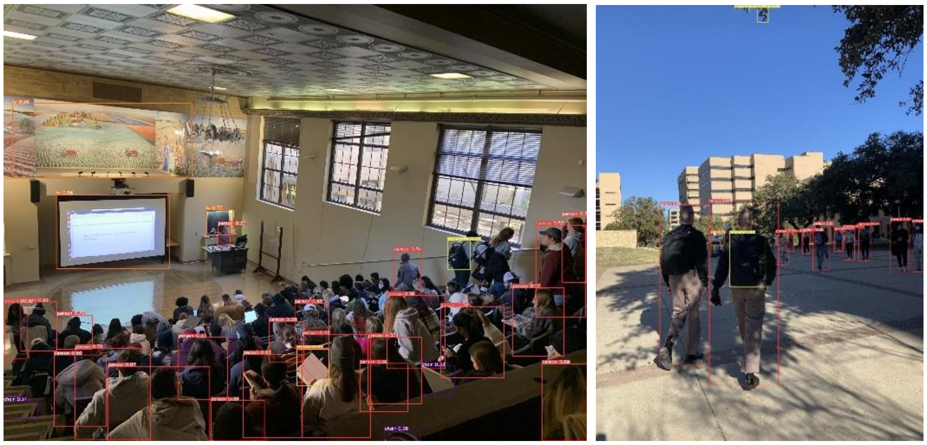


Figure 3.8: Crowd detection results of campus images. (Left) The image in a classroom with a high-dense crowd. (Right) The outdoor image with a less-dense crowd.

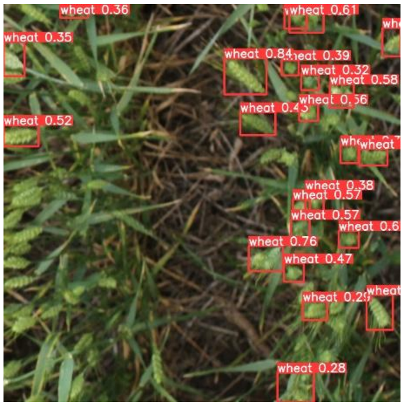


Figure 3.9: The detection example of low-resolution images in GWHD_2021 dataset with a 50-epoch trained YOLOv5 model.

4. CONCLUSION

4.1 Summary

We have implemented and tested the YOLOv5 convolutional neural network architecture for object detection and counting in different places, including classrooms, streets, and farms. Our results have confirmed that our method is reliable and can achieve reasonable counting accuracy. The counting accuracy is improved by comparing to a previous result on the same dataset. We keep developing our methods for more accurate and robust counting with uncertainty quantification.

In the future, we plan to use multi-task learning for counting tasks with high-dense objects in images. Multi-task learning method means a model is learning multiple tasks simultaneously, such as the combination of classification task and segmentation regression task [14]. It has been used in many projects of objection detection. Due to the robustness of the YOLO models in the past many years, many researchers have applied multi-task learning with YOLO based on the given task. For example, for tracking the driving route, the segmentation and object detection tasks are performed at the same time to detect drivable area and traffic flows respectively. In our study, in order to improve the performance of YOLOv5 on counting, we plan to perform count regression and YOLOv5 at the same time.

YOLOv5 makes predictions on locations and categories of objects in images. Its YOLO layer, the head of the model, can output the prediction based on the extracted features. To improve the counting accuracy further, an additional head is needed. Our planned multi-task learning method is to use linear regression as a counting head along with the original head of YOLOv5. The backbone and the neck of YOLOv5 will be retained as a baseline model because their extracted features will be used to train the counting head. In other words, the high-level features will be frozen and not be updated by the counting head. The rest of low-level features will be copied to the counting head for further training. Instead of bounding boxes on objects, the label for the counting head is the number of objects in each object class in images. In our ablation studies, different loss functions will be tested, including mean square error (MSE) and mean absolute error (MAE). When training the multi-task YOLO model, we will weigh the object detection loss and the counting loss accordingly to identify the best-performing architecture for counting.

REFERENCES

[1] O. Ronneberger, P. Fischer, and T. Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015. 234-241.

[2] Liu, Xueting. "An AI-Horticulture Monitoring And Prediction System With Automatic Object Counting." Master Thesis, *Texas A&M University*, 2019.

[3] W. Guo, B. Zheng, A. B. Potgieter, J. Diot, K. Watanabe, K. Noshita, D. Jordan, X. Wang, J. Watson, S. Ninomiya. "Aerial imagery analysis quantifying appearance and number of sorghum heads for applications in breeding and agronomy." *Frontiers in plant science*. 2018. 1544.

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

[5] U. Bhangale, S. Patil, V. Vishwanath, P. Thakker, A. Bansode, D. Navandhar, “Near Real-time Crowd Counting using Deep Learning Approach.” *Procedia Computer Science*. Vol. 171. 2020. 770-779. doi.org/10.1016/j.procs.2020.04.084.

[6] V. Badrinarayanan, A Kendall, R. Cipolla, " SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2017. 01. doi: 10.1109/TPAMI.2016.2644615.

[7] D. Zhou and Q. He, "Cascaded Multi-Task Learning of Head Segmentation and Density Regression for RGBD Crowd Counting," *IEEE Access*, vol. 8. 2020, doi: 10.1109/ACCESS.2020.2998678. 101616-101627.

[8] Liu, Xueting. "An AI-Horticulture Monitoring And Prediction System With Automatic Object Counting." *Master Thesis, Texas A&M University*, 2019.

[9] G. Castellano, C. Catiello, M. Cianciotta, C. Mencar. “Multi-view Convolutional Network for Crowd Counting in Drone-Captured Images.” *Computer Vision – ECCV 2020 Workshops*. 2020. 08. 588-603. doi:10.1007/978-3-030-66823-5_35

[10] D. Maulud, A. Abdulazeez. “A Review on Linear Regression Comprehensive in Machine Learning." *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol 169. 2020. 11. 280-291

[11] E. David, M. Serouart, D. Smith, S. Madec, K. Velumani, S. Liu, Xu Wang, F. Pinto, S. Shafiee, I. S. A. Tahir, H. Tsujimoto, S. Nasuda, B. Zheng, N. Kirchgessner, Helge Aasen, A. Hund, P. Sadhegi-Tehran, K. Nagasawa, G. Ishikawa, S. Dandrifosse, A. Carlier, B. Dumont, B. Mercatoris, B. Evers, K. Kuroki, H. Wang, M. Ishii, M. Badhon, C. Pozniak, D. Shaner LeBauer, M. Lillemo, J. Poland, S. Chapman, B. Solan, F. Baret, I. Stavness, W. Guo, "Global Wheat Head Detection 2021: An Improved Dataset for Benchmarking Wheat Head Detection Methods." *Plant Phenomics.* vol. 2021. 9. doi.org/10.34133/2021/9846158

[12] T. Lin, M. Maire, S. Belongie, L. Bourdev,R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. “Microsoft COCO: Common Objects in Context.” *Computer Vision and Pattern Recognition (cs.CV), FOS: Computer and information sciences, FOS: Computer and information sciences.* 2014. arXiv:1405.0312v3.

[13] T. Ayalew, J. Ubbens, I. Stavness. “Unsupervised Domain Adaptation For Plant Organ Counting." *Computer Vision Problems in Plant Phenotyping (CVPPP).* 2020. 09. 14. arXiv:2009.01081v1.

[14] D. Wu, M. Liao, W. Zhang, X. Wang, "YOLOP: You Only Look Once for Panoptic Driving Perception." *Computer Vision and Pattern Recognition (cs.CV).* Vol. 6. 2022. arXiv:2108.11250v6