DEEP LEARNING-BASED OBJECT DETECTION OF BARLEY SEEDS

by

Jiayi Li

A Thesis Submitted in

Partial Fulfillment of the

Requirements for the Degree of

Master of Science

in Computer Science

at

The University of Wisconsin-Milwaukee

May 2021

ABSTRACT

DEEP LEARNING-BASED OBJECT DETECTION OF BARLEY SEEDS

by

Jiayi Li

The University of Wisconsin-Milwaukee, 2021
Under the Supervision of Professor Zeyun Yu

Computer vision techniques have been widely used in the food manufacturing industry today due to their speed, convenience, and low labor cost. As an important raw material in beer, barley seeds largely determine the flavor and taste of the beer brewing process. To ensure the quality of beer, beer brewers will strictly screen the varieties of barley seeds and ensure the purity of malt. Traditional manual detection needs a lot of professional training, and because of the high similarity between different kinds of barley seeds, tedious and time-consuming manual detection leads to a high error rate. However, chemical testing requires professional equipment, reagents, and laboratories, which have high-cost performance. Thus, an efficient and accurate object detection technique is considered to replace manually distinguishing barley seed types. This thesis uses the deep learning-based object detection network YOLOv3 model to help automatically and accurately detecting barley locations and identifying seed types from iPhone-based images of barley seeds. The barley seed samples used in this project are all provided by our industrial collaborator, and captured by iPhone 11 or iPhone 11 pro in high-definition resolutions. A total of nine varieties of barleys are trained in this study, and the data set includes images of a single grain and images of multiple grains (multi-categories). In this experiment, the

best mAP (mean Average Precision) value we obtained is 97.1%, the model recognition (localization and classification) precision is 91.2%, and the recall rate is 95.9%.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

This thesis project benefited from conversations with many intelligent people in the Big Data Analytics and Visualization Laboratory at the University of Wisconsin- Milwaukee. Professor Zeyun Yu gave me lots of high-level advice from his years of experience. I would like to offer my gratitude to my teammate Yaying Shi, who works with me on this project. I would also like to thank our lab group for providing their knowledge, help, and insightful suggestions. Last but not least, this thesis would not become possible without the valuable guidance and dataset kindly provided by Dr. Yin Li from MaltEurop.

1 Introduction

1.1 Structure and Variety of Barley Seeds

In botanical terms, a barley seed is a caryopsis, a fruit containing a single seed. Barley grains are small and divided into two halves longitudinally by a crease over the entire length of the grain, commonly 6-8mm long and 3-4mm wide [17]. Some samples of barley grains are shown in Figure 1-1, where the pictures (left) without creases show the dorsal side, and another side with creases is the ventral side of the grain (right) [18].

Figure 1-1 Barley seeds. Dorsal side of hulled seeds (left); Ventral side of hulled seeds (right)[18]

The variety of barley seeds is related to the growing conditions of barley (such as geographical environment, soil nutrition, water composition, planting season, fertilization, etc.) as well as the spikelet location (e.g., two-rowed and six-rowed barley) on which the seeds are attached. These factors cause great differences in the shape and color of barley grains, the morphology of the lemma base, the small base spines, the density of the spikelet arrangement, and the horizontal state of the grains [19]. Workers can distinguish the varieties of barley through these differences. Figure 1-2 shows six types of barley seeds of different varieties.

Figure 1-2 Different Varieties of Barley Seeds

1.2 Brewing Technology

Beer, as a kind of low concentration drinking wine, is cheap and nutritious. It is deeply loved by consumers and gradually becomes a necessity in people's life. From accidental discovery to stable and mature brewing technology with rich and diverse flavors, the brewing industry has developed for thousands of years. With the evolution and development of brewing technology, the selection of raw materials has always been an important link. Malt is the soul of beer. As a raw material, it largely determines the flavor of beer during the brewing process. There are many varieties of barley, and it will take on different colors and flavors depending on where it is grown and how it is treated. Therefore, beer brewers will conduct strict preliminary tests for the quality and purity of malt, to ensure the quality of beer and reduce production risks.

There are two main methods for detecting barley varieties: physical tests and chemical tests. The commonly used method for physical tests is morphological feature analysis, which classifies varieties by visual observation of the external differences of seeds [20]. However, the size of barley seeds is too small, and the variety is slightly different, so they need to be distinguished by

professionals who have been trained for a long time. Also, the accuracy of naked eye recognition is low, and the labor cost is high. Chemical tests distinguish the type of barley through protein profiling and some specific indicators of malt, such as malt boiling color, wort viscosity, saccharification power, and β-glucan content, etc. [20]. However, chemical methods are time-consuming and require professional testing equipment, reagents, and laboratories, which are costly.

In recent years, with the development of big data, computer vision analysis has emerged in the brewing industry with its advantages of fast, efficient, and low-cost, and has gradually replaced knowledge-based detection methods [16].

1.3 Deep learning

With the rapid increase in the amount of data and the development of artificial intelligence, deep learning has been widely used in many fields. Replacing knowledge-based applications with predefined logic equations, deep learning establishes high-latitude abstract attributes by combining sample features, to discover distributed feature representations of data. Its end goal is to allow a machine that has the ability to analyze and learn like human brains, and to recognize data such as text, images, and sounds [15]. In short, a deep learning algorithm is an algorithm that automatically analyzes and obtains rules from data sets and uses the rules to predict unknown data. As one of the core problems of deep learning in the field of machine vision, the task of object detection is to find all interest targets in the image, marking their position and size, and perform classification and recognition. Since various objects have different appearances, shapes, postures, and are interfered with by factors such as illumination and occlusion during the imaging process, object detection has always been the most challenging problem in the field of machine vision [15].

Machine learning includes three methods: supervised learning, unsupervised learning, and reinforcement learning. Among them, the training set requirements for supervised learning include input (features) and output (targets). The targets in the training set are labeled by people. To solve a classification problem, supervised learning is one of the top choices. It obtains the optimal model through the existing training samples and then uses the model to map all inputs to the corresponding output, and makes simple judgments on the output to achieve the purpose of classification. Our study mainly uses the deep convolutional neural network model. We use a large amount of data as the training data set and constantly adjust the variables on the neural network to improve the accuracy of the model. We can evaluate the accuracy of training through the loss function. After the training is completed, we use the unused raw data as the test set to make predictions.

2 Artificial Neural Network

The artificial neural network is an algorithm, which is based on the abstract modeling of the human brain neuron network structure, so as to imitate the human brain for data recognition. The artificial neural network is a complex network model formed by connecting a large number of simple nodes (neurons). Each node represents an activation function, and the nodes are connected by weight values [1]. (Figure 2-1)

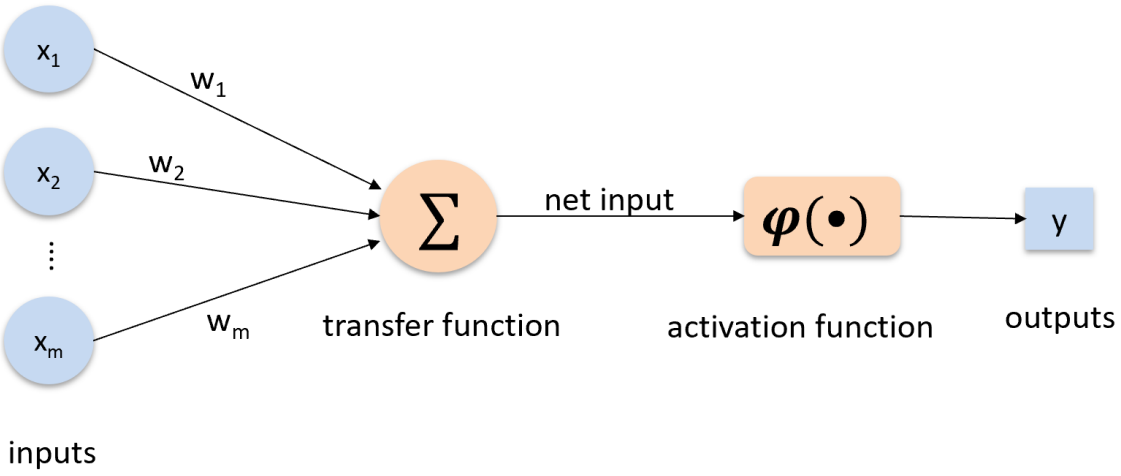

Figure 2-1 A Single Neuron

2.1 Convolutional Neural Network (CNN)

The convolutional neural network is a kind of multi-layer neural network, which is good at dealing with machine learning problems related to images, especially a lot of images. Through a series of computes, convolutional networks reduce the dimensionality of the image recognition problem with huge data volume, and finally train them. A traditional convolutional neural network is divided into four layers: convolutional layer, pooling layer, activation function layer, and fully-connected layer [2].

The convolutional layer and the pooling layer can be collectively called the feature extraction layer [3]. The convolutional layer is composed of a convolution kernel with learnable parameters. The width and length of the convolution kernel can be changed, and the depth must be consistent with the number of channels in the input layer [4]. The pooling layer controls the feature space size of the feature map, which is equivalent to a down-sampling process.

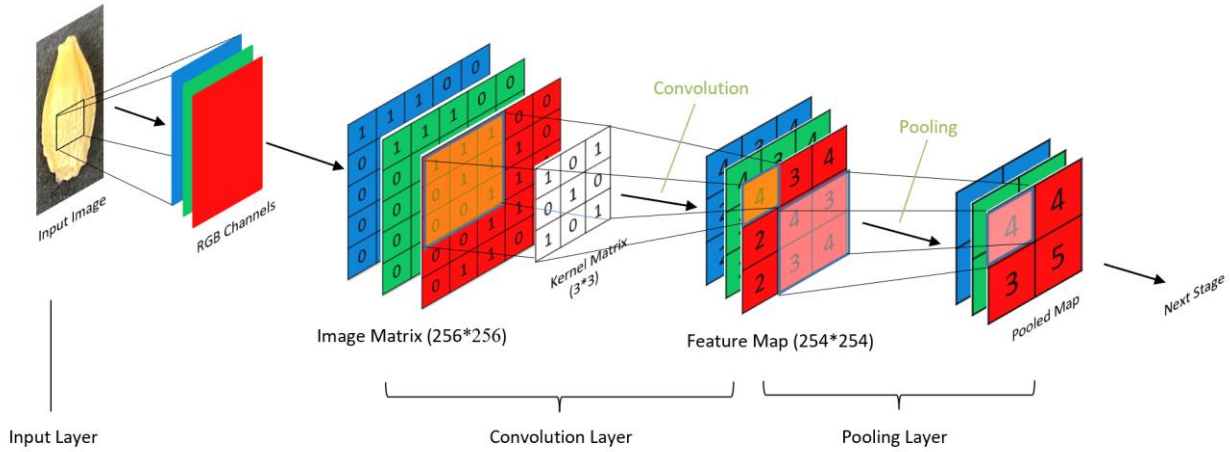

Figure 2-2 Image Feature Extraction Process

As shown in Figure 2-2, the input is an RGB image, so the input map volume is composed of three channels of red, green, and blue. Assuming that the input image size is 256×256, using 96 convolution kernels with a size of 3×3×3 to scan, and we can get 96 two-dimensional feature maps of 254×254. There are two main operations for pooling: maximum pooling and average pooling. In the figure, we do maximum pooling with 2x2 filters to obtain a pooled map of 127×127.

The fully-connected layer in CNN is different from the sliding convolution of the convolutional layer. All units in each layer of the fully connected network are completely connected to the previous layer [4]. In most cases, CNN will use multiple fully-connected layers, which can better solve nonlinear problems, and its main purpose is classification (Figure 2-3). Assuming that

there are 4 types of objects in the picture, and the 4 scores given by the model correspond to the probabilities of the 4 types. The highest score is the output prediction type.

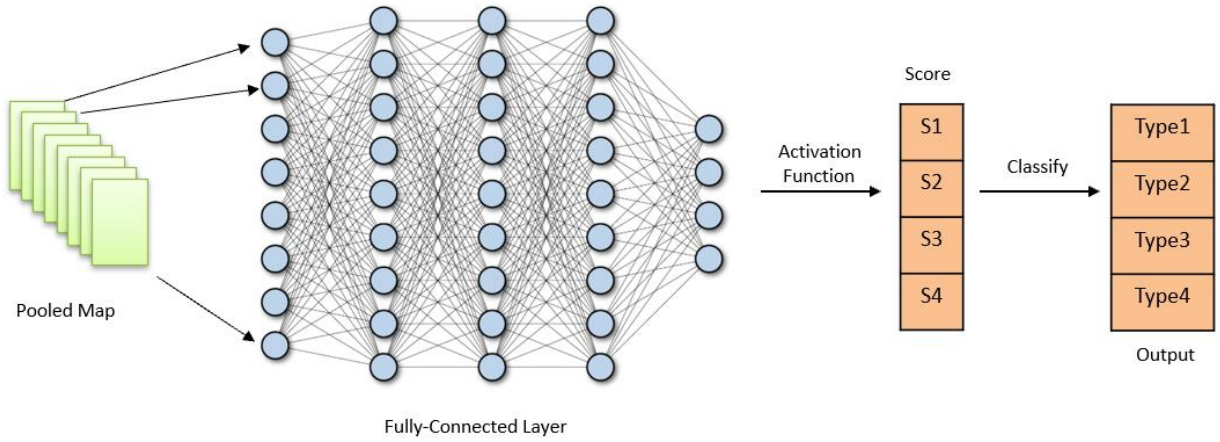

Figure 2-3 Fully-connected Layer and Classification

Theoretically speaking, the increase of the number of neurons and the number of fully-connected layers can better improve the nonlinear expression ability of the model and improve the learning ability of the model. However, too many neurons and layers may result in overfitting. The computing time is long, and the efficiency is reduced.

2.2 Object Detection

Object detection is one of the core problems in the field of computer vision. Its task is to find all objects of interest in the image and identify their category and location. Object detection algorithms based on deep learning can be mainly divided into two categories: following the traditional object detection pipeline (two-stage framework) and taking object detection as a regression or classification problem (one-stage framework) (Figure 2-4) [5]. For the two-stage framework, after feature extraction, it needs to generate a region proposal (a pre-selected box that may contain the object to be examined) at first, and then samples are classified through a convolutional neural network. Representative algorithms are SSP-Net, R-CNN series, etc. For

the one-stage framework, after feature extraction, it does not need region proposals and directly extracts features from the network to predict object classification and location. Representative algorithms include OverFeat, SSD, RetinaNet, and YOLO series [5].

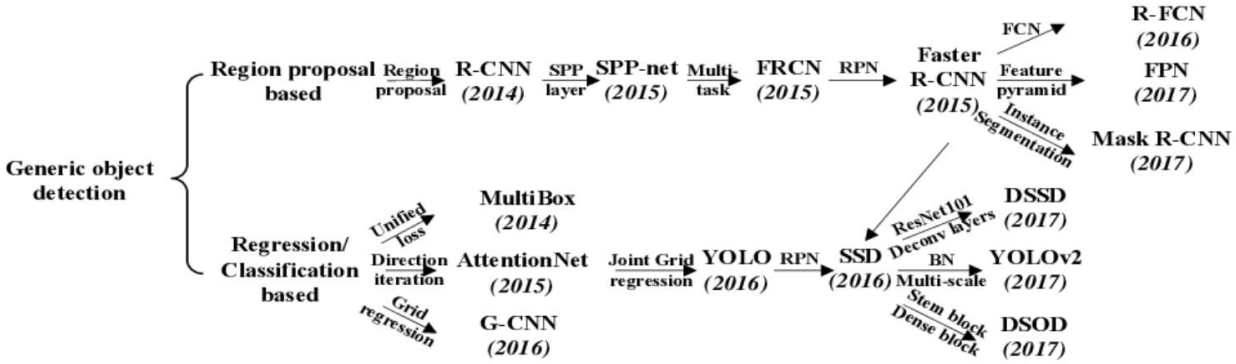

Figure 2-4 Two Types of Frameworks: Region Proposal Based & Regression/Classification Based [5]

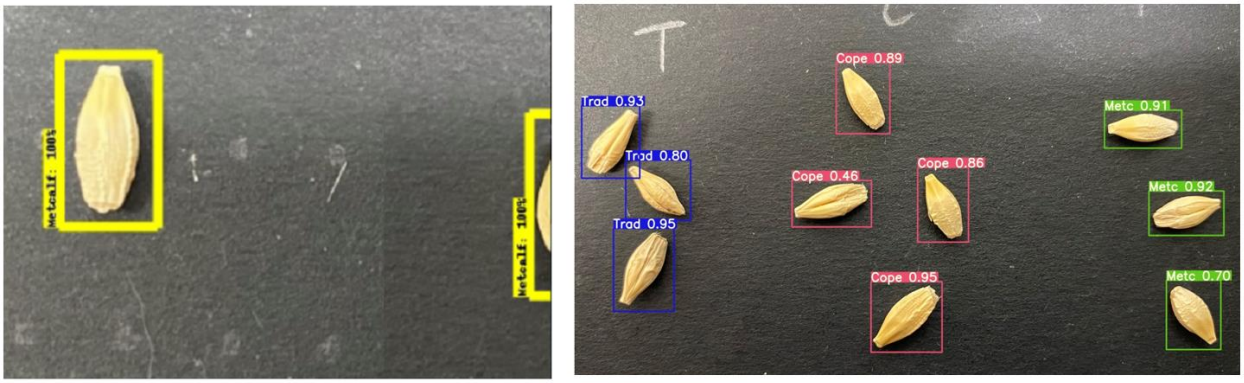

Figure 2-5 Result of Two Network Models: Faster R-CNN-Inception-v2 Network (left); YOLOv3 Network (right)

From the output results, there is no significant difference between the two frameworks, and both are used for the detection of small objects (see Figure 2-5). However, in the actual application process, Faster R-CNN [6] cannot be implemented for real-time detection. Although Faster R-CNN has a great improvement in the speed and accuracy compared with R-CNN, it still needs to acquire regional proposals, and then classify each proposal with a large amount of computation, large memory, and longer running time. Compared with the R-CNN series, the YOLO series are

not as good as the R-CNN in recall rate and the accuracy of the localization, but there is not a huge gap. Besides, the YOLOv3 [7] network model is simple, with faster computing speed, and is a real-time object detection system.

2.3 Model Evaluation

2.3.1 Precision, Recall, and Accuracy

The prediction results of the model can be divided into four types: TP, TN, FP, and FN (Table 2-1). Accuracy is a common evaluation index of model classification ability, and it is the ratio between the total number of true predictions and the total number of predictions. Generally, the higher the accuracy, the better the classifier, but we cannot only consider the accuracy. Precision is the proportion of samples classified as positive that are actually positive samples. Recall rate is the measure that means how good the model finds all the positive.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Table 2-1 Confusion Matrix

	Predicted Class		
True Class		**Positive**	**Negative**
	True	True Positive (TP)	True Negative (TN)
	False	False Positive (FP)	False Negative (FN)

2.3.2 F1 Score

We aim to achieve high precision and a high recall rate, but often the two indicators are contradictory. Thus, in many cases, we use F1 score as a tradeoff between precision and recall. A good F1 score always means a good classification model.

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

3 Methods: Model Architecture

3.1 You Only Look Once (YOLOv3)

You Only Look Once (YOLO) is a framework for object detection after RCNN, Fast RCNN, and Faster-RCNN, which was first proposed by Joseph Redmon and Ali Farhadi et al in 2015 [8]. In CVPR 2017, Joseph Redmon and Ali Farhadi came up with YOLO9000 [9], and in 2018 they published YOLOv3 [7]. The biggest highlight of YOLO is that it runs very fast and can be used in real-time systems.

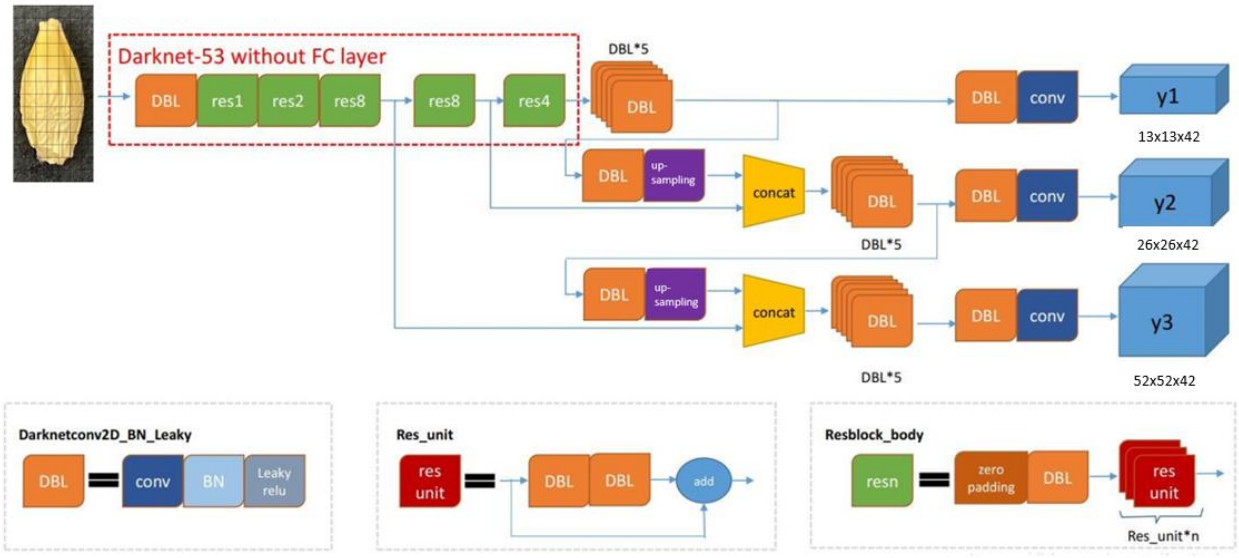

Figure 3-1 YOLOv3 Network Architecture[10]

The YOLOv3 algorithm does detection by dividing the grid. It can be divided into three parts: Darknet-53, up-sampling, and detection. For the prediction of different sizes, the method is like feature pyramid networks (FPN), using multi-scale to detect objects of different sizes. The small grid cell detects small objects, and the large grid cell detects large objects. YOLOv3 adds several convolutional layers to the basic feature extraction network, to process the combined feature map. There are three feature maps of different scales, as shown in Figure 3-1 (y1, y2, and y3),

and using up-sampling to ensure that the two tensors have the same scales when they are connected [10]. Assuming the input size is 416×416, the output feature maps are 13×13×42, 26×26×42, and 52×52×42. 42 = 3×(5+9), where 3 means each grid cell has three anchor boxes, 5 means there are five parameters of each bounding box (the x-coordinate and y-coordinate of the center of the object, height and width of the box, and confidence), and 9 means our project has 9 types. Multi-scale is used to detect objects of different sizes.

Different from traditional CNN, YOLOv3 does not have a pooling layer and fully-connected layer. It adds a Batch Normalization (BN) layer and LeakyReLU after each convolutional layer. The BN layer assists in the regularization of deep neural networks, which can improve the convergence rate of the model and prevent the model overfitting [9]. Moreover, unlike Faster RCNN using the hand-picked anchor boxes, YOLOv3 uses k-means clustering to generate bounding boxes that are easier to learn. Unlike the predicted offset in the Faster RCNN, YOLOv3 follows the same approach as the YOLOv2 to directly predict the relative position and uses more bounding box priors (nine anchors), which makes the network more stable. Another improvement for Yolov3 is that it uses the Darknet-53 network to the previous Darknet-19, further deepening the network.

3.2 Darknet-53

Different from YOLO using Googlenet, YOLOv2 uses a new Darknet-19 as a backbone network. Darknet-19 includes 19 convolutional layers and 5 max-pooling layers (Figure 3-2 left)[9]. Because the layer loses fine-grained features from down-sampling, YOLOv2 does not detect small objects very well. To solve this problem, YOLOv3 combined Darknet-19 and residuals network (ResNet) to design a new network called Darknet-53.

Compared with Darknet-19, Darknet-53 has a total of 106 layers (Darknet-19 has a total of 30 layers), which contains 53 convolutional layers. Darknet-53 cancels the maximum pooling layer in Darknet-19, and it changes the size by increasing the stride of the convolution kernel.

Darknet-19

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

Darknet-53

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 3-2 Darknet-19 [9] vs. Darknet-53 [7]

As shown on the right side of Figure 3-2, at the beginning of the network, it has a convolution kernel (3×3) with 32 filters, then there are 5 sets of repeated residual blocks (resblock). Each resblock consists of a single convolutional layer and a set of repetitive convolutional layers. The repetitive convolutional layers are repeated 1, 2, 8, 8, and 4 times respectively. In each repeated convolution layer, it does a 1×1 convolution operation first and then does a 3×3 convolution operation. The number of filters is first reduced by half and then restored. Each group's first

single resblock convolution operation is a convolution operation with a stride of two, so the entire YOLOv3 network has a total of five dimensionality reductions. That is, the feature map will be reduced to 1/32 of the original size.

Darknet-53 has larger than Darknet-19 and has a slightly slower computing speed [7], and YOLOv3 is a bit larger than YOLOv2. However, YOLOv3 is more accurate and better at detecting small objects than YOLOv2. If you are looking for computing speed, tiny-darknet [11] is a good choice to replace Darknet-53.

3.3 Loss Function

The loss function is the key to determining the learning quality of the neural network, and its role is to describe the gap between the predicted value of the model and the true value. The smaller the loss function, the better the model performance. In an ideal situation, if the prediction matches the true value perfectly, the loss function is equal to 0. The YOLO network uses a simple sum-square error.

$$
\begin{aligned}
&\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2
\end{aligned}
$$

Figure 3-3 The Loss Function of YOLOv1[8]

In the papers of the YOLO series, only YOLOv1 gave the equation of loss function clearly, as shown in Figure 3-3, in that *S* is the number of grid cells and *B* is the number of bounding boxes. 1_{ij}^{obj} is 1 when box *j* and grid cell *i* are matched together and 0 otherwise; 1_{ij}^{noobj} is 1 when box *j* and grid cell *i* are not matched together; 1_{i}^{obj} is 1 when grid cell *i* has an object present. In this loss function, the first two items are bounding box coordinate regression, the third and fourth items are bounding box score prediction (confidence error), and the last item is class score prediction (classification error) [12].

$$
\begin{aligned}
Loss = & \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} [(\sigma(t_x)_i^j - \sigma(\hat{t}_x)_i^j)^2 + (\sigma(t_y)_i^j - \sigma(\hat{t}_y)_i^j)^2] + \\
& \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{I}_{ij}^{obj} [({t_w}_i^j - \hat{t}_{w}{}_i^j)^2 + ({t_h}_i^j - \hat{t}_{h}{}_i^j)^2] + \\
& \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{G}_{ij} (C_i^j - \widehat{C}_i^j)^2 + \\
& \sum_{i=0}^{S^2} \sum_{j=0}^{B} \sum_{c \in classes} \mathbb{I}_{ij}^{obj} (p_i^j(c) - \hat{p}_i^j(c))^2
\end{aligned}
$$

Figure 3-4 The Loss Function of YOLOv3 [13]

Compared with the loss function of YOLOv1, YOLOv3 cancels the two parameters λ_{coord} and λ_{noobj} (Figure 3-4). Moreover, in YOLOv3, Redmon puts confidence and classification prediction in each bounding box, that is, for each bounding box has a pair of confidence and classification prediction, while YOLOv1 is a classification prediction shared by all bounding boxes [7]. In the loss function, the number of grid cells is not fixed. The convolutional layer has no limits on the size of the input. The model randomly changes the size of the input every training period before moving to the next period [9]. This allows the model to perform well on different sizes of

images. As shown in Figure 3-4, $\widehat{C}_i^j = 1$, if the bounding box of the grid cell is responsible for predicting an object. However, when a bounding box is not responsible for the prediction of the ground truth box in the corresponding grid cell, but the IOU with the ground truth box is greater than the set threshold, $G_{ij} = 0$ (all errors of this bounding box are ignored).

3.4 Intersection Over Union (IoU)

Intersection over union is a matrix to measure the ratio between the predicted bounding boxes and the ground-truth bounding boxes (Figure 3-5). Normally, we predefine a threshold value for IoU = 0.5.

If IoU > 0.5, we set it as True Positive,

If IoU <= 0.5, we set it as False positive,

If IoU > 0.5 and the object has wrongly classified, we set it as False Negative.

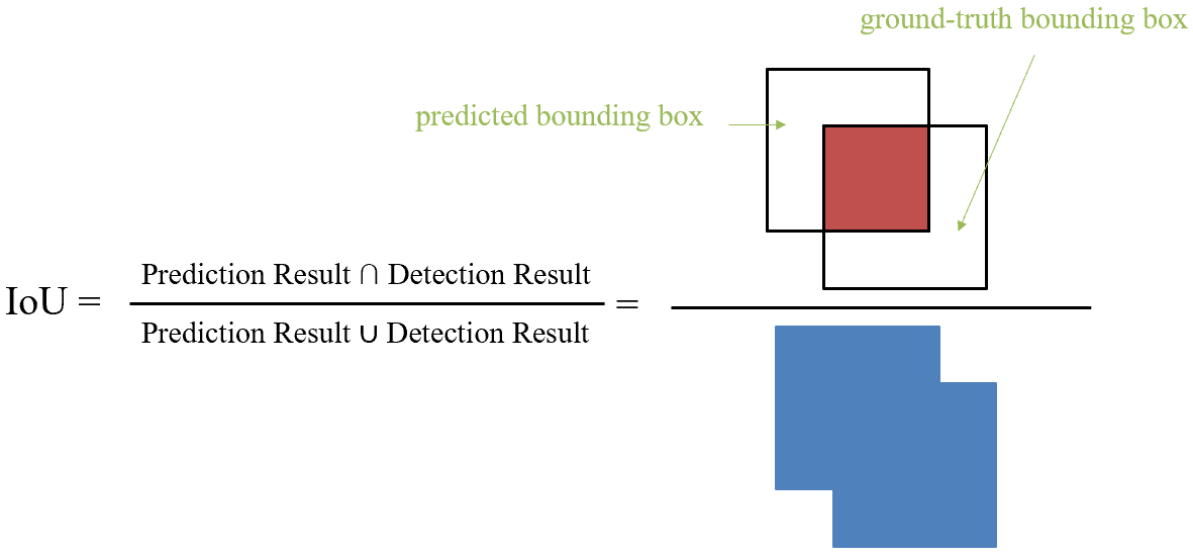

Figure 3-5 Intersection Over Union

3.5 Mean Average Precision (mAP)

Average precision (AP) is a popular metric to measure the accuracy of object detectors. Average precision computes the average precision value for recall value over 0 to 1. The calculation formula of AP is as follows [21],

$$AP = \sum_{r=0}^{1} (r_{n+1} - r_n) P_{interp}(r_{n+1})$$

$P_{interp}(r)$ is interpolated precision, representing the maximum value of *precision* corresponding to all recall rates r' greater than the specified recall rate r. In general, AP is for a particular category in the dataset, and mAP is for the entire dataset. The mAP is between 0 and 1, the bigger, the better.

4 Methods: Data Preparation

4.1 Data Collection

A clear and true data set will ensure the authenticity and stability of the training model. The barley seeds of this project were all provided by our industrial collaborator, and the samples were from barley farms in the United States during the 2018 and 2019 planting seasons. There are nine varieties of sample barley, three types of six-row barley (Celebration, Legacy, and Tradition), and six types of two-row barley (Copeland, Hockett, Scarlett, Metcalfe, Synergy, and Expedition), as shown in Figure 4-1.

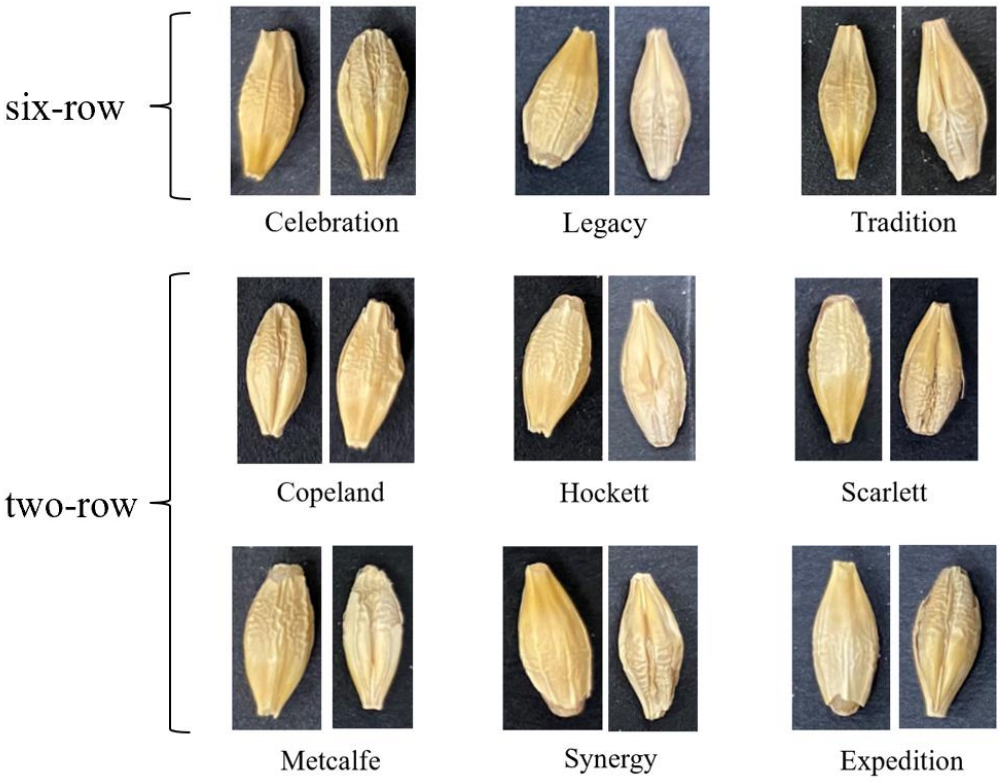

Figure 4-1 Nine Barley Varieties

All the data was captured manually by project members using the iPhone 11 or iPhone 11 Pro. We collect various types of images as raw data, including a single seed, multiple seeds of the

same variety, and a mixture of multiple seeds of different varieties. The background of the data image is black, and all seeds are arranged in random directions, and the seeds are both on the ventral and dorsal sides (Figure 4-2). To ensure the efficiency and accuracy of the experiment, the outer skin of the selected sample seeds is basically complete. Totally, there are more than 3700 images of data and nearly ten thousand seeds.

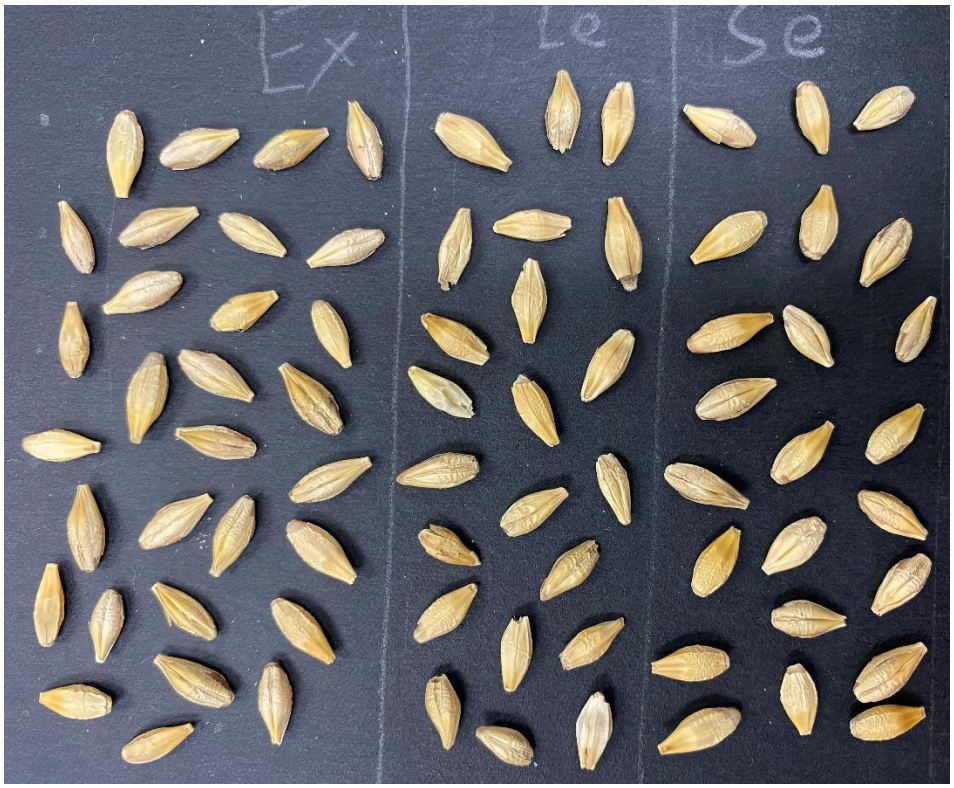

Figure 4-2 Raw Data of Mixed Varieties

4.2 Data Processing

All collected images have been manually labeled to train and evaluate our model. The marking tool we use is a free image annotation tool on the Internet, named ImgLab [14] for data labeling. The annotations are saved in Pascal VOC format. Each image will generate a corresponding .xml file, which includes the name of the image labeling type and the two-dimensional coordinates of the bounding box.

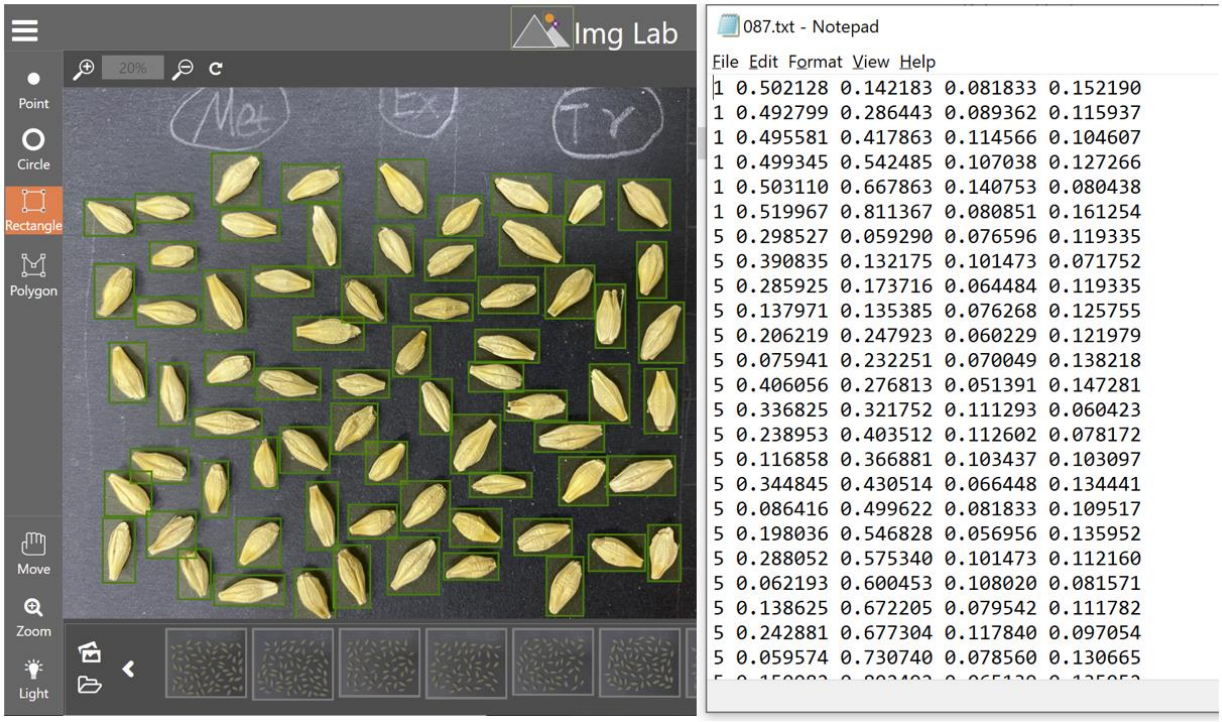

Figure 4-3 Image Label

Figure 4-3 shows the process of data labeling and the final form of data storage, where the left side is the software interface of ImgLab, and the right side is the input format corresponding to the image annotation during model training (the bounding box data has been normalized). The annotation file is divided into 5 columns, the first column is the number corresponding to the seed variety name, and the last four columns are the coordinate values of the upper left (x_{min}, y_{min}) and lower right (x_{max}, y_{max}) corners of the bounding box. One text file corresponds to an image, and each row of data represents the location of a seed in the image.

5 Experimental Results and Discussion

5.1 Training platform

The network models used in the experiment are all written in the Python programming language, using the Pytorch deep learning framework. The models are trained on an Nvidia GeForce RTX 2080Ti GPU platform with 12GB of memory.

5.2 Network Details

We store the configuration information of the network model in yolov3.cfg. This model is trained for 1000 epochs with 4 batch sizes, and the learning rate is 0.0001. All data is divided into two data sets: training set, testing set. The allocation ratio is 0.8:0.2, of which 80% (2903 images) of the data is used as the training set to train the model, 20% (727 images) is the validation set to test the performance of the model, and an additional 122 raw images as a test set. The training time of the model is roughly 31 hours.

5.3 Result

The data set for this experiment includes single barley images for 9 categories, and mix categories barley images (Copeland, Metcalf, Tradition). The process and results of this model training are shown in Table 5-1 and Figure 5-1. With an IoU of 0.5 and non-maximum suppression of 1.00, we got the mAP (mean Average Precision) value of 0.97. There were 2903 images in the training data set and 727 images in the test data set. The model recognition (localization and classification) precision was 91.2%, the recall rate was 95.9%, and the F1 score was 93.3%.

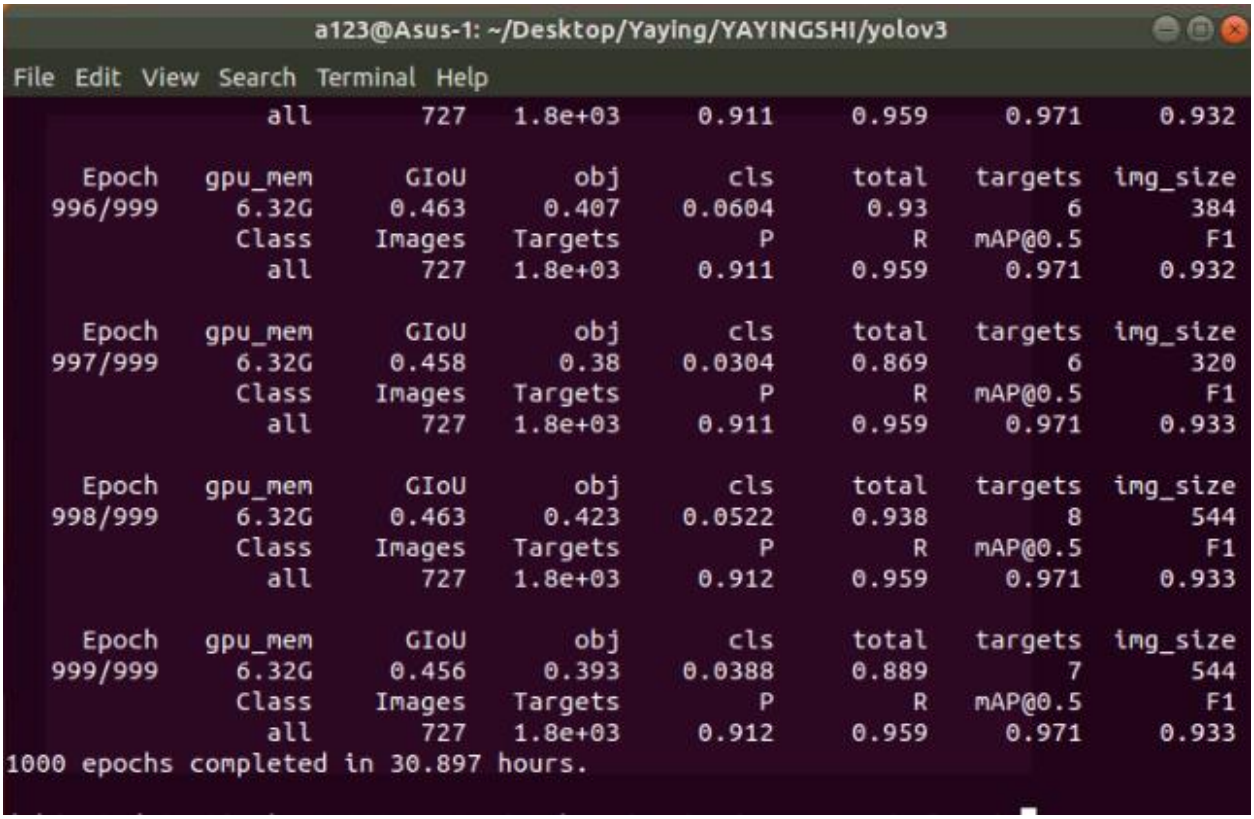

Figure 5-1 Training Process

Table 5-1 Training Result

	# Training Images	**# Test Images**	**Targets**	**mAP**
YOLOv3	2903	727	1.8e+03	0.971
	Precision	**Recall**	**F1 Score**	**Total Time**
	0.912	0.959	0.933	30.897 h

In the detection process, we detected a total of 122 images, including single seed images and mixed-type seed images. Part of the image detection results are shown in Figure 5-2. The output content includes the size of the image, the type and number of seeds in the image, and the time taken for detection. The detection time of all images waved between 0.011s to 0.013s.

```
File  Edit  View  Search  Terminal  Help
image 71/122 data/rest/images/3631.jpg: 512x384 45 Metcs, Done. (0.012s)
image 72/122 data/rest/images/3632.jpg: 512x384 10 Copes, 37 Metcs, Done. (0.013s)
image 73/122 data/rest/images/3641.jpg: 512x384 45 Metcs, Done. (0.012s)
image 74/122 data/rest/images/3642.jpg: 512x384 45 Metcs, Done. (0.012s)
image 75/122 data/rest/images/3651.jpg: 512x384 45 Copes, Done. (0.012s)
image 76/122 data/rest/images/3652.jpg: 512x384 45 Metcs, Done. (0.012s)
image 77/122 data/rest/images/3661.jpg: 512x384 45 Metcs, Done. (0.012s)
image 78/122 data/rest/images/3662.jpg: 512x384 45 Metcs, Done. (0.012s)
image 79/122 data/rest/images/3671.jpg: 512x384 45 Copes, Done. (0.013s)
image 80/122 data/rest/images/3672.jpg: 512x384 44 Metcs, Done. (0.012s)
image 81/122 data/rest/images/3681.jpg: 512x384 45 Metcs, Done. (0.012s)
image 82/122 data/rest/images/3682.jpg: 512x384 15 Copes, 31 Metcs, Done. (0.012s)
image 83/122 data/rest/images/3691.jpg: 512x384 45 Metcs, Done. (0.011s)
image 84/122 data/rest/images/3692.jpg: 512x384 45 Metcs, Done. (0.012s)
image 85/122 data/rest/images/3701.jpg: 384x512 5 Copes, 6 Metcs, 7 Trads, Done. (0.012s)
image 86/122 data/rest/images/3702.jpg: 448x512 2 Copes, 60 Metcs, Done. (0.013s)
image 87/122 data/rest/images/3711.jpg: 320x512 4 Copes, 3 Metcs, 3 Trads, Done. (0.011s)
image 88/122 data/rest/images/3712.jpg: 320x512 5 Copes, 6 Metcs, 5 Trads, Done. (0.011s)
image 89/122 data/rest/images/3721.jpg: 448x512 41 Copes, 38 Metcs, Done. (0.013s)
image 90/122 data/rest/images/3722.jpg: 512x512 39 Copes, 44 Metcs, Done. (0.013s)
image 91/122 data/rest/images/3731.jpg: 512x512 39 Copes, 18 Metcs, Done. (0.013s)
image 92/122 data/rest/images/3732.jpg: 448x512 27 Copes, 37 Metcs, Done. (0.013s)
image 93/122 data/rest/images/3741.jpg: 384x512 8 Metcs, 8 Trads, Done. (0.012s)
image 94/122 data/rest/images/3742.jpg: 448x512 43 Copes, 49 Metcs, Done. (0.013s)
image 95/122 data/rest/images/3751.jpg: 320x512 7 Copes, 6 Metcs, 7 Trads, Done. (0.011s)
image 96/122 data/rest/images/3752.jpg: 448x512 32 Copes, 31 Metcs, Done. (0.013s)
image 97/122 data/rest/images/4.jpg: 512x320 1 Celes, Done. (0.011s)
image 98/122 data/rest/images/401.jpg: 512x320 1 Copes, Done. (0.011s)
image 99/122 data/rest/images/402.jpg: 512x320 1 Copes, Done. (0.012s)
image 100/122 data/rest/images/403.jpg: 512x320 1 Copes, Done. (0.012s)
image 101/122 data/rest/images/404.jpg: 512x320 1 Copes, Done. (0.011s)
image 102/122 data/rest/images/405.jpg: 512x320 1 Copes, Done. (0.011s)
image 103/122 data/rest/images/406.jpg: 512x320 1 Copes, Done. (0.011s)
```

Figure 5-2 Image Detection Process

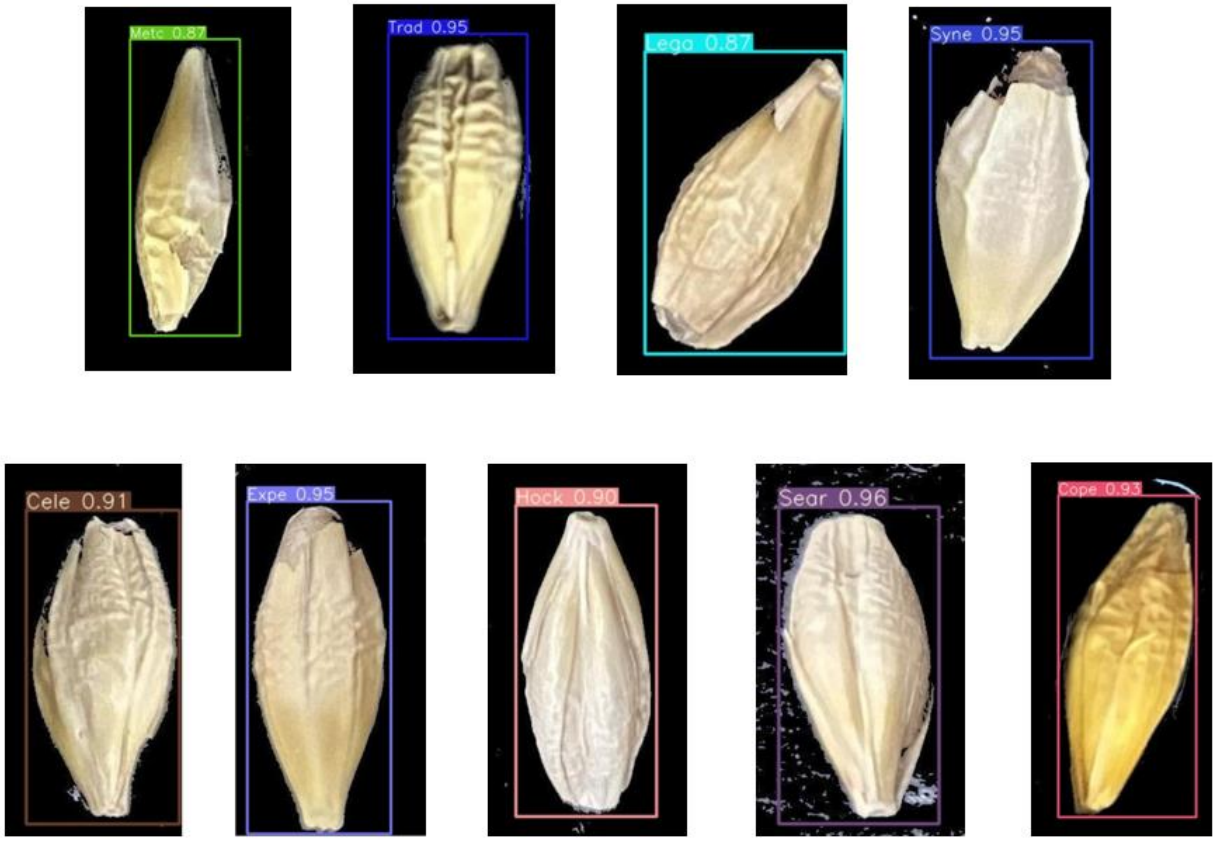

Figure 5-3 Single Seed Test Results

Figure 5-4 Good Results for Mixed types

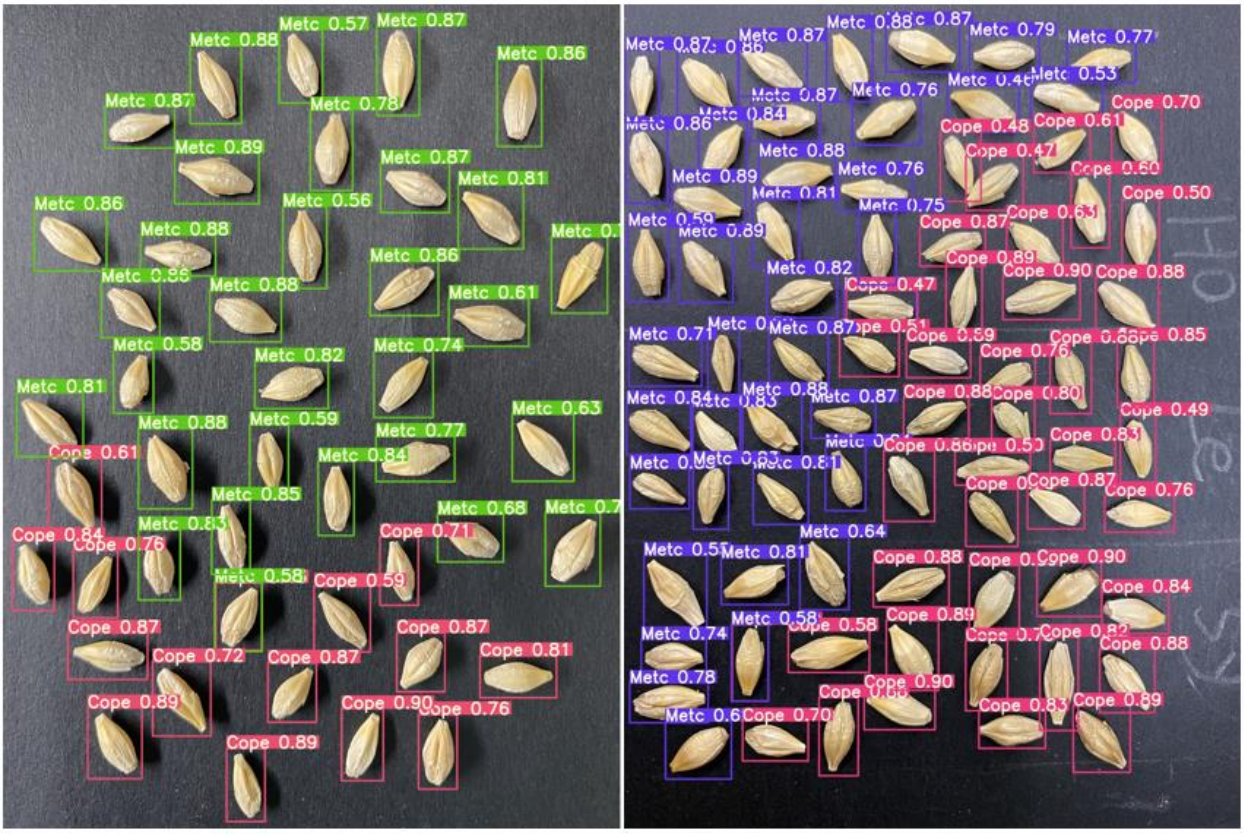

Figure 5-5 Bad Results for Multiple Seeds

Figure 5-3 shows the test results of a single seed. The detection results of 9 types of single seeds were relatively accurate, and there was no undetected or incorrect classification. However, tests on multiple seeds have yielded mixed results. The good results are shown in Figure 5-4 and Figure 2-5 (right side), where a single type of multiple seeds or multiple types of mixed images can be clearly and accurately detected. The bad result is shown in Figure 5-5, where the seed type of the left image is all Metcalf barley, but there are some wrong recognition barleys as Copeland. There are three types of seeds on the right side in Figure 5-5, namely Hockett, Legacy, and Synergy. To verify the effectiveness of the model, these three types of seeds did not include multiple seed images in the training set, only a single image, and all the detections were wrong. So, it proves that when there is a big difference between the training set and the test set, even if the model trained well, the effect is not good when detecting multiple seeds.

Furthermore, there was also a case in the validation set the seed of an image has not been detected (Figure 5-6). In the verification set of 122 images (about 1415 seeds), only one seed was not detected, so the detection accuracy of our model was above 99%.

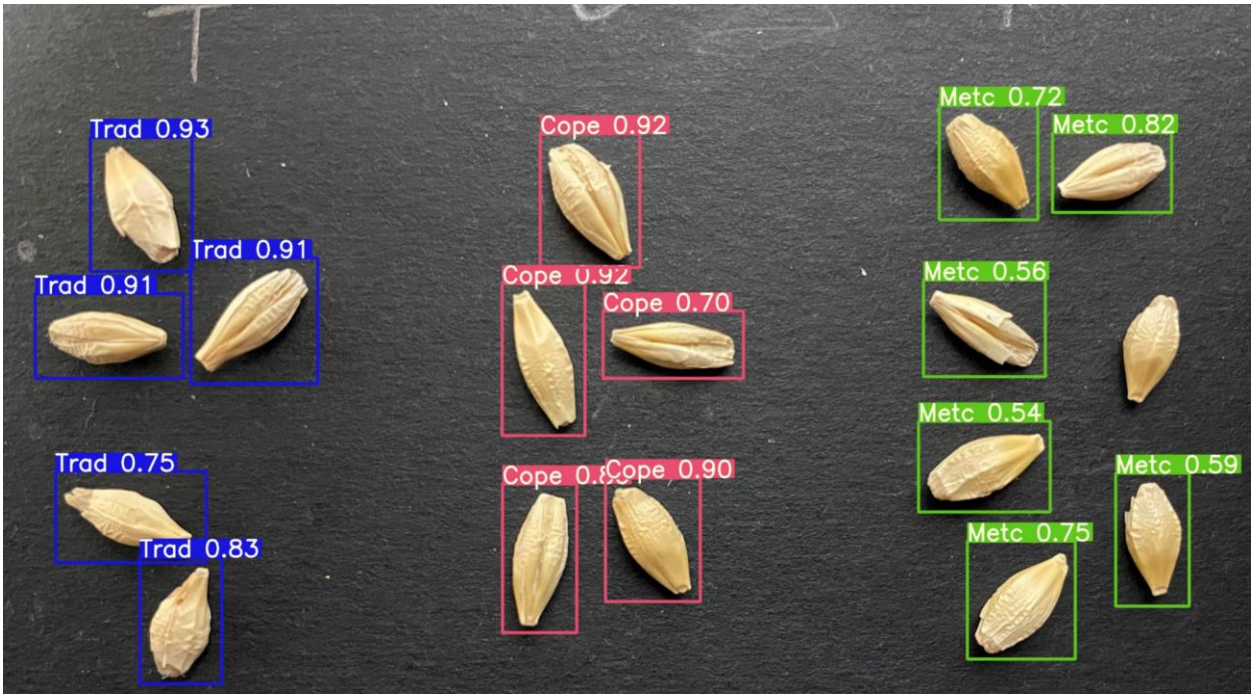

Figure 5-6 Seed Undetected

5.4 Discussion

There are a large number of single-seed images in the training set, so the recognition effect for similar single-seed is good. The training set contains single seeds of Hockett, Legacy, and Synergy types, but there are serious errors in detecting these three types of multi-seeds images. Therefore, the features abstracted by single-seed training cannot be suitable for multi-seeds mixed types detection. And the more the number of seeds in the image, the lower the resolution of a single seed, and the more difficult it is to extract the features of the seeds. If the number of seeds in the image is appropriately reduced, the effect may be better. Besides that, the bad results need more methods to achieve high precision accuracies, like YOLOv4, YOLOv5, or additional adjustable datasets.

6 Conclusion and Future Work

The end goal of this project is to develop an iPhone App to automatically detect and classify barley seeds. So far, our work proves that the deep learning-based method was sufficient to replace the traditional knowledge-based method to seed detection because of two keys factors: accuracy and efficiency. The best recognition precision of the model was 91.2%, the mAP value we obtained was 97.1%, and the recall rate was 95.9%. In addition, the calculation speed of the model is fast, and the computing power is less than the R-CNN series models. Compared with tedious and time-consuming manual detection, our model only takes 0.012 seconds to detect an image containing about 50 barley seeds. AI algorithm efficiency is much higher than manual detection, and it can avoid errors caused by human factors. Based on the project observation, the YOLOv3 network was effective in barley seed detection. In subsequent experiments, we will find a way to further improve the detection accuracy.

For the previous study, we used InceptionV3 models to train the single type of barley seeds for classification experiments on Matlab. The accuracy of nine varieties of barley classification is 95.7%. Therefore, we will try to separate the detection and classification. YOLOv3 performs detection and classification at the same time. All parameters are mixed, and there are no special parameters responsible for detection or classification. So, it increases the difficulty of learning and reducing accuracy. The detection accuracy of YOLOv3 is more than 99%. Thus, we can use YOLOv3 to detect and cut the detected objects into individual images and classified them in the Inceptionv3 model.

Focusing on the bad results above, we need to add more multi-seed data and ensuring the number of each type is relatively balanced. Since the seeds in the marginal area of the image and the seeds with unobvious skin creases are prone to over-explode during sampling, we will preprocess

the sample data, such as enhancing the sharpness and reducing the brightness. To improve the training time, we will try to use a more powerful GPU and tiny-darknet to replace Darknet-53 and evaluate model performance.

To facilitate the use of the developed AI algorithm, this project will develop a flexible and portable mobile application. The trained model is stored in the cloud GPU, and will be used to detect and classify the barley seeds in the images that the App uploads to the cloud GPU in real-time. After GPU analysis and detection, the result will be returned to the mobile client in time. All computing will be done in the cloud, so the App does not take up much memory and computing power on the phone. Because all the data needed for this project was taken on a mobile phone, there is no need to worry about the pixels of the smartphone that may have too much influence on the results when the App is used.

References

[1] Hassoun, M. H. *Fundamentals of Artificial Neural Networks.* IEEE TRANSACTIONS ON INFORMATION THEORY, vol.42(4), pp.1322 (1996, July).

[2] Albawi, S.; Mohammed, T. A.; Al, Z. S., *Understanding of a convolutional neural network.* International Conference on Engineering and Technology (ICET). (Jan.6, 2017)

[3] Kang, X., Song, B., & Sun, F. (2019). *A Deep Similarity Metric Method Based on Incomplete Data for Traffic Anomaly Detection in IoT.* Applied Sciences, vol.9(1), pp.135.

[4] Gonzalez R C. (2018). *Deep Convolutional Neural Networks* [Lecture Notes][J]. IEEE Signal Processing Magazine, vol.35(6), pp.79-87.

[5] Zhao, Z.-Q., Zheng, P., Xu, S.-T., & Wu, X. (2019). *Object Detection with Deep Learning: A Review.* IEEE Transactions on Neural Networks and Learning Systems, vol.30(11), 3212–3232.

[6] Ren, S., He, K., Girshick, R., & Sun, J. (2017). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.39(6), 1137–1149.

[7] Redmon, J., & Farhadi, A. (2018, April 8). *YOLOv3: An Incremental Improvement*. ArXiv.Org. https://arxiv.org/abs/1804.02767

[8] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015, June 8). *You Only Look Once: Unified, Real-Time Object Detection*. ArXiv.Org. https://arxiv.org/abs/1506.02640v1

[9] Redmon, J., & Farhadi, A. (2017, July 1). *YOLO9000: Better, Faster, Stronger*. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

[10] Muzhan, (Sep.12, 2018) *YOLOv3 of the YOLO Series [Deeply Analysis]*. CSDN.com. From https://blog.csdn.net/leviopku/article/details/82660381.

[11] Redmon, J. *Tiny Darknet*. https://pjreddie.com/darknet/tiny-darknet/

[12] Amaia Salvador. (2017, April 28). *Object Detection (D2L5 Insight@DCU Machine Learning Workshop 2017)*. Github. Form https://www.slideshare.net/xavigiro/object-detection-d2l5-insightdcu-machine-learning-workshop-2017

[13] *Some Details about YOLOv3*. (Jun. 2019). Jianshu.com, Form https://www.baidu.com/link?url=Pxkdhv_wfnh7WQwjZPHWFvF2vkr-lVDaJ3pG_TUaTilia4cDUvAokMXsTRO1qKv6&wd=&eqid=af613303000156f9000000066062aab3

[14] Gupta, A, et.al. *ImgLab*. GitHub. https://github.com/NaturalIntelligence/imglab

[15] *Object detection: Comparative analysis of Faster-RCNN and YOLO V3 models*. (August 11, 2020). Form https://blog.csdn.net/weixin_43483381/article/details/107944903.

[16] Li, Y.; Maurice, M. (2018). *Malting in 2038.* World Brewing Summit, San Diego, CA, USA.

[17] Ulla Holopainen-Mantila, U. (2015). *Composition and structure of barley (Hordeum vulgare L.) grain in relation to end uses: Dissertation*. VTT's Research Information Portal.

[18] Wabila, C., Neumann, K., Kilian, B., Radchuk, V., & Graner, A. (2019a). *A tiered approach to genome-wide association analysis for the adherence of hulls to the caryopsis of barley seeds reveals footprints of selection.* BMC Plant Biology, 19(95).

[19] Briggs, D. E., Brookes, P. A., Stevens, R., & Boulton, C. A. (2004). *Brewing: Science and Practice (Woodhead Publishing Series in Food Science, Technology and Nutrition)* (1st ed.). Woodhead Publishing.

[20] Zhou, Y. (2013). *The Quality of Malt and Its Influence on Beer Brewing*. BEER TECH. (China), Vol.10, pp.49.

[21] Hui, J. (2020, February 7). *mAP (mean Average Precision) for Object Detection - Jonathan Hui*. Medium. Form https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173