SeedScope

Seed type detection with deep learning applications

Muhammed Berkay Yılmaz 21328935676

Selin Akalın 40444602194

Alperen Bayarslan 40840923640

***Abstract*— SeedScope is an image processing project, focusing on automated seed sorting for efficient plant breeding. Utilizing deep learning, the system distinguishes seed shapes and colors, providing detailed information to users. The experiment involves capturing seed images, researching characteristics, labeling, and employing data augmentation. The You Only Look Once (YOLO) algorithm is used for training, aiming to prevent overlearning. Future research extends to GMO product identification, contributing to genetic research.**

Keywords: Seed Sorting, GMO Identification.

I. INTRODUCTION

SeedScope, addresses the need for automated seed sorting in agriculture and plant breeding. The conventional manual process of examining individual seeds for diversity assessment is time-intensive, prompting the development of an innovative system leveraging image processing and deep learning. The second part tries to describe the problem in real life, the third part gives information about solving methods and the last part has brief conclusion.

II. PROBLEM IDENTIFICATION

It is critical to correctly identify and differentiate between kinds of seeds in plant breeding and agriculture sectors. This exercise, however, requires examination of one seed at a time in order to estimate its diversity. Thus, it is imperative to design an automated seed sorting system equipped with image processing technology.

This system aims precisely sorting each one seed by taking into consideration its form, color, mass, and distinctive attributes relating to various plants seeds.

III. PROBLEM SOLVING METHODS AND EXPERIMENTS

The YOLO (You Only Look Once) algorithm is a real-time object detection system that divides an image into a grid and predicts bounding boxes and class probabilities directly within each grid cell. Unlike traditional object detection methods, YOLO processes the entire image in a single forward pass, enabling faster and more efficient real-time detection of multiple objects within the scene.

YOLOv5

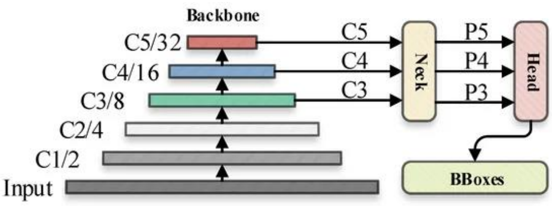

Figure 1 The default inference flowchart of YOLOv5

The image was processed through a input layer (input) and sent to the backbone for feature extraction.

Backbone

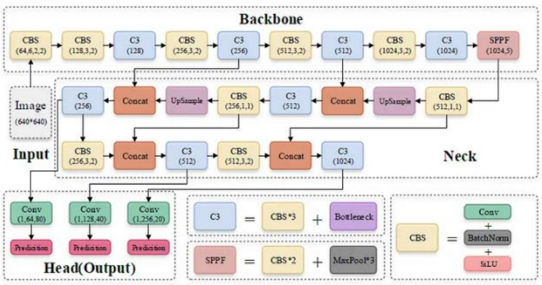

Figure 2 YOLOv5: Model Architecture

CSPDarknet53, the backbone of YOLOv5, obtains feature maps of varying sizes and fuses these features through the feature fusion network, also known as the neck. This results in the creation of three feature maps, P3, P4, and P5, each with dimensions of 80×80, 40×40, and 20×20 respectively. These maps are used to detect small, medium, and large objects in the image.

The main structure of CSPDarknet53 consists of stacking multiple CBS (Conv + BatchNorm + SiLU) modules and C3 modules, and finally connecting one SPPF module. The CBS module assists the C3 module in feature extraction, while the SPPF module enhances the feature expression capability of the backbone.

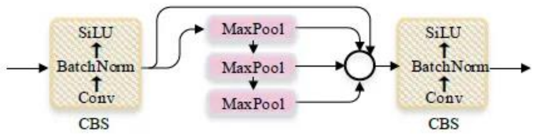

Figure 3 Structure of SPPF

SPPF avoids the repeated operation of SPP as in SPPNet by max pooling the previously max pooled features. This significantly improves the running speed of the module.

Neck

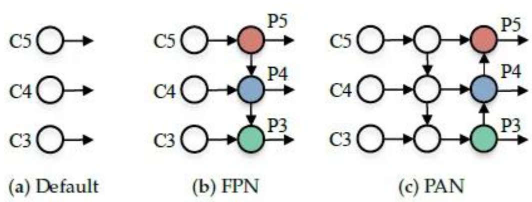

Figure 4 (a) Without Feature Fusion, (b) FPN and (c) PAN.

YOLOv5 employs the techniques of FPN and PAN. The fundamental concept of FPN is to up-sample the output feature map (C3, C4, and C5) generated from multiple convolution down-sampling operations from the feature extraction network. This process creates a series of new feature maps (P3, P4, and P5) for detecting targets of different scales.

Head

After the three feature maps are sent to the prediction head, calculations for confidence and bounding-box regression are executed for each pixel in the feature map using the preset prior anchor. This results in a multi-dimensional array (BBoxes) that includes information about the object class, class confidence, box coordinates, width, and height.

By setting the corresponding thresholds (confthreshold, objthreshold), the array is filtered to remove unnecessary information. A non-maximum suppression (NMS) process is then performed to output the final detection information.

The coordinate value of the upper left corner of the feature map is set to (0, 0). The unadjusted coordinates of the predicted center point are represented by rx and ry. The adjusted prediction box information is represented by gx, gy, gw, gh. The prior anchor information is represented by pw and ph. The offsets calculated by the model are represented by sx and sy.

The process involves adjusting the center coordinate and size of the preset prior anchor to match the center coordinate and size of the final prediction box.

IV.CONCLUSION

This project was implemented to perform seed identification and classification with YOLOV5, an object recognition algorithm. The model is trained on a dataset containing images of 4 types of seed with 50 photos for each. The images were taken by ourselves to produce a clearer and more consistent dataset. For training the model with larger dataset,image augmentation is used to increase the number of photos. In the future, we are planning to develop a mobile app with different variety of seeds and with a larger data set. This mobile application can be used to enlighten people, from young to old, who want to learn new information about agriculture. In this way, we aim to reach wider range of community and increase people interested in agriculture. In addition, since it is often difficult for hobby farmers to reach experienced farmers, we think this project is an effective learning tool.

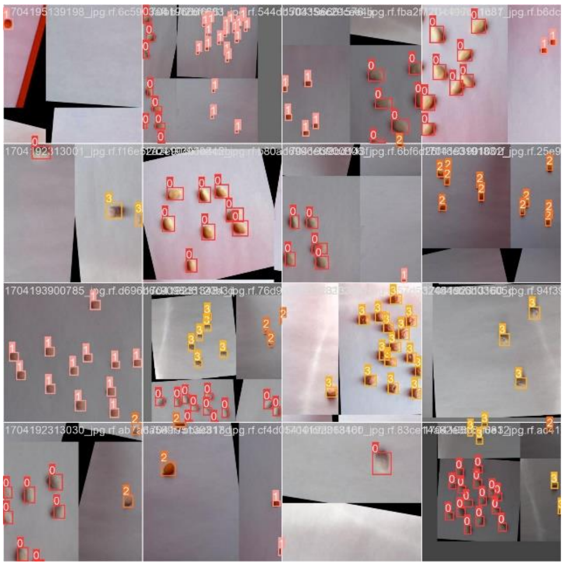

Figure 5 Annotated images

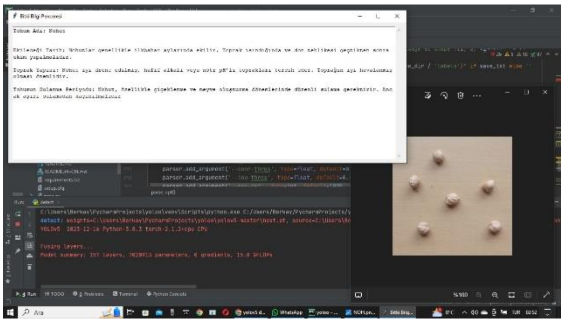

Figure 6 Result of project

Algorithm: Determining the diversity of seeds

Input: Image of the labeled seeds

Output: Type of seed

---

1. Defining Arguments
2. Creating Recording Directories
3. Preparation of the Model and Data Loader
4. Inference (Detection) Process
5. Visualization and Processing of Results
6. Performance Timing and Printing Results
7. Optimization

Table 1: Algorithm of our Python code

V. COMPARISON

In this part of the report, similar result from an article is compared with training response of this project.

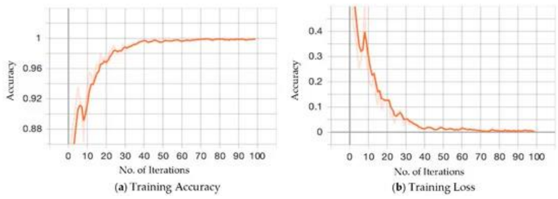

Figure 7 Training accuracy and loss of the CNN model.

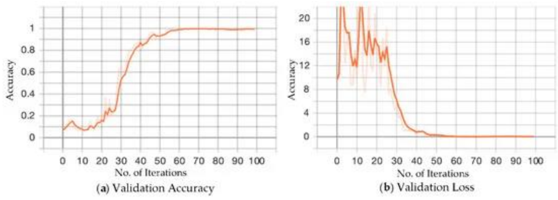

Figure 8 Validation accuracy and loss of the CNN model

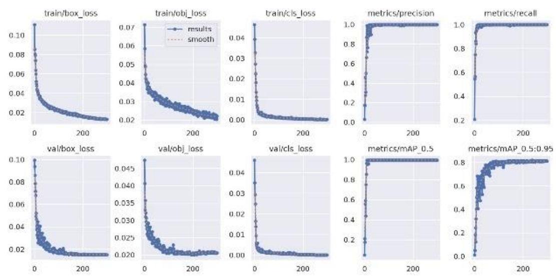

Figure 9 Result of our training

V. REFERENCES

[1] A Convolution Neural Network-Based Seed Classification System, by Yonis Gulzar, Yasir Hamid, Arjumand Bano Soomro, Ali A. Alwan and Ludovic Journaux, Symmetry 2020, 12(12), 2018

[2] SF-YOLOv5: A Lightweight Small Object Detection Algorithm Based on Improved Feature Fusion Mode, Haiying Liu, Fengqian Sun, Jason Gu and Lixia Deng, Sensors 2022, 22(15), 5817

[3] https://docs.ultralytics.com/yolov5/tutorials/architecture_description/

[4] https://github.com/ultralytics/yolov5