Global Wheat Detection

Final Presentation

Our Kaggle Journey

Lukas Julius Eule & Janine Berndt

Global Wheat Detection

- Introduction
- EDA
  - Dataset – Basic Overview
  - Bounding Box Analysis
- Faster R-CNN
  - Without Augmentation
  - With Augmentation
- Conclusion - Outlook
- Dataset Acknowledgement/ References

Introduction

- Objective:
  - Detect wheat heads in outdoor images to improve data on wheat density and size globally
  - Aid farmers and researchers in assessing crop health, maturity, and yield

- Challenges in Detection:
  - Overlapping plants and blurred images
  - Variations in genotype, colour, and maturity
  - Diverse global growing conditions

- Dataset:
  - Training data: France, the UK, Switzerland, and Canada
  - Test set: Australia, Japan, and China
  - Supported by organizations like the Global Institute for Food Security and DigitAg

- Goals:
  - Improve data on wheat density and size across global varities
  - Support more informed and effective crop management

Dataset – Basic Overview

Files

- train.csv - the training data
- sample_submission.csv - a sample submission file in the correct format
- train.zip - training images
- test.zip - test images

train.csv

	image_id	width	height	bbox	source
0	b6ab77fd7	1024	1024	[834.0, 222.0, 56.0, 36.0]	usask_1
1	b6ab77fd7	1024	1024	[226.0, 548.0, 130.0, 58.0]	usask_1
2	b6ab77fd7	1024	1024	[377.0, 504.0, 74.0, 160.0]	usask_1
3	b6ab77fd7	1024	1024	[834.0, 95.0, 109.0, 107.0]	usask_1
4	b6ab77fd7	1024	1024	[26.0, 144.0, 124.0, 117.0]	usask_1

! No missing values

```
Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 147793 entries, 0 to 147792
Data columns (total 5 columns):
 #   Column     Non-Null Count   Dtype
---  ------     --------------   -----
 0   image_id   147793 non-null  object
 1   width      147793 non-null  int64
 2   height     147793 non-null  int64
 3   bbox       147793 non-null  object
 4   source     147793 non-null  object
dtypes: int64(2), object(3)
memory usage: 5.6+ MB
None
```

Bounding Box – Basic Analysis

Bounding Box – Basic Analysis

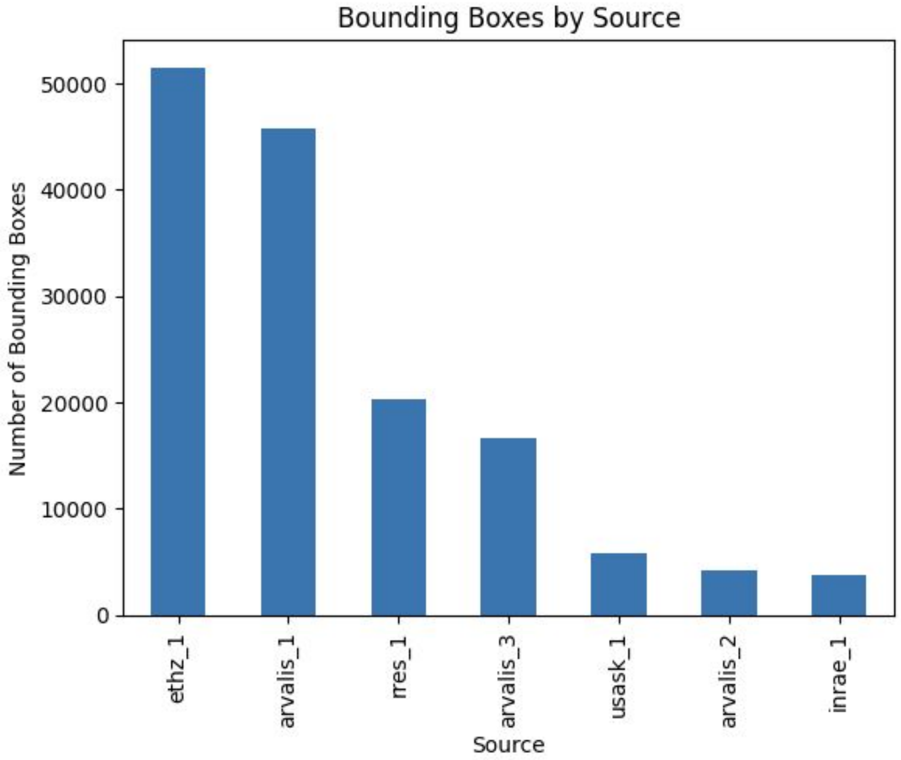
Bounding Boxes Structures

```
0      [834.0, 222.0, 56.0, 36.0]
1     [226.0, 548.0, 130.0, 58.0]
2     [377.0, 504.0, 74.0, 160.0]
3     [834.0, 95.0, 109.0, 107.0]
4     [26.0, 144.0, 124.0, 117.0]
Name: bbox, dtype: object
[<class 'list'>]
```

```
Total Images: 3373
Images with Bounding Boxes: 3373
Images without Bounding Boxes: 0
```

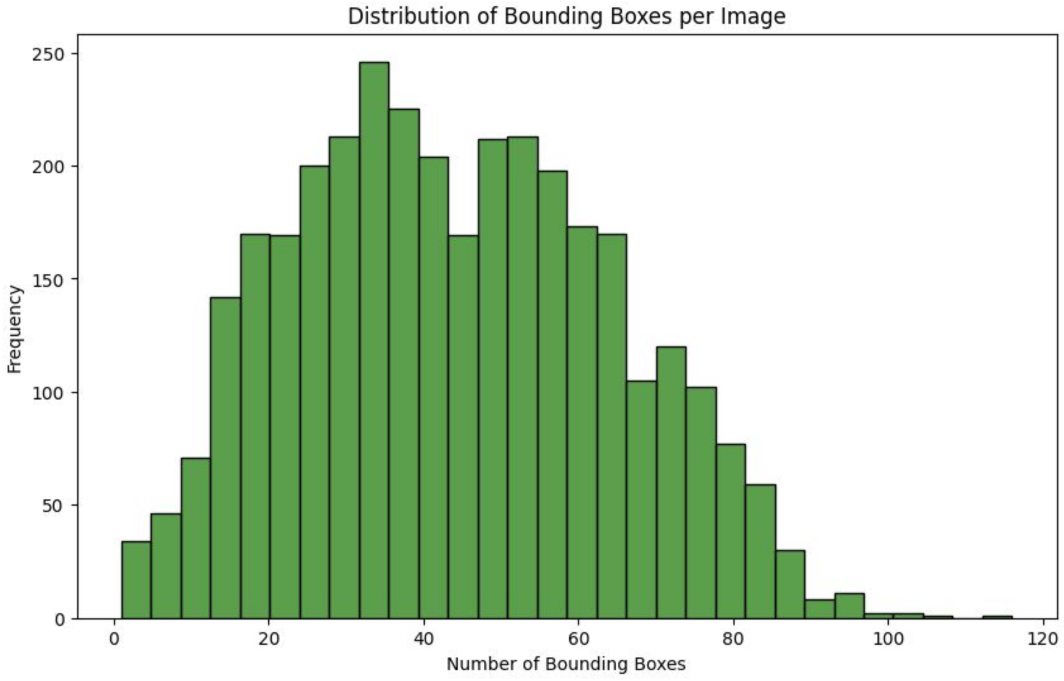
```
Number of images in the folder 'train_images': 3422
Number of unique images in the CSV file: 3373
```

Bounding Boxes per Source

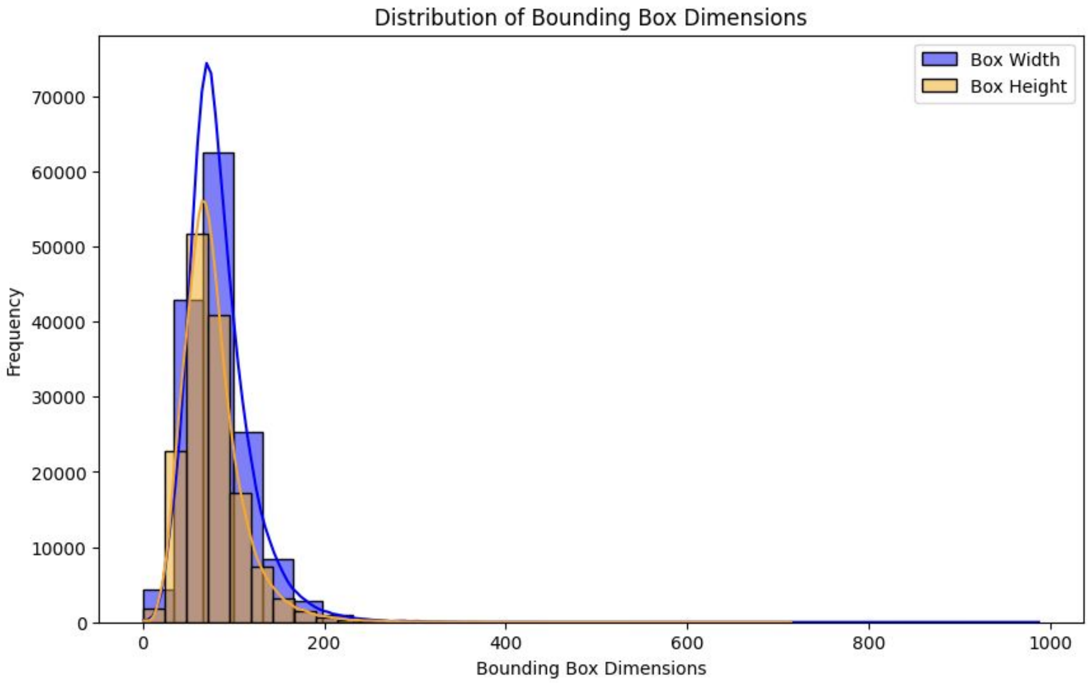


Bounding Box - Basic Analysis

Distribution of Bounding Boxes per Image



Distribution of Bounding Boxes Dimensions

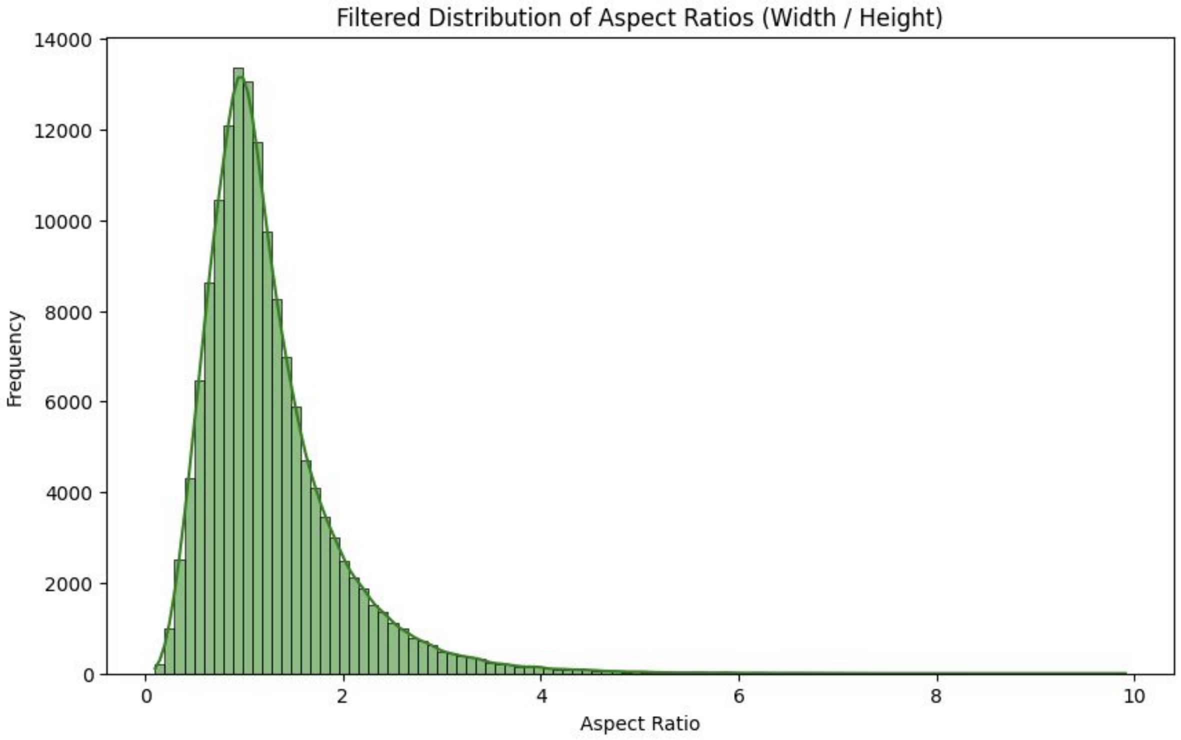


Bounding Box – Insights
Aspect Ratios

Definition:

Aspect Ratio = Width / Height

- proportional relationship between the two dimensions, providing insights into the shape of the object being enclosed
- aspect ratio greater than 1 indicates a wider box, while an aspect ratio less than 1 indicates a taller box.



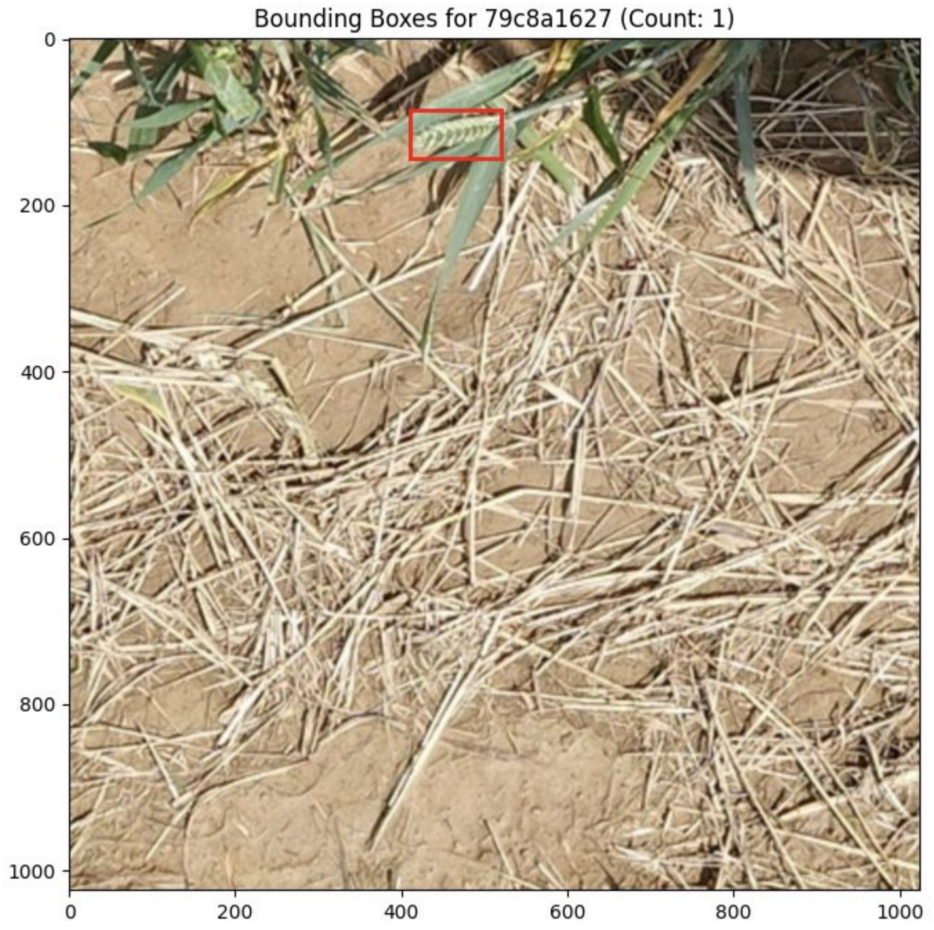
Filtered distribution

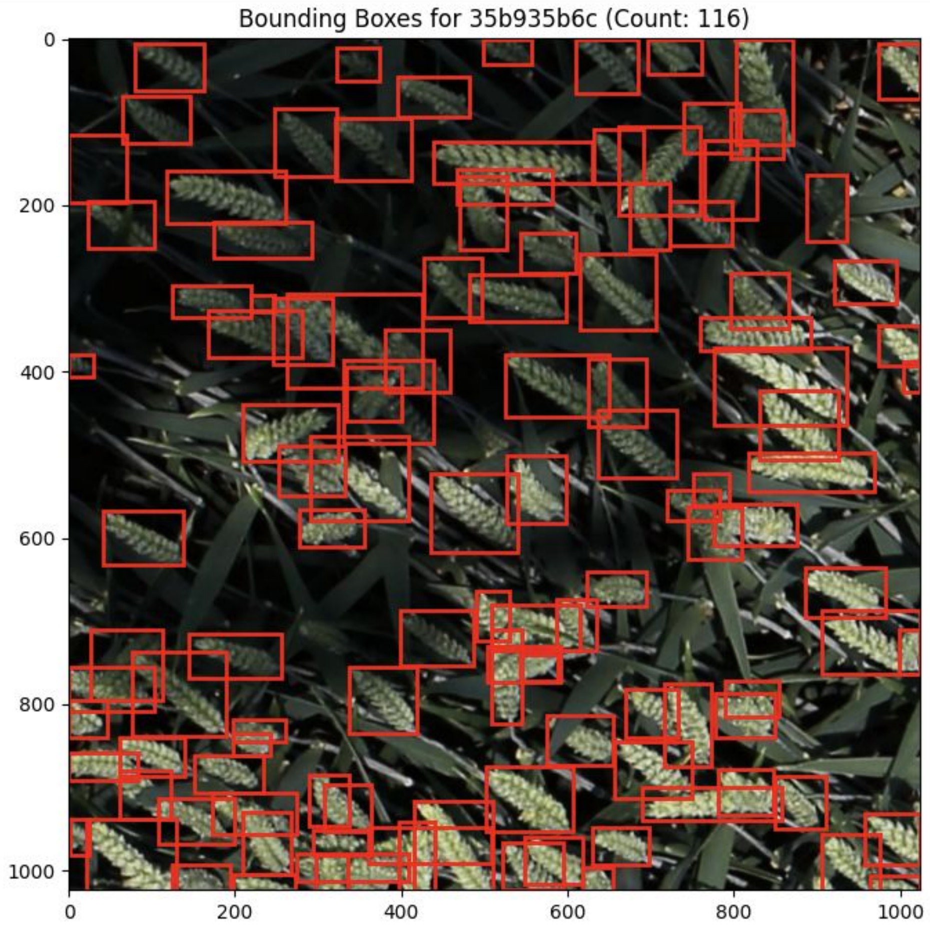
- to focus on bounding boxes with reasonable aspect ratio
- to remove outliers

Key message:

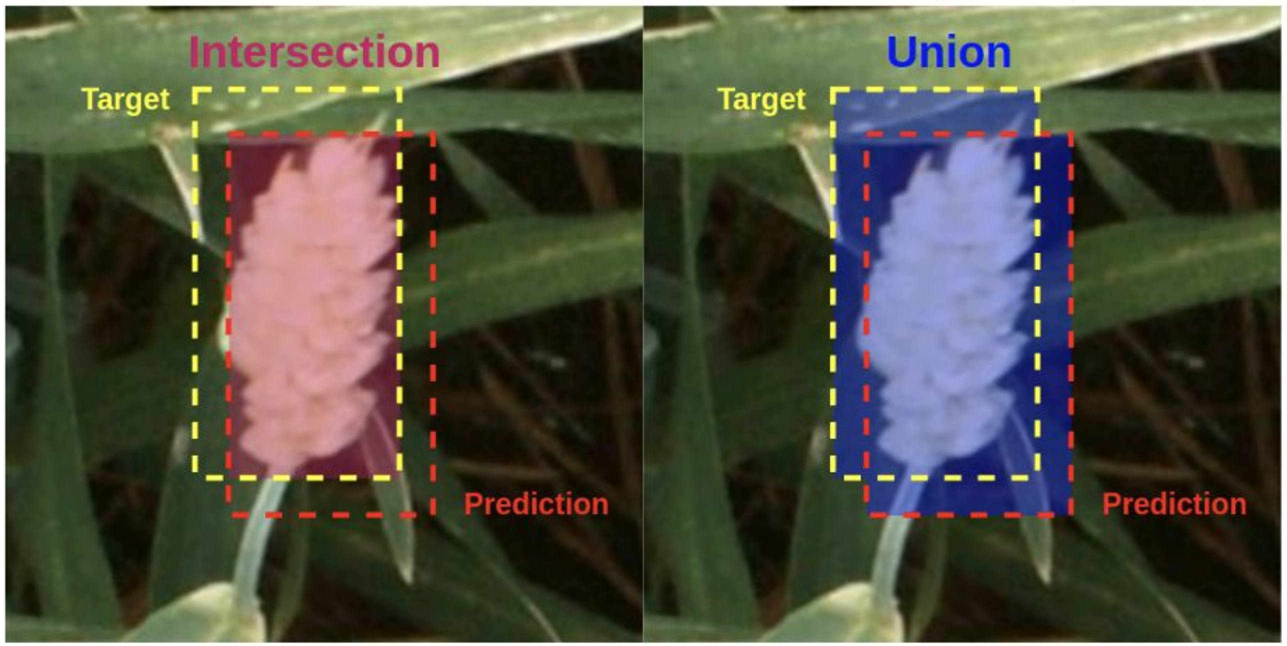
- majority of boxes have an aspect ratio close to 1 indication there are roughly square

Bounding Box – Insights
Edge Case Images





Metric: Intersection over Union (IoU)



$$\frac{\text{Intersection}}{\text{Union}} = \text{Intersection over Union}$$

$$\text{Objectness}_{score}(\text{IoU}) = \begin{cases} \text{IoU} > 0.7 & : \text{positive} \\ 0.5 < \text{IoU} \leq 0.7 & : \text{positive} \\ \text{IoU} < 0.3 & : \text{negative} \\ 0.3 \leq IoU \leq 0.5 & : \text{not negative/ positive} \end{cases}$$

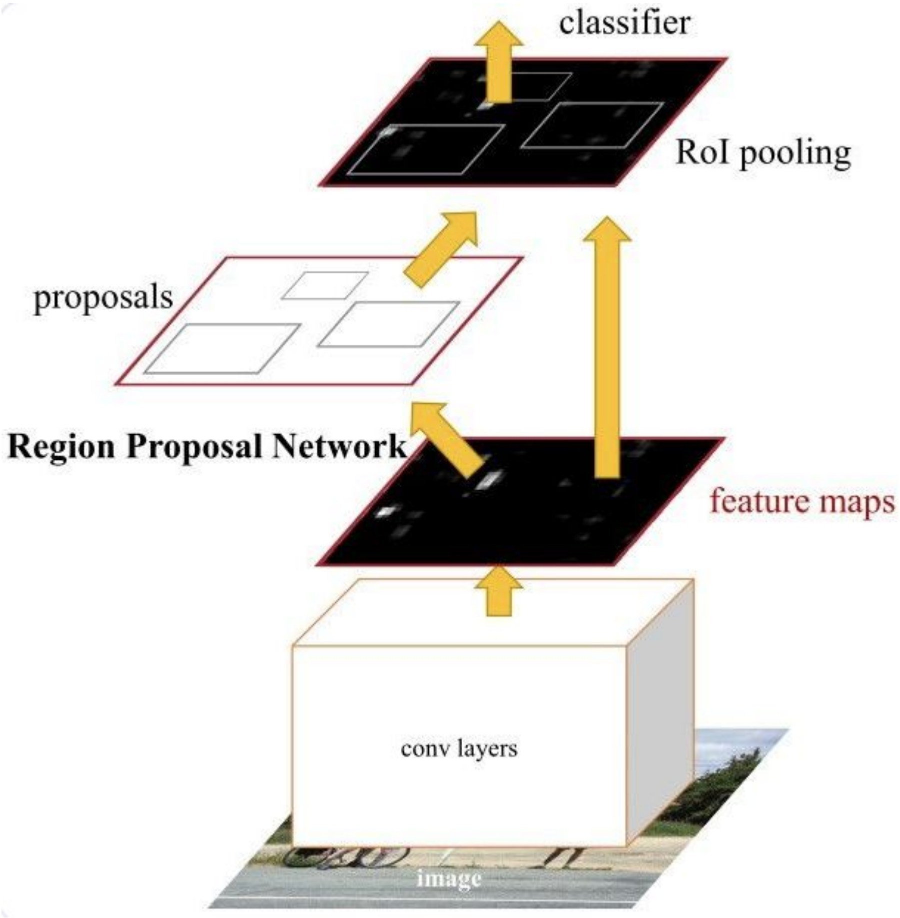
Metric: Mean Average Precision

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives} + \text{False Negatives}}$$

1. Calculates the IoU for all Bounding Boxes in the image
2. Calculates the precision of the prediction for all bounding boxes above a IoU threshold (to be done for thresholds 0.5, 0.55, 0.6, 0.65, 0.7, and 0.75)
3. Takes the mean of the precision values for all thresholds

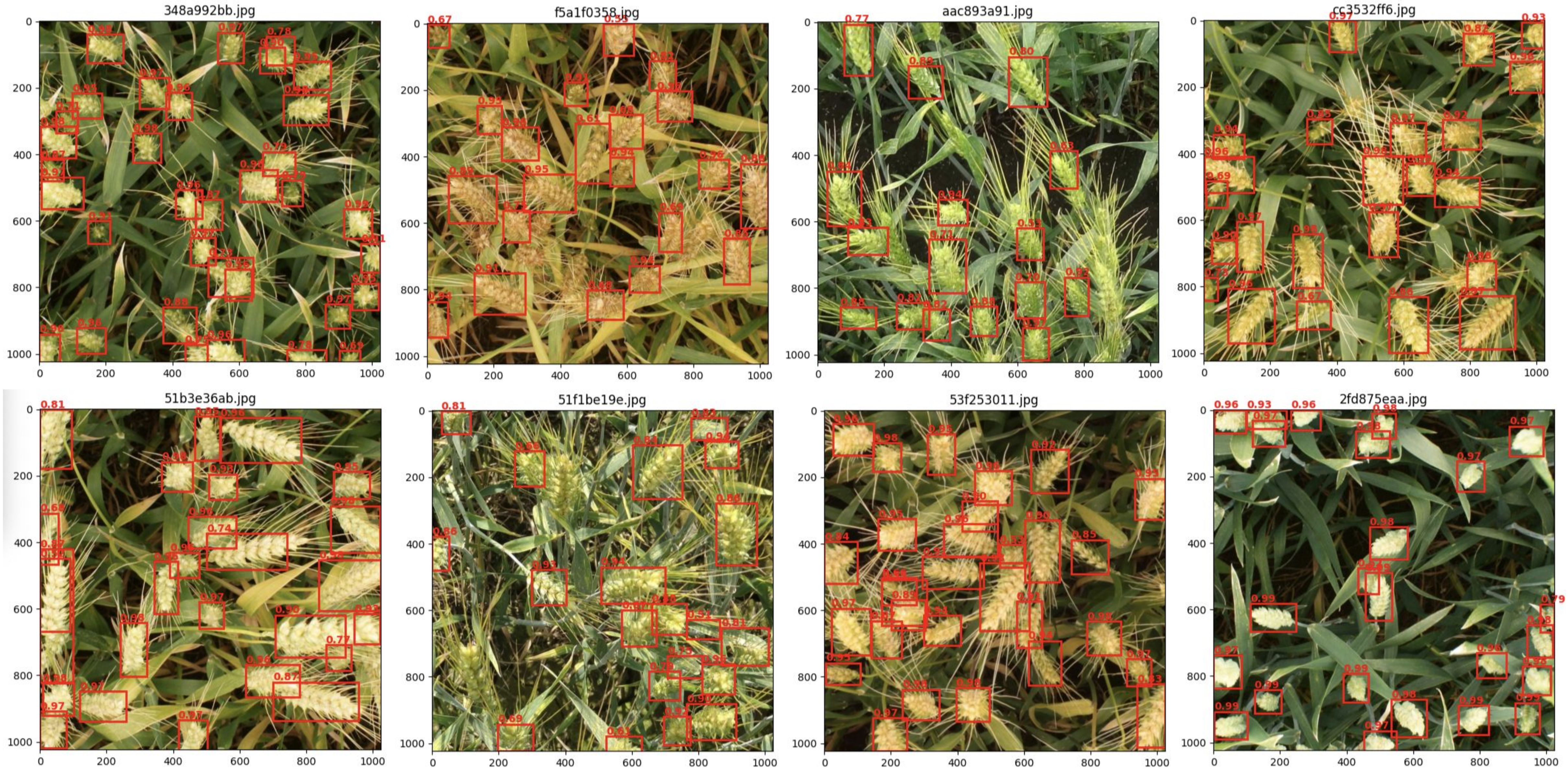
Faster R-CNN (in a nutshell)

1. Input Image – The image is fed into a deep CNN (e.g., ResNet)
2. Feature Extraction – Convolutional layers generate feature maps
3. Region Proposal Network (RPN):
4. RoI (Region of Interest) Pooling:
   - Extracts fixed-sized regions from feature maps
   - Ensures compatibility with the classifier
5. Object Classification & Bounding Boxes Regression
   - Assigns class label to each RoI
   - Refines bounding boxes coordinates for accurate localization
6. Final Output – Detected objects with bounding boxes



Source: *Tutorial: Faster R-CNN Explained for Object Detection Tasks*, DigitalOcean, retrieved 05.02.2025

Faster R-CNN without Augmentation



Data Pipeline

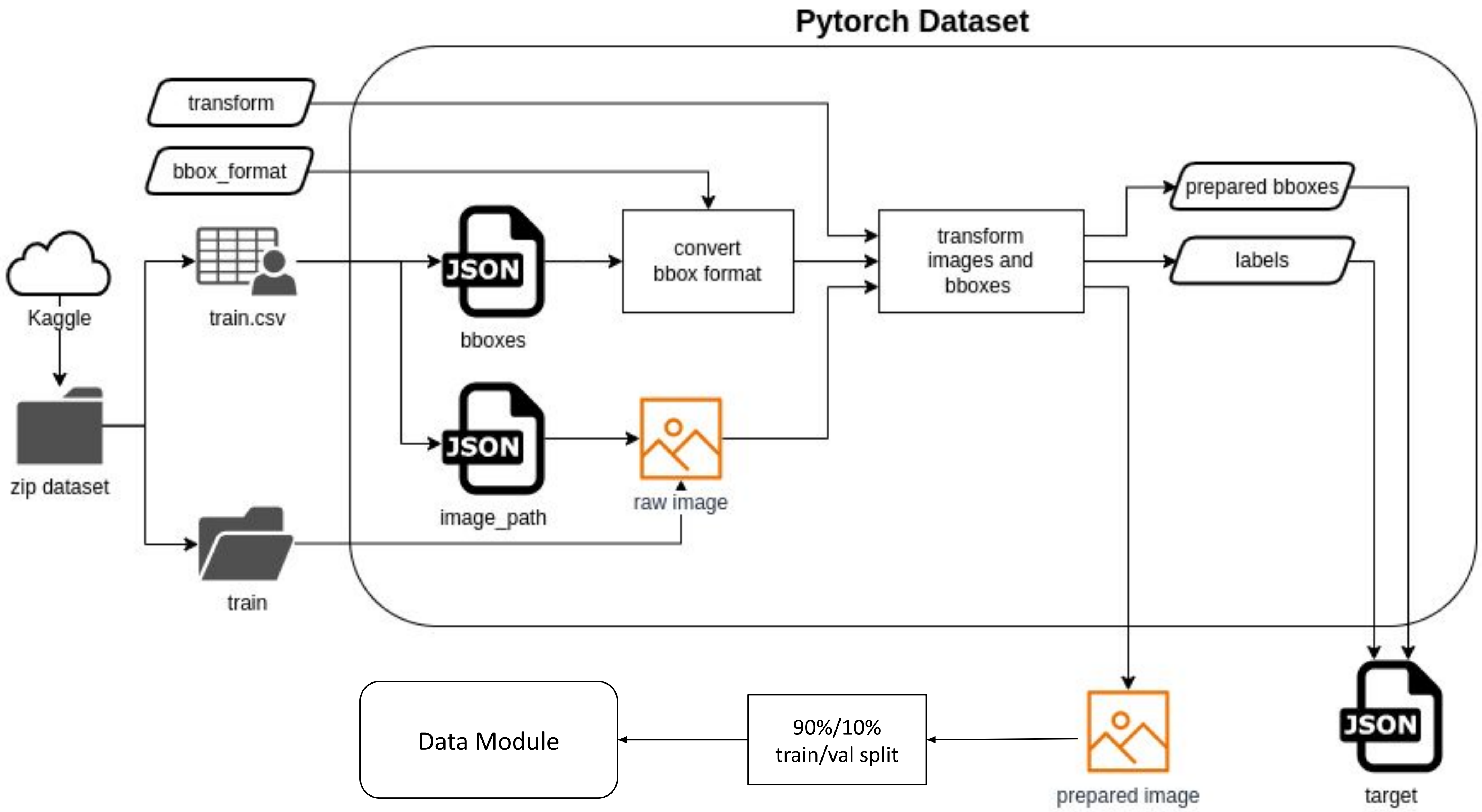
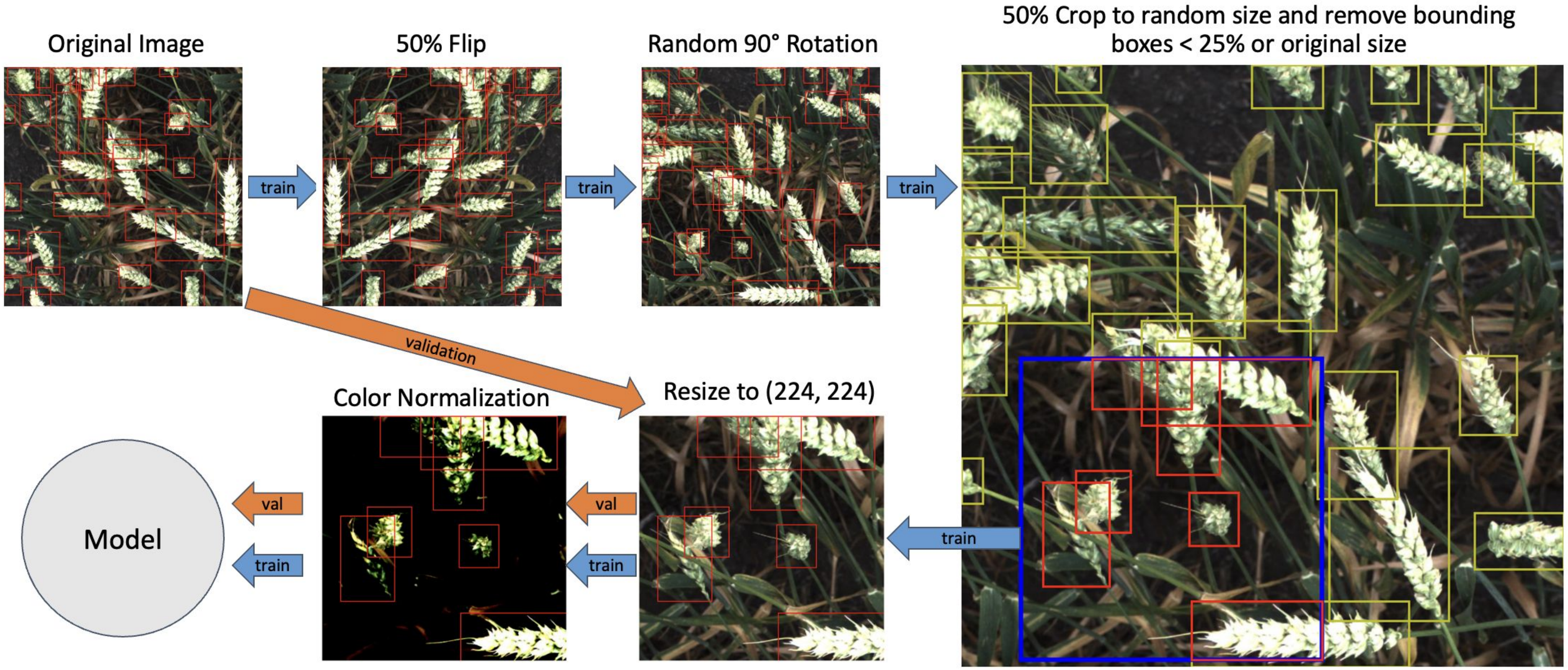


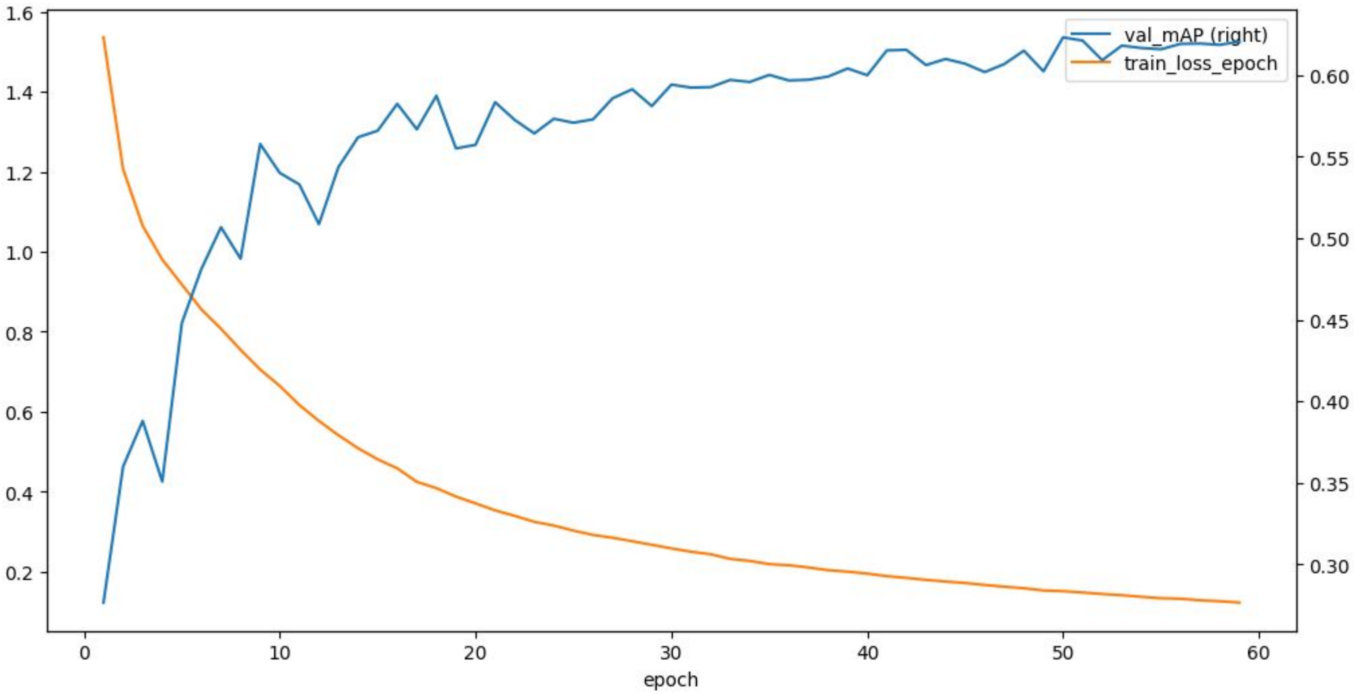
Image Augmentation

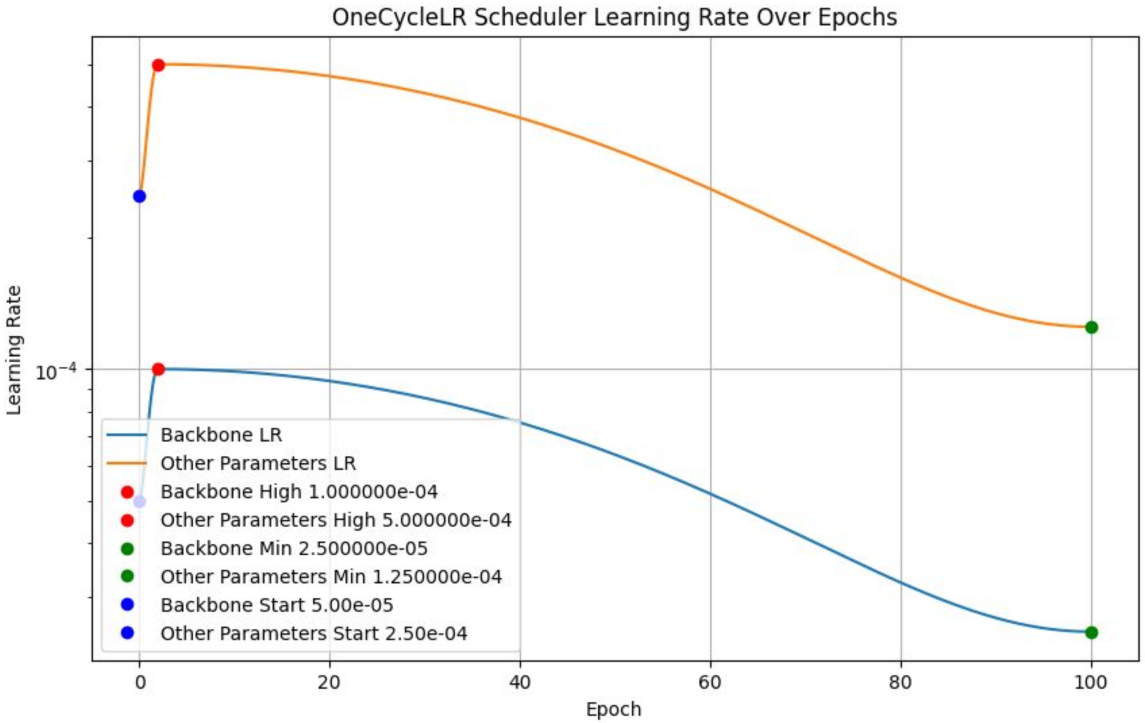


Faster R-CNN with Different Backbones

Backbone	Weights	Augmentation	Epochs (early stopping)	val_mAP
ResNet-50	ImageNet	Rotation, Flip and Normalization	30	0.514453
ResNet-50	ImageNet	Rotation, Flip and Normalization	100	0.535898
ResNet-50	PlantNet-300k	Rotation, Flip, Normalization and Random Crop	66	0.493499
ViT Base Patch 16 224	PlantNet-300k	Rotation, Flip, Normalization and Random Crop	60	0.623210

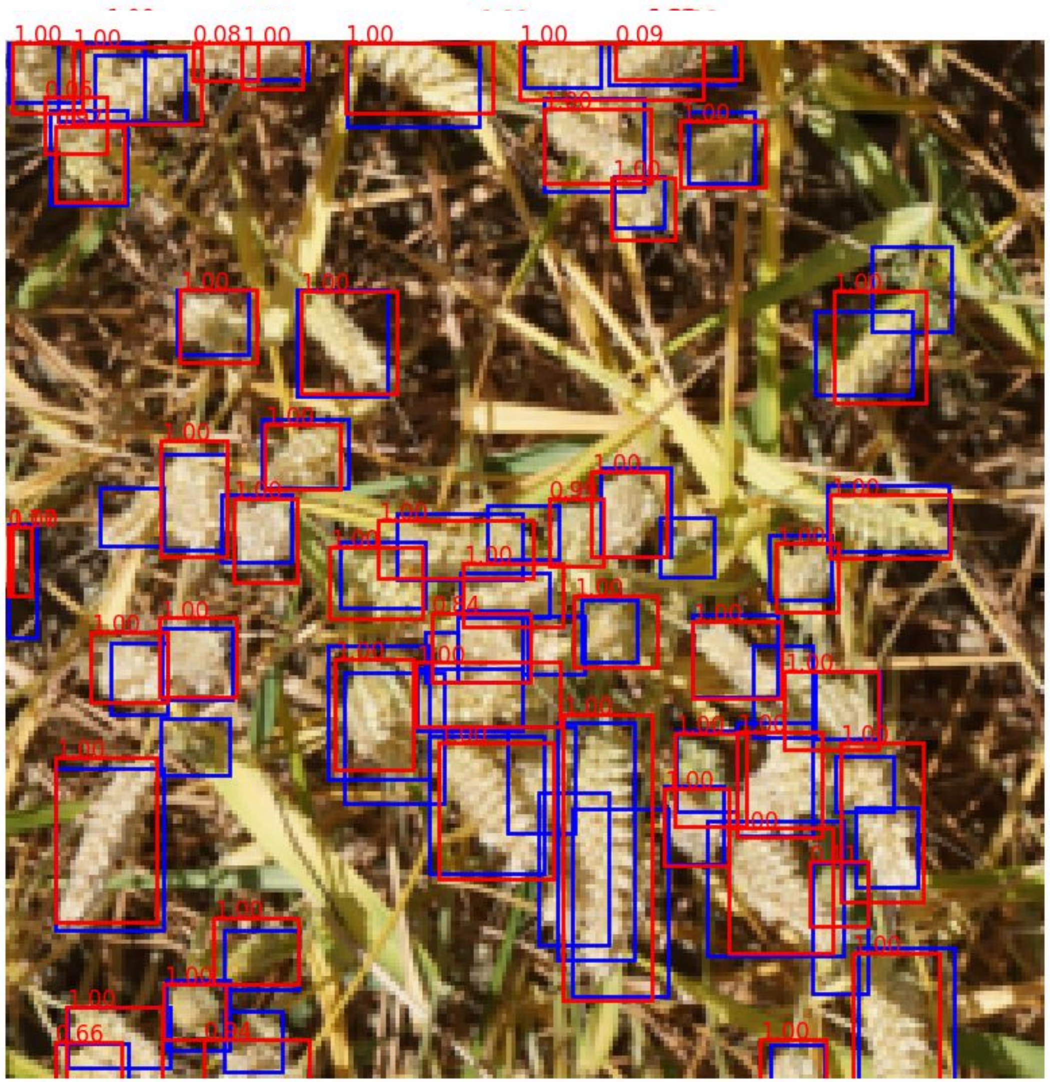
Training Faster R-CNN with ViT Backbone (Best Model)

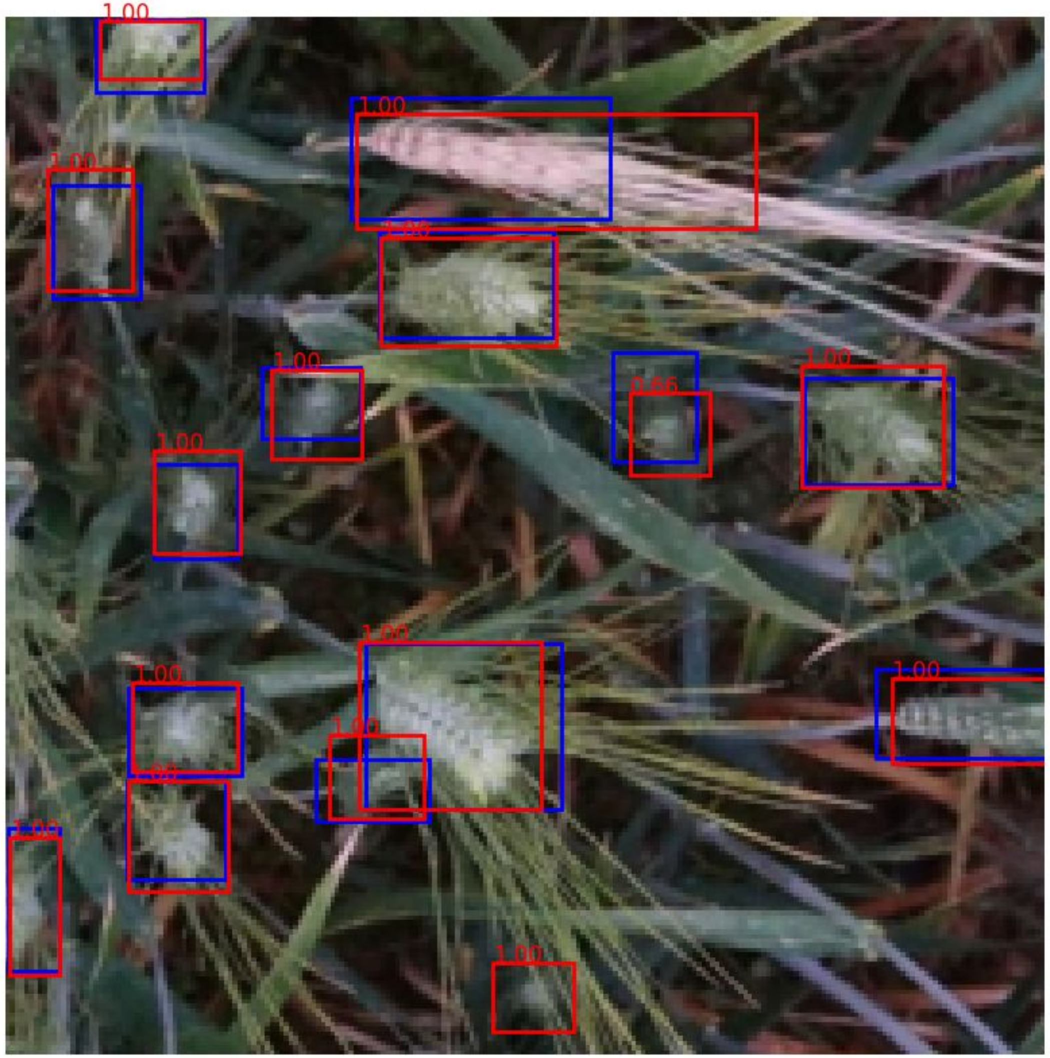




anchor_sizes	12, 24, 36
aspect_ratios	0.5, 0.67, 1.0, 1.5, 2.0
batch_size	2
AdamW weight_decay	0.0001
early_stopping_patience	10 epochs (watching val_mAP)

Best Model Results





Conclusion - Outlook

- **Results - Best Model**
  - Backbone: ViT Base Patch 16 224
  - Weights: PlantNet-300k
  - Full augmentation applied: Rotation, Flip, Normalization and Random Crop
  - Best validation mAP = 0.623210
  - Best private test mAP = 0.5363
  - Best public test mAP = 0.6261

- **Limitations**
  - Faster R-CNN is slower than single-shot detectors (SSD, YOLO)
  - Computationally pretty intensive

- **Outlook – What should we do better next time**
  - Check for more data (even on Kaggle) and with other competitors
  - Use Backbone with larger input size
  - Do research: reading papers

Dataset Acknowledgement/ References

- Global Wheat Detection Dataset. (2020). Retrieved January 09, 2025, from https://www.kaggle.com/competitions/global-wheat-detection
- Najib, M. R. N. B. F. (n.d.). *ViT-PlantNet300K* [Machine learning model]. Hugging Face. Retrieved February 01, 2025, from https://huggingface.co/janjibDEV/vit-plantnet300k
- Gad, A. F., & Skelton, J. (2024). *Tutorial: Faster R-CNN explained for object detection tasks*. DigitalOcean. Retrieved February 01, 2025, from https://www.digitalocean.com
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). *Faster R-CNN: Towards real-time object detection with region proposal networks*. arXiv. https://doi.org/10.48550/arXiv.1506.01497
- He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). *Mask R-CNN*. arXiv. https://doi.org/10.48550/arXiv.1703.06870

Thank you for your attention

Lukas Julius Eule & Janine Berndt