Spotting the Golden Grains: Innovative Wheat Detection Techniques

Rohan Venkatesh Sirigeri

1. OVERVIEW OF THE PROJECT

The goal of this project is to develop a sophisticated approach to a Learning model capable of identifying wheat grains in the crops accurately. The data set acquired for the model is a 1024X1024X3 image and the output is the coordinates of the wheat grains in that image. After conducting a thorough investigation, I have discovered that the problem of wheat grain detection has been previously tackled which only involves advanced Deep Learning techniques. However, I aim to improve the existing solutions by incorporating various advanced data augmentation techniques. As a detection problem, accurately identifying different types of wheat grains can be challenging due to the significant variations in their appearances. To overcome this, I plan to implement cutting-edge data augmentation[1] techniques to increase the diversity of the data which is my primary contribution to this project. This approach will enable the model to learn from a more extensive range of samples, leading to higher accuracy in identifying different types of wheat grains.

The model will help various stakeholders in the wheat industry, including farmers, harvesters, and stores. By utilizing this technology, farmers can efficiently identify the different types of wheat grains they are growing, which can help optimize their farming practices. Harvesters can sort the harvested wheat into distinct categories with greater ease, saving time and labor. Finally, stores can use this technology to ensure the quality and consistency of their wheat products, leading to increased customer satisfaction. The successful completion of this project will undoubtedly lead to numerous benefits for all stakeholders involved in the wheat industry.

2. APPROACH

For this project, we first have to Augment the data and then perform Learning of the model based on the augmented data.

2.1. Data Augmentation

Data augmentation is a popular technique used in machine learning to artificially increase the size of a dataset by creating modified versions of existing data points. According to [2]using data augmentation helps in increased diversity of the dataset, and better use of limited data which in turn helps in achieving better accuracies. Some data augmentation techniques used in the project are:

2.1.1. Mosaic

Mosaic[3] augmentation is an advanced technique used in image processing and computer vision applications. It involves dividing the original image into a grid of equal-sized blocks, shuffling them randomly, and then reassembling the image with the shuffled blocks. This process leads to a new image that resembles a mosaic, where the details and features of the original image are present in each block. Mosaic augmentation can be performed in different ways, depending on the size of the grid and the shuffling strategy used. Additionally, the blocks can be shuffled in various ways, such as randomly or in a predefined pattern.

2.1.2. Random Affine

Random affine[4] augmentation is an advanced technique used in image processing and computer vision applications. It involves applying random transformations to the original image, such as the angle of rotation, scale factor, or shear value and translation, to create a new augmented image. The goal of this technique is to increase the diversity of the training data and improve the generalization capability of the model. By applying random affine transformations, the model can learn to recognize objects from different viewpoints and orientations, leading to improved accuracy and performance.

2.1.3. HSV color space

HSV[5] (Hue-Saturation-Value) color space augmentation is an image processing technique used to create augmented images for training machine learning models. In HSV color space, each pixel in an image is represented by three values: hue, saturation, and value. HSV color space augmentation involves randomly modifying the hue, saturation, and value channels of an image to create new variations of the original image. To perform HSV color space augmentation, the original image is first converted from the RGB color space to the HSV color space using an appropriate image processing library, such as OpenCV. Then, random modifications are applied to the hue, saturation, and value channels by multiplying each channel by a random gain value between 0.5 and 1.5. The resulting values are then clipped to a maximum value of 255 to ensure they are within the valid range for the image pixel values.

2.1.4. Albumentations

The Albumentations[6] library is designed to be fast and efficient, with support for both CPU and GPU acceleration. It is also highly customizable, allowing users to specify the exact transformations they want to apply to an image and how those transformations should be applied. One of the key benefits of Albumentations is its ability to apply multiple transformations to an image in a single pass. This can help reduce the amount of processing time required to apply the augmentations, making it a good choice for large datasets and real-time applications such as lens distortion, motion blur, and rain effects.

There are some disadvantages in using Data Augmentation like increased training time, risk of overfitting and difficulty in choosing the right augmentations. But the advantages supersede the disadvantages of our approach.

2.2. Learning Model

The learning model used for this approach is a Pre-Trained model called FasterRCNN with the pre-defined weights of the COCO dataset. The pre-Trained model is used here to save time, to generalize for new and unseen data, and to improve accuracy by using the weights of the model trained before which helps in reaching better weights in fewer epochs.

2.2.1. FasterRCNN with RESNET50 as Backbone

The Faster R-CNN[7] architecture is a two-stage object detection framework that uses a Region Proposal Network (RPN) to generate potential object locations and a Fast R-CNN network to classify and refine the proposed regions. The RPN generates proposals by sliding a small network over the convolutional feature map, which are then filtered and refined by the Fast R-CNN network. The ResNet50 convolutional neural network is a deep neural network architecture that is pre-trained on a large-scale image classification task. It is known for its ability to extract high-level features from images and has been widely used as a backbone feature extractor for various computer vision tasks, including object detection. By combining the Faster R-CNN architecture with a pre-trained ResNet50 model as a backbone feature extractor, the model can achieve state-of-the-art object detection performance with high accuracy and fast inference time.

Some of the cons of using Pre-Trained models are limited customization, dependency on the quality and size of pre-training data, and overfitting of the model. But this supersedes the Advantages of this approach for this project.

The Data Augmentation aspect of the project is coded on the submitter's own. Adding this aspect will help in achieving better accuracy when used in real-world applications. If in the future this model is deployed to a robot to automate harvesting, based on the angle, and translation of the camera of the robot, it has to accurately locate the wheat grains in the crops.

The Learning Model aspect of the project is taken from online resources[8]. This pre-trained model will help in faster learning of the model as it uses weights taken from the COCO[9] dataset's learning.

3. EXPERIMENTAL PROTOCOL

The dataset to perform this experiment is acquired from an existing Kaggle[10] competition. This dataset comprises 1024 X 1024 X 3 images segregated as a training dataset comprising 3422 images and a test dataset comprising 10 images with a .csv file comprising all the coordinates of wheats present in each image of the training dataset. This coordinates is very much important for the project because it helps the model to learn the features of the wheat grains present in the image. Some of the images from the dataset are in Figure1.

Fig. 1. Input Image from Dataset

The evaluated success can be derived qualitatively by seeing the detected wheat grains in the test image. The model can be deemed successful if it has detected almost all the wheat grains in a particular image as shown in Figure2.

This project utilizes Deep Learning Framework like PyTorch to train the model. The model used is a PyTorch Pre-Trained model along with the weights file. Since the dataset is very small (around 643.57 MB), it takes around five minutes to train the model with this dataset when run in a GPU accelerator[11]. GPU helps in acquiring an efficient timeframe for model training.

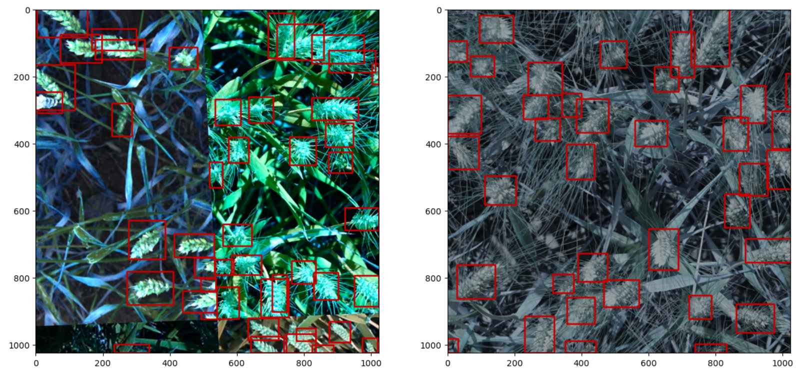

Fig. 2. Sample Expected Result

4. RESULTS

By performing the above approach on the computational resources mentioned above, we obtained the results as shown in Figure3.

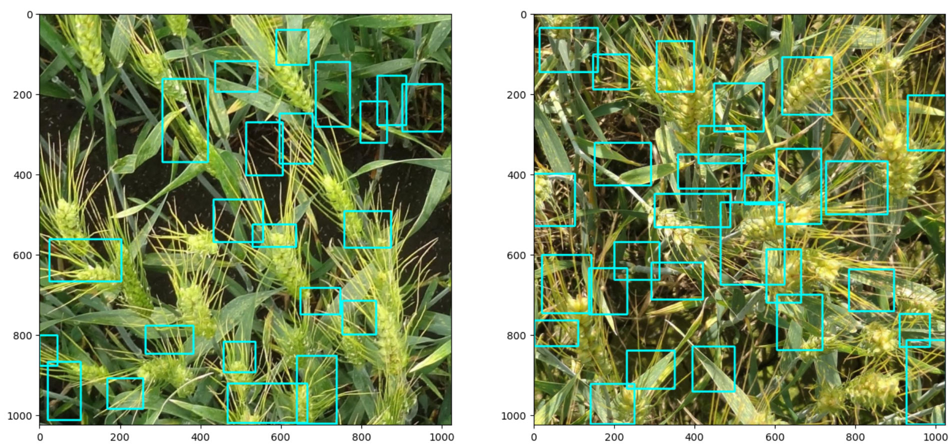

Fig. 3. Result Obtained

From Figure3, we can see that the model has almost detected the wheat grains present in the crop.

To check whether the result obtained from my model is as good as the state-of-the-art model as shown in Figure4 below, we can see it is almost as same as the result obtained by our model. The model has reached good results as compared to the state-of-the-art model because our approach2 includes different augmentation techniques. This difference will increase when our approach2 is deployed in the real-world application tool like robots, etc as our model uses augmentation techniques.

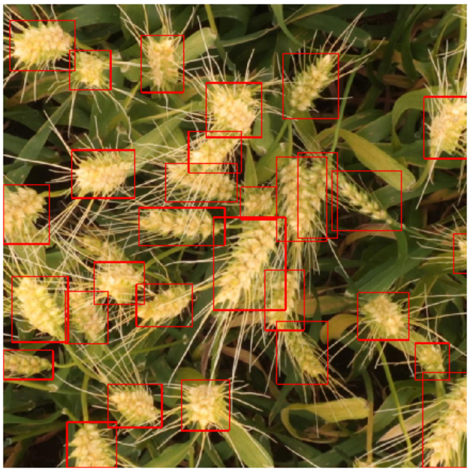

Fig. 4. Result of the State Of the Art Model

Leaving comparison with the state-of-the-art model aside, when we compare the results obtained with the ground truth i.e. the expected results from Figure2, we can see that our approach2 almost recognizes the wheat grains like the expected result's Figure2.

5. ANALYSIS

Some of the limitations of the Faster R-CNN model include high computational requirements, longer training times, and the inability to handle objects that vary significantly in size and shape. Additionally, the model's performance may be affected by factors such as lighting conditions, camera angles, and image quality.

Some of the advantages of the Faster R-CNN model include high accuracy in object detection, the ability to handle occlusions and overlapping objects, and the capability to generate region proposals and classify objects in a single forward pass.

6. DISCUSSION AND LESSONS LEARNED

Through this project, I learned the importance of exploring various data augmentation techniques to improve model accuracy, the benefits of transfer learning and pre-trained models for more efficient and effective training, the value of GPU acceleration for speeding up the training process, the importance of selecting an appropriate model architecture and backbone for the specific task, and the potential impact of the project on various stakeholders and how it could be applied in practical settings.

Acquiring these skills will prove advantageous in my future endeavors as a computer vision researcher/ML engineer.

References

[1] "Data augmentation :," https://medium.com/lansaar/what-is-data-augmentation-3da1373e3fa1, accessed: 2023-05-10.

[2] "Ai tool :," https://chat.openai.com/, accessed: 2023-05-10.

[3] "Mosaic image stiching :," https://pylessons.com/OpenCV-image-stiching, accessed: 2023-05-10.

[4] "Random affine transformation :," https://docs.opencv.org/3.4/d4/d61/tutorial$_{w}arp_{a}ffine.html, accessed : 2023-05-10.$

[5] "Hsv color spaces :," https://docs.opencv.org/4.x/df/d9d/tutorial$_{p}y_{c}olorspaces.html, accessed : 2023 - 05 - 10.$

[6] "Albumentations :," https://albumentations.ai/docs/getting$_{s}tarted/installation/, accessed : 2023 - 05 - 10.$

[7] Y. Zhou, S. Wen, D. Wang, J. Mu, and I. Richard, "Object detection in autonomous driving scenarios based on an improved faster-rcnn," *Applied Sciences*, vol. 11, no. 24, p. 11630, 2021.

[8] "Online resources :," https://www.kaggle.com/code/pestipeti/pytorch-starter-fasterrcnn-inference/notebook, accessed: 2023-05-10.

[9] "Coco dataset :," https://cocodataset.org/home, accessed: 2023-05-10.

[10] "Kaggle dataset link :," https://www.kaggle.com/competitions/global-wheat-detection/data, accessed: 2023-05-10.

[11] "Kaggle gpu :," https://www.kaggle.com/product-feedback/361104, accessed: 2023-05-10.