Strawberry disease instance segmentation

Bruno Kreiner

28. Mai 2023

Zusammenfassung

TASK: Zusammenfassung der Fragestellung, Datenlage, ML Approaches und wichtigsten Resultate inkl. Kennzahlen. Ca. 200 Wörter.

This mini-challenge focuses on the task of detecting and segmenting seven different kinds of strawberry diseases found on leaves, fruit, and flowers using instance segmentation. A Mask R-CNN model, as referenced in a previous paper [ABPL21], served as a benchmark but could not be reproduced due to package dependency problems. Without any modifications, a YOLOv8 segmentation model immediately surpassed the baseline Mask R-CNN model by around 10% in terms of mAP. Marginal improvements were seen in the YOLOv8 model through fine-tuning, changing the learning rate, adjusting the optimizer, and selecting the largest segmentation model available on ultralytics.com. However, challenges persist, such as ambiguity in annotating overlapping instances of disease clusters and potential overfitting. Despite these, the evaluation in the notebooks showcases opportunities for further optimization of the YOLOv8 model.

1 Fragestellung

TASK: Beschreibt die Ausgangslage und Anwendungsdomäne eurer Fragestellung. Formuliert dazu explizit Hypothese(n), Forschungsfragen oder Ziele.

The food industry faces a number of challenges in today's world. On one hand, increasing populations create a higher demand for food supplies and on the other hand, climate change leads to extreme weather conditions and diseases impacting food production. Innovative technological advances in the field of food production, packaging and food processing as well as ways to detect and handle diseases are necessary more than ever. In this project, the automatic detection of 7 different strawberry diseases will be analyzed using machine learning. This can help identify diseases faster and in an efficient way. The data was collected by the members of the AI lab, Computer Science and Engineering department, JBNU. It was labeled using the LabelMe website for an instance segmentation task. This means, for each object (disease) in each image, there exists a bounding box and an exact segmentation mask of that object. JBNU also released a paper [ABPL21] where they tested Mask R-CNN [HGDG18] with ResNet [HZRS15] backbones and different data augmentation techniques. They used mAP@50 as the metric. For mAP@50, predictions only with an intersection over union (IoU) of 0.5 are considered.

The following list shows this project's scope and goals:

- Task: Instance segmentation on strawberry plant images
- Goal: Correctly identify, localize strawberry disease in each image and place segmentation masks for each disease.
- Research question: How can the performance of baseline model (as per this reference paper) be improved by adapting certain parts of the Mask R-CNN architecture (i.e. switching backbones and other layers in the Mask R-CNN architecture)? Can we further fine-tune hyperparameters or use more data augmentation? Is there another instance segmentation model that achieves a better mAP@50?
- Hypothesis: We can improve the mAP@50 by adjusting the Mask R-CNN architecture or using a different model.

The baseline models in [ABPL21] achieved a 72.06% and 71.69% mAP@50 with Resnet-50 and Resnet-101 respectively. Using data augmentation and an improved training strategy, these scores improve to 81.37% and 82.43%.

2 Datenlage

TASK: Wählt einen Datensatz mittels welchem die Fragestellung gelöst werden kann. Es können Bilder (2D, 3D), Videos, Signale aus Sensordaten, Punktewolkendaten, Audiodaten oder ähnlich verwendet werden. Es kann ein öffentlich verfügbarer Datensatz, ein eigener Datensatz sowie ein Datensatz aus einem Geschäftsbetrieb sein. Die Daten müssen nicht mit den DS Fachex-pert:innen geteilt werden, sofern der Lösungsweg nachvollziehbar ist. Da wir in dieser Mini-Challenge ein supervised Problem lösen, sollen Ground Truth Daten bzw. Labels vorhanden sein oder mit geringem Aufwand generiert werden können. Bemerkung: Für diese Mini-Challenge kann mit einem «zu

kleinen» Datensatz gearbeitet werden. In einem solchen Falle, soll dieser Punkt entsprechend diskutiert werden.

The strawberry disease dataset is a high quality dataset where most images are close ups of individual leaves and fruits. It is available at Kaggle [Afz21]. It consists of 2500 images in total with corresponding segmentation annotation files (one per image) for seven types of diseases found in Strawberry plant. The diseases can be seen in figure 1. The data was collected from multiple greenhouses under natural illumination conditions in South Korea and the diseases were verified by experts. The images were processed to 419 x 419 resolution. The dataset is split into 1450, 307 and 743 images for training, validation and test sets. For image there exists an individual json file with a list of ground-truth annotations. Each annotation has the label name as a string and a list of points with x and y coordinates that span the segmentation mask polygon.

Abbildung 1: Figure 2 from [ABPL21] showing the seven types of strawberry diseases.

Visualizing class distribution of the training data, we can see a rather big class imbalance in figure 2.

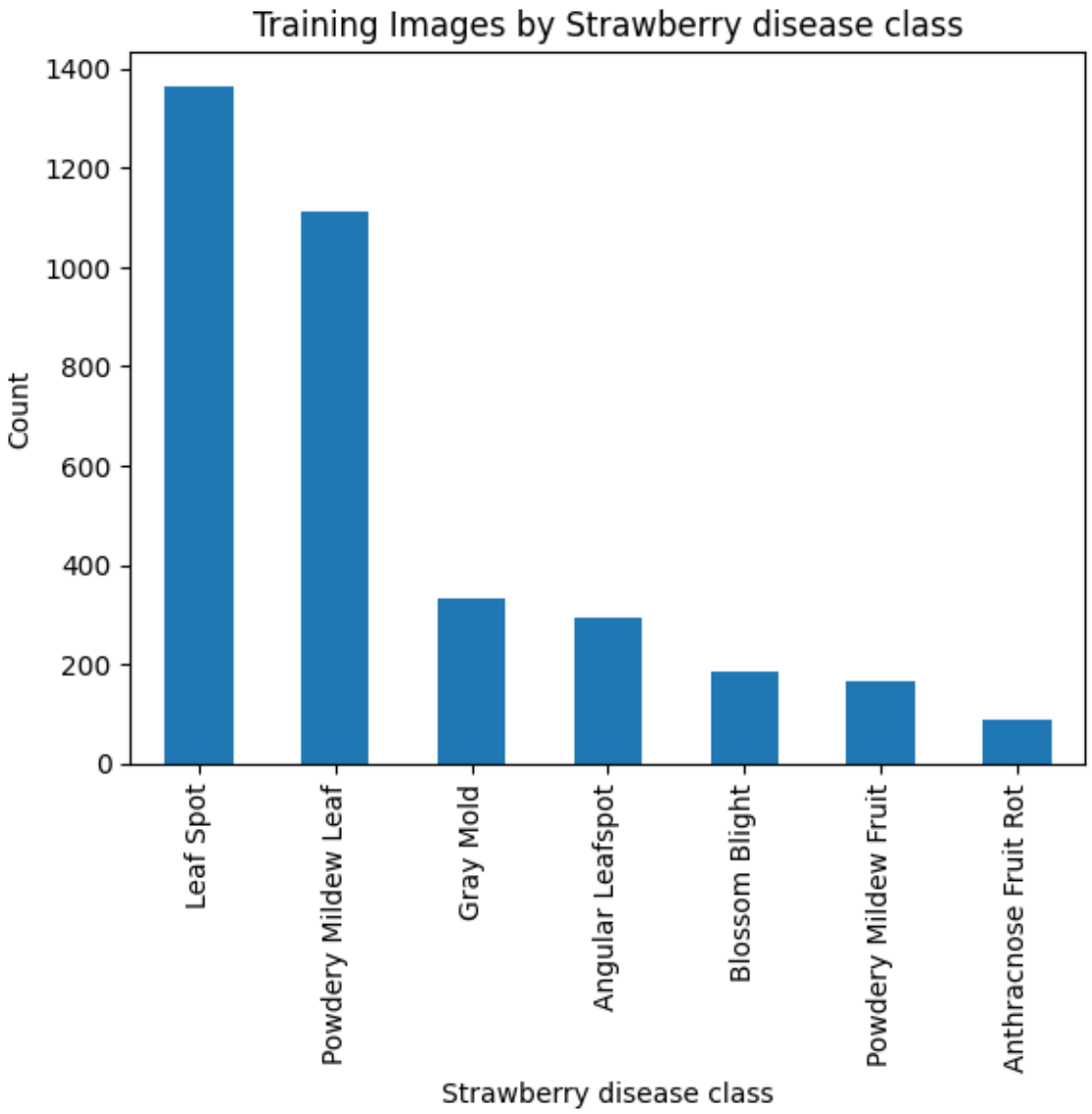

Abbildung 2: Training data class distribution.

This doesn't match with the performance of the baseline Mask R-CNN from [ABPL21] in figure 3 where overrepresented classes don't acheive a higher average precision performance. This is probably due to the inherent difficulty in predicting certain diseases. The most frequent class is "Leaf Spot"yet achieves a low average precision, while "Gray Mold"has the highest score and looks to be easier to identify, probably due to its color.

Class	AP for ResNet50 (%)	AP for ResNet101 (%)
Angular Leafspot	79.93	**81.16**
Anthracnose Fruit Rot	**71.46**	63.63
Blossom Blight	**87.90**	82.25
Gray Mold	92.29	**93.90**
Leaf Spot	71.93	**73.33**
Powdery Mildew Fruit	68.02	**70.91**
Powdery Mildew Leaf	85.66	**89.87**

Abbildung 3: Table 8 in [ABPL21] showing Average precision per class for the fine-tuned models.

3 Methoden/Vorgehen

TASK: Beschreibt und diskutiert die Wahl eurer Lösungsansätze (Baseline, DL Architekturen, Evaluation) gemäss eurer Fragestellung. D.h., warum habt ihr was WIE gemacht? Ca. 300-500 Wörter. Die folgenden Teilkapitel sollen direkt in einem Notebook beantwortet werden.

The project tries to follow the "Recipe for Training Neural Networks"by Karpathy [Kar19] shown in figure 4. Therefore, the first steps are data inspection and the training of a simple baseline. We take the Mask R-CNN model from [ABPL21] and try to reproduce it. To achieve similar model performance, a prominent Mask R-CNN Github repository was chosen [Abd17] with over 23 thousand stars. While this seemed like a simple task, the repository hasn't been updated in over 4 years and there were dependency problems with mismatched Keras and Tensorflow versions. The repository also has over 1.8 thousand open issues. For simplicity, a PyTorch implementation of Mask R-CNN was used and can be inspected in Model Testing Notebook where input data is visualized and overfitting is tried with 10 training samples. The hyperparameters from the existing paper were copied. The learning rate is set to 0.0001, the batch size to 2, the optimizer is Stochastic Gradient Descent, the whole model was trained (Resnet isn't frozen) and the images are set to 512 x 512. The paper sets 512 x 512 as the minimum image size and 960 x 960 as the maximum, but the maximum size is not required to be set since all images in the data set are 419 x 419. The one thing added was mixed precision training using PyTorch's integrated functions to improve computational power. One thing not used in this project's training was the improved training strategy mentioned in the paper. The images were not normalized prior to testing the baseline since the paper doesn't mention it. In Model Testing Notebook, the model fails to learn from the shortened training set and this issue remains when trying different learning rates. The evaluation shows that the model predicts too many bounding boxes for some images and none at all for others. The model also begins to predict no bounding box for every image after some time. Setting up the PyTorch training notebook took a lot of time and even the model trained with all the data and 50 epochs couldn't get past 20% mAP@50. This can be seen in Model Full Training. Here, a gradient accumulator was used to free up some memory on the GPU. Different learning rates were tested. For smaller learning rates, the model doesn't learn fast enough and after 50 epochs it is under 20% mAP@50. For bigger learning rates, it stops training after 20-30 epochs and stays at just above 20% mAP@50.

To not lose too much time debugging the code, other models were tried in the meantime. To choose a fitting model, the best model in the COCO instance segmentation ladderboard was chosen. Tested on the COCO test set, EVA achieved the best average precision. EVA is a large vision model that can be used as a base model for multiple tasks. Following their installation guide, the correct packages had to be installed. After failing locally with a new environment and debugging, a dockerfile was set up for further testing. This didn't seem to fix the problem even when making sure to have the correct Cuda and PyTorch versions installed for EVA and its dependencies on Detectron2 (a library by Facebook AI Research (FAIR) with implementations of multiple state-of-the-art models). The issue is shown in the notebook here. The same issue came up trying out the model Mask DINO due to the same dependencies on Detectron2.

As a last resort, YOLOv8 [JCQ23] was tried after discovering Roboflow, a website to streamline data processing and vision model development. Roboflow lists multiple notebooks for vision tasks in their repository. They work by downloading a dataset uploaded to Roboflow which it automatically converts to the right data format. In the strawberry dataset case, Roboflow automatically converted the LabelMe formatted data into YOLOv8 format. Roboflow also came with errors. On the Firefox browser, uploading too much data at once led to errors. On Google Chrome it worked better but the training and test set ended up with duplicated data. This problem wasn't detected until the YOLOv8 models were already trained. The validation set had no duplicates and this is what the evaluation focuses on. The duplicated data probably had a very minor effect on the training of the YOLOv8 models. YOLOv8 provides a lot of different models. For this project, the small instance segmentation (yolov8s-seg.pt) model was chosen. For YOLOv8 the remaining tasks from the "Recipe for Training Neural Networks"can be completed. Meaning overfitting, regularization, fine-tuning and ßqueezing the juice out of it".

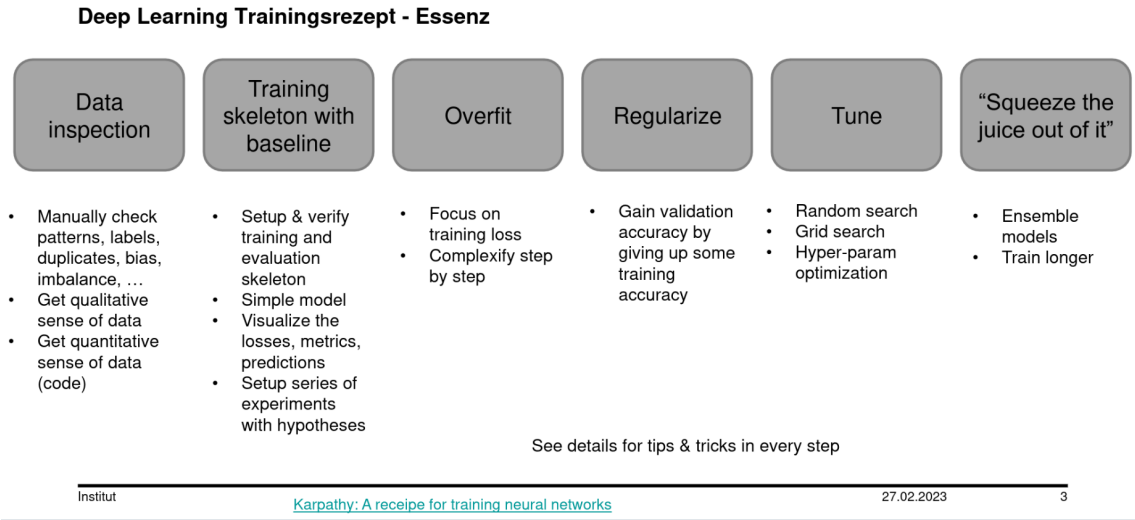

Abbildung 4: DLBS deep dive slide showing Karpathy's guideline for model training.

3.1 Datenanalyse

TASK: Führt eine Datenanalyse anhand dieser drei Schritte durch [?]:

- *Prüft die Daten mittels visueller Inspektion*
- *Erfasst ein qualitatives Verständnis der Daten*
- *Analysiert die Daten quantitativ (mittels Code)*
- *Befolgt für die Datenanalyse Richtlinien von [Lon23] und diskutiert pro Person 1 Richtlinie:*
  - *Do take the time to understand your data*
  - *Don't look at all your data*
  - *(Do make sure you have enough data)*
  - *Do talk to domain experts*
  - *Do survey the literature*
  - *Do think about how your model will be deployed*
  - *Don't allow test data to leak into the training process*

Guideline for data analysis (1 per person):
to-do: Describe the following wrt our dataset and analysis applie

- Do survey the literature
- Do take time to understand your data

3.2 Setup eines Training-Evaluations-Code-Skeletts inkl. Baseline

TASK:

- *Richtet ein Code-Skelett für die Trainings-Evaluations-Zyklen ein*
- *Wählt ein einfaches Modell als Baseline, geeignete Metriken für die Evaluation der Netzwerke und diskutiert die Wahl inkl. Parameter. Die Baseline soll trainiert/generiert, analysiert und visualisiert werden (Loss, Predictions etc.).*
- *Definiert pro Person 1-2 Ablationstests zu den Hypothesen/Fragestellungen, welche durchführt und verifiziert werden.*
- *Befolgt und diskutiert pro Person 1 ausgewählter weiterer Aspekte aus [?]*
- *Befolgt zudem für das Trainings-Evaluations-Skelett Richtlinien von [Lon23] - diskutiert pro Person jeweils 1 Richtlinie:*
  - *Do use an appropriate test set*
  - *Do use a validation set*
  - *Do save some data to evaluate your final model instance*
  - *Don't use accuracy with imbalanced data sets*
  - *Do evaluate a model multiple times*
  - *Don't do data augmentation before splitting your data*
  - *Don't allow test data to leak into the training process*

3.3 Overfit

TASK:

- *Fokussiert auf den Loss des Trainingssets*
- *Fügt schrittweise zusätzliche Komplexitäten ein und dokumentiert diese*
- *Wählt pro Person individuell eine Modell-Architektur zur Lösung eurer Fragestellung aus und führt ein Overfitting gemäss [?] durch. Wählt pro Person jeweils 2 unterschiedliche Aspekte von [?] aus, welche ihr implementiert bzw. durchführt und diskutiert.*
- *Befolgt für das Overfitting Richtlinien von [Lon23] und besprecht pro Person 1 Richtlinie:*
  - *Don't use inappropriate models*
  - *Do try out a range of different models*
  - *Do evaluate a model multiple times*
  - *Don't do data augmentation before splitting your data*
- *Optionale Bonusaufgabe: Vergleicht und evaluiert eine zusätzliche DL Architektur*

3.4 Regularize

TASK:

- *Gewinnt an Validierungsgenauigkeit durch Verzicht auf gewisse Trainingsgenauigkeit*
- *Arbeitet pro Person individuell mit eurem Modell weiter und führt pro Person 2 Regularisierungsexperimente gemäss [?] durch*
- *Vergleicht eure Modelle in geeigneter Weise und befolgt dabei die Richtlinien von [Lon23] - diskutiert pro Person 1 Richtlinie:*
  - *Do evaluate a model multiple times*
  - *Don't assume a bigger number means a better model*
  - *Do use statistical tests when comparing models*

  - *Do correct for multiple comparisons*
  - *Don't always believe results from community benchmarks*
  - *Do be transparent*
  - *Do report performance in multiple ways*
  - *Do be careful when reporting statistical significance*
  - *Do look at your models*
- *Wählt eines eurer Modelle als bestes aus und diskutiert die Entscheidungsfindung*

3.5 Tune

TASK:

- *Fährt nun gemeinsam mit dem ausgewählten Modell fort. Und führt eine der folgenden Tuning-Varianten durch.*
  - *Random search*
  - *Grid search*
  - *Hyper-param optimization*
- *Befolgt für das Tuning Richtlinien von [Lon23] - diskutiert gemeinsam 1 Richtlinie:*
  - *Do evaluate a model multiple times*
  - *Do optimise your model's hyperparameters*
  - *Do be careful where you optimise hyperparameters and select features*

3.6 Optionale Bonusaufgabe: Squeeze the Juice Out of It

TASK:

- *Wählt eine Methode aus dem Bereich:*
  - *Ensemble models*
  - *Train longer*
- *Befolgt dafür Richtlinien von [Lon23] - diskutiert gemeinsam 1 Richtlinie:*
  - *Do consider combinations of models*
  - *Do evaluate a model multiple times*

4 Wichtigste Resultate

TASK: Beschreibt eure wichtigsten Resultate in ca. 400-500 Wörtern. Verwendet ggf. Abbildungen. In this project, various configurations of the YOLOv8 model were evaluated for an instance segmentation task. For chapter 3, a Mask R-CNN skeleton was trained. As already discussed, the Mask R-CNN model didn't learn and the results of the baseline Mask R-CNN model from existing research couldn't be reproduced. Using YOLOv8 as a backup, a 10% mAP improvement could be achieved over the baseline from the existing paper. Different kinds of YOLOv8 models were trained over 100 epochs. These models show a steady decrease in training loss for box prediction, segmentation and class prediction. The base model already performs very well at around 92-93% mAP50. Additional data augmentation techniques can potentially decrease performance due to YOLOv8's inbuilt data augmentation. The "Base XL"performed the best on the validation data. It was trained using YOLOv8's XL model. All other models, which come very close to it, were trained using YOLOv8's small model. A test run with a smaller learning rate (factor of 10) and full augmentation strategy based on the augmentation steps from the existing research follows the "Base XL"closely while some other models actually worsen with data augmentation. This can be seen in figure 5.Using the same data augmentation, using the AdamW optimizer or setting the pretrained parameter in YOLOv8 to true also improved results. The existing paper uses Ëdge Detectänd

"Color Enhancement"for training their best Mask R-CNN model. The direct implementation of those augmentations were not found in common augmentation libraries such as PyTorch or Albumentations. Therefore, Ëdge Detect"was left out and "Color Enhancement"was replaced with "RGB Shift". Interestingly, smaller learning rates led to faster learning, contradicting expectations. Just changing hyperparameters or using the base model with data augmentation didn't lead to improved training results. The use of dropout (regularization) improved the validation mAPs by almost 1% in comparison to its baseline. Despite validation loss not being tracked correctly by YOLOv8, results indicate possible overfitting since dropout clearly improved results. Furthermore, a spike in training loss at the 90th epoch was observed, without any discernible impact on the validation metrics. The confusion matrices revealed that most false positives or negatives were for the background class, which is also shown in the existing research for the Mask R-CNN. Manual evaluations showed accurate box and class predictions, with some limitatrions in detailed mask placement. It occasionally struggled to create detailed and accurate masks, especially for diseases appearing in clusters. This might be due to a misplacement of bounding boxes in the data set by researchers using the LabelMe website since for one cluster, multiple boxes can be placed on individual disease spots. In summary, this project demonstrated the robust capabilities of YOLOv8 in instance segmentation tasks, with potential further improvements through careful manipulation of learning rates, dropout and other hyperparameters. It highlights the importance of understanding in-built data augmentation functions and their impact on model performance. A more detailed description is in the notebook "yolov8_evaluation.ipynb".

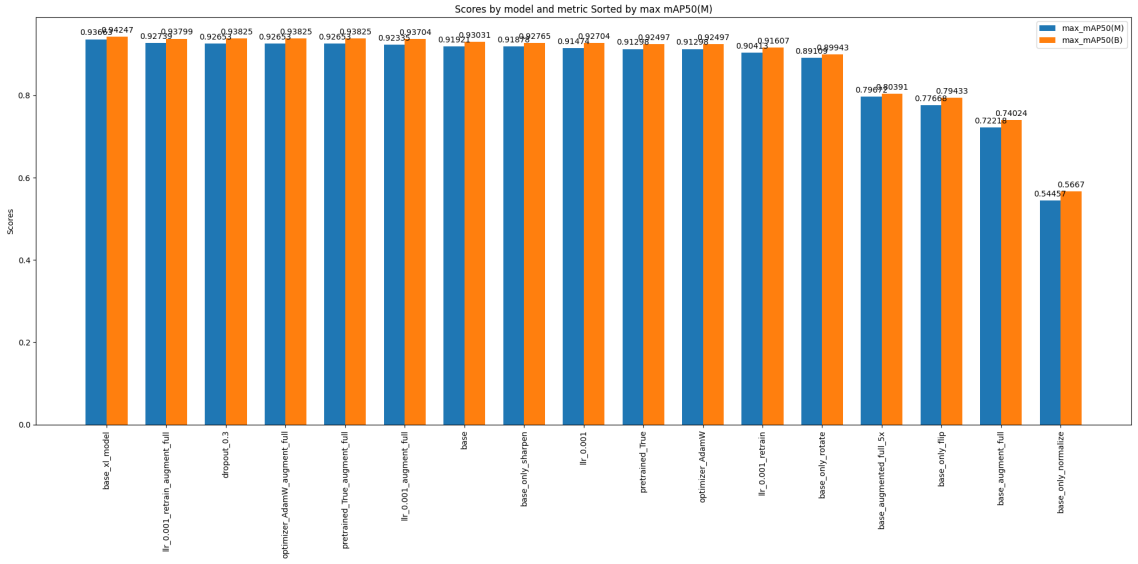

Abbildung 5: Bar Plot of mAP50 performances for YOLOv8 models

5 Diskussion

TASK:

- *Diskutiert eure Resultate explizit in Bezug auf eure Fragestellung(en) in ca. 500-600 Wörtern. Mit Diskutieren ist gemeint, Vor- und Nachteile, Chancen und Risiken eures Lösungsweges bzw. der Resultate spezifisch für euren Anwendungsfall zu besprechen. Wie sind die Resultate der Baseline im Vgl. zu den optimierten Modellen? Welche Fälle funktionieren gut, welche nicht? Lässt sich die Fragestellung mit den gewählten Modellen und Daten beantworten?*
- *Befolgt für die Diskussion die Richtlinien von [Lon23]:*
  - *Don't generalise beyond the data*
  - *Don't always believe results from community benchmarks*
  - *Do be transparent*

Using YOLOv8 as a replacement for Mask R-CNN proved to be a good decision in terms of mAP performance. The disadvantage of this method is that it's not completely clear how YOLOv8

works in detail. The source code is open on GitHub, yet it requires a lot of work to understand it and be able to really tune the model using custom code. The goal of this project was to use Mask R-CNN and go into detail regarding model architecture. Due to package problems, this problem couldn't be tackled. Modifying the model architecture of YOLOv8 would have been an appropriate task following these problems, but it wasn't feasible with the time left for this project. Since the model architecture is publicly available, future improvements could look into modifying this model architecture configuration file.

The advantage of using Roboflow for the dataset and YOLOv8 is that it is extremely fast to train a good model and bring it into production. These models have been studied a lot and the simplest model is already really good. Roboflow's automatically trained model also beats the Mask R-CNN model from previous research. The risk there is that researchers don't have the option to modify each module inside the YOLOv8 model since everything is predefined. For casual users this is a positive. Tuning YOLOv8 might not even be the best approach since these models have already been optimized. Maybe the correct approach is to have the best possible data quality possible. This project looked at very high-quality data where experts verified the diseases. Even then, clear annotation rules and guidelines should be placed so that every bounding box is distinct and makes sense. This way, we can make sure models train effectively. The models are trained on this specific data "type". If someone takes a picture of a plant from a bigger distance, the model might not recognize any disease, since it was trained on mainly close-up pictures. For production, this needs to be taken into account.

Finally, not every open source project on GitHub is easily implemented and some require a lot of labor to make it work. Also, the existing paper as well as the YOLOv8 documentation leave out some details for interpretation. Mainly the data augmentation wasn't clear in the paper and for YOLOv8, there is no documentation part about how exactly images are loaded and if data augmentation is done for YOLOv8 models by default. The YOLO team has been under critique for not releasing more documentation and I had to ask questions in their community forum to get more insight into the model.

6 Reflexion

TASK: Reflektiert abschliessend in ca. 150-200 Wörtern. Was sind die Schlussfolgerungen für die spezifische Fragestellung? Was ist in dieser Mini-Challenge gut gelaufen? Was würdet ihr das nächste Mal anders machen? Was würdet ihr ggf. in der Aufgabenstellung ändern? Fokus auf fachlicher Reflexion.

This mini-challenge had its difficulties. Since I was alone, I had to find my own solutions and sometimes I didn't have a guiding hand and wasted some time trying to make other models work. I also had the interim presentation for my bachelor's thesis in the meantime and was exhausted. For the chapter 3 tasks specifically, I didn't know how much to write in the notebooks and if the things I wrote needed to be included in here.

All in all, I think this project was a success since the YOLOv8 model outperforms the baseline by a significant margin. I also gained some insights into the models. Be it by looking into source codes and trying to install everything (I did learn about docker again) or by evaluating the training of my models. I'm looking forward to more projects in ML for vision and some barriers I saw as daunting have been shattered.

7 Code

TASK: Zugang zu aufgeräumtem und mit README versehenem Git-Repository. Die Abgabe darf keine auskommentierten Codestellen enthalten (ausser pip install statements). Bemerkung zu Code Kommentaren: Code Kommentare sind hilfreich, wenn sie erklären, weshalb eine Entscheidung getroffen wurde oder weshalb eine Rechnung in der bestimmten Art und Weise ausgeführt wurde oder gemäss welcher Quelle etwas implementiert wurde. In der Praxis wird jedoch bei der Code-Entwicklung oft vergessen, die Code-Kommentare anzupassen. Deshalb sind Kommentare oft obsolete. Auskommentierter Code wird in der Praxis typischerweise nicht getestet und später weiss man nicht mehr, ob dieser noch funktioniert oder nicht. Die Empfehlung von mir lautet deshalb: So viele Kommentare wie nötig, aber so wenige wie möglich. Grundsätzlich sollte der Code so geschrieben sein, dass er selbsterklärend ist.

8 Optionale Bonusaufgabe: Lerntagebuch

TASK: Optional darf der Abgabe ein Lerntagebuch beiliegen, welches regelmässig dokumentiert, wie der Lernfortschritt war. Bspw. kurz ein paar Fragen beantworten, analog zu einem Scrum Daily. Was hast du an diesem Tag gemacht? Was ist gelungen? Wo gibt es aktuell Probleme? Wer könnte bei diesen Problemen helfen?

8.1 4/17/2023 10:00 - 13:00

Goal: Explorative data analysis

Üploaded the data on CSCS and started looking at the images including labels. Initial plots: - Number of images in train, val and test set - pixel size - types of diseases as labels in json files - number of images per disease type

There are in total 2500 images available for seven types of diseases seen in Strawberry plant, fruit and flower. All images are of size (419,419) with 3 color channels. Training images: There are 1450 images. Images for seven types of diseases are different in count . The same is true in validation set which has 743 images. This must be kept in mind and mentioned in report."

8.2 4/19/2023 10:00-11:45

Goal: 1. Draft version of Pitch slide 2. proposal on additional evaluation criteria 3. If bonus task, then plan(what how) 4. Mini-challenge tasks and goals per week to be achieved"

We discussed in a short call the problem statement including hypothesis, methods, and evaluation criteria for pitch. Navjot agreed to create a draft presentation. Later that day, the draft version was uploaded on OneDrive for me Bruno to review. We agreed to meet in person before pitch presentation and discuss on next steps

1) Challenge 1: Evaluation criteria should be measurable. One option is that we plan mini-challenge in sprint form with backlog for next 5 weeks. But sprint planning in this context doesn't make much sense as we have clear steps to follow such as using best practices from Karpathy. We can't just pick any task in a given week. What would make sense for us, is to target one task per week and then meet at the end of the week to review/retrospect 'what went well' and 'what could have been done differently'. Risk here is that if we miss targets for 2 weeks, it will have an impact on evaluation we defined.

We agreed to use the evaluation critieria around weekly goals and review/retrospective

8.3 4/25/2023 12:15-16:00

Goal: Pitch presented next steps for this week agreed

1) One Slider pitch was presented with information on Data, Research question and methods we plan to apply. Our approach would be to take the linked paper and choose one of the model as baseline. Each of us will work on another model variation, which is still to be decided. 2) Research on Mask R-CNN model, paper from kaggle on approach followed and which models were compared 3) Read about instance segmentation 4) The challenge was to keep our focus on simple models and also try to change some Mask R-CNN internal architecture to study the model. We can then choose a completely different model and compare it to our Mask R-CNN evaluation and fine-tuning.

8.4 4/28/2023 14:15-15:30

Goal: Start documenting the report on Overleaf: Research question, Starting situation, About data

Documented the task at hand and its application in the area of research as well as questions which we would like to address. About data: what data available, how many images and labels Updated references: Kaggle dataset link as well as reference paperFirst time working with overleaf takes some time to get used to it.

8.5 5/2/2023 14:30 - 16:15

Goal: Understand Skeleton model for mini-challenge

Today, we discussed the first version of Mask RCNN model skeleton on PyTorch which runs but is very slow. It runs on kaggle with GPU support but not yet on our local machine (Linux Mac). The documentation of PyTorch Mask RCNN specifies that the model expects input data and labels in a specific struture. This is now in place but it has to be improved including transformation. First training epochs for Model was slow on Kaggle. We need to figure out about mean Average precision and if we manage to achieve it in the given time. Another challenge faced was about the data structure: the model expects input images and labels as a list of dictionary where each dictionary contains the keys "boxesänd corresponding value as list of bounding boxes for that image. Similar dictionary for "labelsänd "masks"

8.6 5/2/2023 16:15 - 17:00

Goal: Weekly progress review in group and retrospective as to what went well, what we could improve and what we need to achieve next.

We went through the list of tasks for this week and documented our progress in seperate report. (see commits on GitHub - pdf document). We thought it would take around 15 mins but as we started our discussion, there were quiet few points to be addressed looked at for next week. This is documented in weekly progress report.

Literatur

[Abd17] Waleed Abdulla. Mask r-cnn for object detection and instance segmentation on keras and tensorflow. https://github.com/matterport/Mask_RCNN, 2017.

[ABPL21] U. Afzaal, B. Bhattarai, Y.R. Pandeya, and J. Lee. An instance segmentation model for strawberry diseases based on mask r-cnn, 2021.

[Afz21] U. Afzaal. Strawberry disease detection dataset, 2021.

[HGDG18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018.

[HZRS15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[JCQ23] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. YOLO by Ultralytics, January 2023.

[Kar19] Andrej Karpathy. A recipe for training neural networks, Apr 2019.

[Lon23] Michael A. Lones. How to avoid machine learning pitfalls: a guide for academic researchers, 2023.