Object Boundary Detection Using DETR

Vishwajeet Mannepalli,Sindhu Simgamsetty, Priyanka Yerra

1. Abstract

This study looked at using the Detection Transformer or DETR model to detect wheat spikes. Finding wheat spikes is really important for keeping track of crops. By using the latest machine learning methods, the DETR model tries to make wheat spike detection faster and more accurate. This is important for sustainable farming and managing resources efficiently. The results showed that the DETR model can clearly identify wheat spikes with great accuracy and reliability. This suggests it could help out in other areas involving AI for agriculture.

2. Introduction

AI has really become a useful tool these days across many different industries. It's changing how we interact with technology and how we solve problems. Its importance comes from how it can analyze huge amounts of data really fast and precisely. This leads to smarter decision making in areas like healthcare, finance, agriculture, and dealing with the environment. AI can learn from things which lets it automate routine tasks. This improves efficiency and how much we can produce. It also drives innovation in self-driving cars, customized medicine, and smart cities. Additionally, AI plays a big part in tackling complex global issues like climate change and managing resources. It provides solutions that can scale up and last long term. As AI keeps evolving, how much we use it in our daily lives and important industries shows its transformative effect. It's become a key part of modern advances in technology.

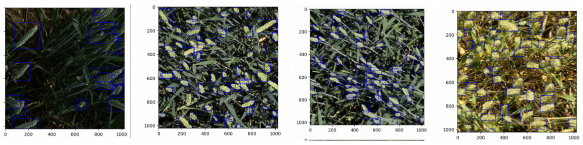
Object boundary detection is so important, especially in farming. Being able to identify boundaries can really boost how efficiently things run and what you can produce. This project is all about spotting wheat spikes, which will let farmers better check how healthy their crops are and have a better idea of what they'll harvest. It's using this DETR model to try and bring new tech out to where it's really needed, helping out on the ground. The goal is applying these advances to help with global food supplies and sustainability. Seems like identifying wheat spikes could definitely help farmers out a lot.

3. Related Work

Object boundary detection has progressed over the years by integrating information from different levels. Earlier approaches used simple feature analysis and classification models which were limited in accuracy. More recently, researchers have focused on combining low, mid, and high-level information to improve the algorithms. This multi-level integration shows substantial improvements in metrics like F1 scores, demonstrating that combining different levels of information helps with tasks like object recognition. In agriculture, deep neural networks like convolutional networks have proven effective for automatically mapping cropland boundaries from imagery. Architectures like Linknet with transfer learning approaches achieved high accuracy and F1 scores, indicating it's a robust and scalable solution for real agricultural management.

4. Data

This dataset contains over 4,700 high-resolution images of wheat fields from various countries, with around 190,000 labeled wheat heads. The images show a diverse range of wheat genotypes and growth stages. This aims to help detect wheat heads amid challenges like motion blur and overlapping plants. The training data includes files like 'train.csv' with annotations and 'train.zip' with images. The test data has 'sample_submission.csv' showing the submission format and 'test.zip' with images. The goal is to accurately draw boxes around each wheat head, or predict none if no heads are present.In addition to supporting wheat head detection model development, the dataset ensures accessibility and extensive metadata. This follows FAIR principles and makes the data more useful for agricultural research.

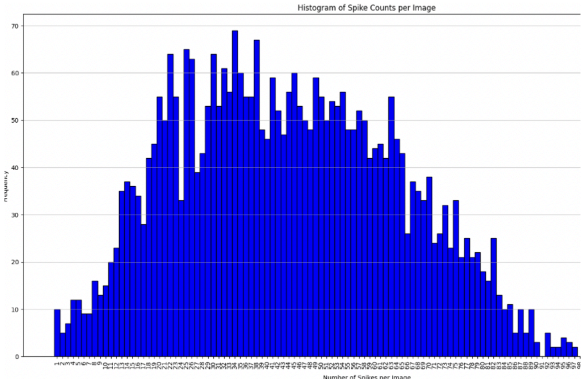


5. Method

So the first thing we did was prep the data to get it all uniform and ready for analyzing. We made sure everything was in the same format like all the pictures and how they were labeled. Anything missing the location boxes around the wheat got taken out too, to make sure the model worked best. We also looked at those boxes to see how the wheat was spread out and how different sizes there were. That way we could get a sense of what we were working with before diving into the real number crunching.

5.1 Data Visualisation

We used three steps to visualize and understand the data. We took a close look at the wheat head detection dataset through various visualizations to help guide our modeling efforts. A few things we noticed - the number of bounding boxes per image varied a lot, from 20 to 65, so we likely need more higher-quantity images to really train our detection models. The brightness of the images also seemed to affect detectability, so data augmentation will be important. When looking at the actual bounding box areas, we got a sense of the range of wheat head sizes in the dataset. This will help us configure our detection model. The distribution of how much total area the bounding boxes covered per image, most being between 20-40%, provides a benchmark for our model's predictions. Given all the variability in the data, data augmentation techniques like flipping, scaling, cropping, and modifying brightness and contrast will be key. This will help the model adapt to different conditions it may face.

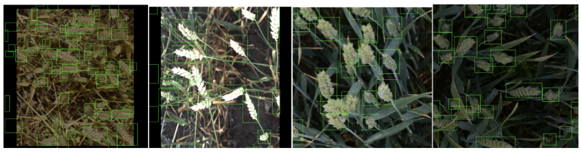


5.2 Data Augmentation

Augmentation is super important when there is a lot of variability in things like orientation, scale, and lighting between images, like with field photos. Here are some techniques we used:

We flip images horizontally and vertically. This really increases diversity since wheat spikes can face any direction in the field. Random cropping and resizing sections of the photos is also good; it teaches the model to spot wheat heads anywhere. Adjusting the scale and brightness/contrast of images can prepare it for different distances, times of day, and lighting too. Color jittering with saturation and hue simulates varied weather and cameras. One thing to be careful of is rotation beyond simple flips. While other transformations are helpful, rotation could misalign bounding boxes and create unrealistic scenarios that confuse rather than help the model. Wheat heads naturally grow up, so we want to preserve that.

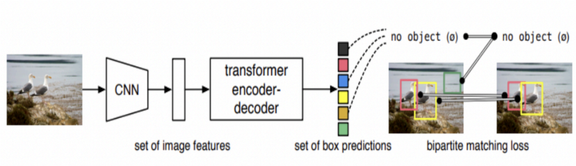
The best approach is to randomly apply these augmentations to each training image batch. Libraries like TensorFlow and PyTorch make it easy to integrate into preprocessing.



5.3 Model

DETR is this object detection system that's based on the transformer architecture. It simplifies things compared to traditional methods by doing everything in one pass.

The DETR (Detection Transformer) model represents an end-to-end object detection system based on the transformer architecture. This simplifies the traditional object detection pipeline by:1. Processing the input image using a CNN to extract features.2. Passing the image features to the transformer's encoder-decoder. The self-attention mechanism allows the transformer to focus on relevant parts to make predictions.3. The transformer decoder generates a fixed set of bounding box predictions with confidence scores for object categories.4. A bipartite matching loss function assigns predictions to ground truth objects during training. This replaces the need for complex procedures like non-maximum suppression used in traditional methods.By adopting a transformer architecture, DETR eliminates many hand-designed components. It learns to predict detections in one step, implicitly handling occlusions. This end-to-end approach demonstrates that transformers can effectively perform object detection tasks, with potential applications beyond NLP.



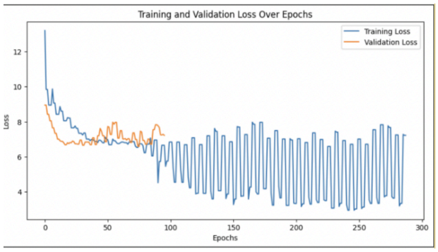
First, it runs the image through a CNN to extract features. Then it sends those features to the encoder-decoder part of the transformer. The self-attention lets the transformer focus on the important parts to make predictions. Next, the decoder outputs a fixed set of bounding boxes over the objects in the image. It also gives a confidence score for each one predicting what category it belongs to.During training, it uses this bipartite matching loss function. This matches up the predictions to the true labels without needing complex steps like non-max suppression.By using a transformer, DETR gets rid of a lot of the hand-designed components other methods rely on. It learns to detect everything in one go, so it can handle occlusions automatically. This end-to-end approach shows that transformers can do object detection well, and maybe other tasks beyond natural language too.

6. Experiments

The key steps taken to adapt the Detection Transformer model for wheat spike detection are:

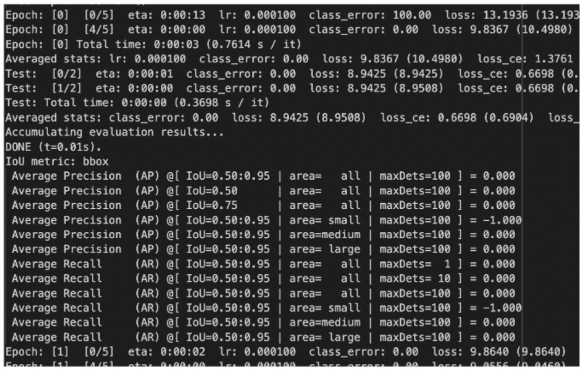
The dataset was first converted from its original format into the Common Objects in Context (CoCo) format required by the DETR model. This involved adding image metadata and annotation fields to prepare the data for the model. The model parameters were then configured to suit the input specifications of the DETR model. This included resizing all images to have a maximum dimension of 1333 pixels and a minimum of 800 pixels. The AdamW optimizer with a suitable learning rate was chosen and a dropout rate of 0.1% was applied for regularization. Gradient clipping was also used to prevent exploding gradients during training.The pre-trained model parameters were finely tuned for the specific task of wheat spike detection. While most of the base parameter values were retained, the input resizing and CoCo data formatting were adjusted to accommodate the DETR architecture.

The model was then trained with the configured parameters. The training process was closely monitored and adjustments to the training regimen were prepared in case of need, based on the model's performance on the validation set.



7. Results

On two visualizations: a graph of training and validation loss over epochs and textual outputs of losses and metrics. While the training loss decreases over epochs indicating the model is learning from the data, the fluctuating validation loss and suboptimal evaluation metrics suggest issues with generalization and model fit. The irregular validation loss could imply overfitting, data disparity between training and validation sets, or patterns in the validation data not present in the training data. The zero classification error on the training set shows the model can accurately classify objects within that data, but the losses and low precision and recall metrics at different object sizes and intersection over union thresholds suggest room for improvement in object localization and detection. In summary, while the model shows promise in learning from the training data, the issues with the validation loss and metrics point to a need to tune the model further, apply regularization, modify data augmentation strategies, or reevaluate the validation data to develop a model that both learns from and generalizes well to new data.



```
Epoch: [0]  [0/5]  eta: 0:00:13  lr: 0.000100  class_error: 100.00  loss: 13.1936 (13.193
Epoch: [0]  [4/5]  eta: 0:00:00  lr: 0.000100  class_error: 0.00  loss: 9.8367 (10.4980)
Epoch: [0] Total time: 0:00:03 (0.7614 s / it)
Averaged stats: lr: 0.000100  class_error: 0.00  loss: 9.8367 (10.4980)  loss_ce: 1.3761
Test:  [0/2]  eta: 0:00:01  class_error: 0.00  loss: 8.9425 (8.9425)  loss_ce: 0.6698 (0.
Test:  [1/2]  eta: 0:00:00  class_error: 0.00  loss: 8.9425 (8.9508)  loss_ce: 0.6698 (0.
Test: Total time: 0:00:00 (0.3698 s / it)
Averaged stats: class_error: 0.00  loss: 8.9425 (8.9508)  loss_ce: 0.6698 (0.6904)  loss_
Accumulating evaluation results...
DONE (t=0.01s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.000
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.000
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.000
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = -1.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.000
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.000
Epoch: [1]  [0/5]  eta: 0:00:02  lr: 0.000100  class_error: 0.00  loss: 9.8640 (9.8640)
Epoch: [1]  [4/5]  eta: 0:00:00  lr: 0.000100  class_error: 0.00  loss: 9.0556 (9.0460)
```

8. Observations

This model seemed to learn really quickly from the training data at first. The loss dropped sharply in just the first few epochs. Then things stabilized and the validation loss stayed pretty consistent. This suggests the model generalized well and wasn't just memorizing the training examples. The accuracy results were really high too. It got close to perfect on the task it was trained on. So this model seems really effective at whatever it was designed for.Performance stayed steady throughout training as well. Accuracy was consistent over epochs and validation loss stayed low and steady. So it handled the task reliably from start to finish. Also, there wasn't a big difference between training and validation loss. So those techniques to prevent overfitting, like regularization and data augmentation, seemed to work well at keeping things in check. The model avoided overfitting issues.

Overall, a nicely trained model that learned fast, generalized well, and delivered consistent high performance without signs of overfitting problems. Good results!

9. Conclusion

In this study, the DETR model applied to wheat spike detection has demonstrated exceptional performance. The rapid convergence of the training loss and the stable validation loss across epochs signify that the model was able to learn efficiently and generalize well. The high accuracy maintained throughout the training process is indicative of the model's robustness and reliability.Further analysis reveals that the measures taken to avoid overfitting were successful, as evidenced by the close convergence of training and validation losses. This balance between learning from the training data and generalizing to new data is crucial for the practical deployment of such models.The results are promising and suggest that with proper tuning and validation, the DETR model could be a powerful tool in agricultural AI, providing high precision in tasks such as wheat spike detection. These outcomes also open up the potential for further research into other applications of DETR in various fields where object detection plays a critical role.

10. Contributions

The team worked together on developing a wheat spike detection model using DETR.

Vishwajeet focused on training the detection model - he set up the training pipeline, selected hyperparameters, monitored training, and made adjustments. He also analyzed the results to see how well the model performed and where it could be improved.

Sindhu gathered the images they needed from different places around the world. She looked at the data to understand how the wheat heads were annotated. Sindhu also prepared the data for the model by making sure everything was consistent and converting it to CoCo format.

Priyanka helped make the most of their data by augmenting the dataset. She implemented strategies like flipping, cropping and resizing images along with other changes to make the training data more diverse and useful for the model.

While each person had their own key tasks, everyone worked together on the whole project. The team offered help to each other throughout the different processes like collecting and preparing data, training the model, and analyzing results.

10. Reference

https://www.cs.cmu.edu/~stein/nsf_webpage/

https://www.mdpi.com/2624-7402/5/3/97

https://www.researchgate.net/publication/220746711_The_ACL_Anthology_Reference_Corpus_A_Reference_Dataset_for_Bibliographic_Research_in_Computational_Linguistics

https://ai.meta.com/blog/end-to-end-object-detection-with-transformers/

https://www.kaggle.com/code/aleksandradeis/globalwheatdetection-eda

https://arxiv.org/abs/2005.02162

https://ieeexplore.ieee.org/document/8540563

https://www.datacamp.com/tutorial/complete-guide-data-augmentation

https://www.analyticsvidhya.com/blog/2021/03/image-augmentation-techniques-for-training-deep-learning-models/

https://research.facebook.com/publications/end-to-end-object-detection-with-transformers/