Plant Detection and State Classification with Machine Learning

Tobias Eidelpes

March 12, 2024

Problem Statement

- Automated detection of water stress

Problem Statement

- Automated detection of water stress
- Automated watering of household plants

Problem Statement

- Automated detection of water stress
- Automated watering of household plants
- Decision-making *in the field*

Problem Statement

- Automated detection of water stress
- Automated watering of household plants
- Decision-making *in the field*
- No research so far in this context

Research Questions

1. How well does the model work in theory and how well in practice?

Research Questions

1. How well does the model work in theory and how well in practice?

2. What are possible reasons for it to work/not work?

Research Questions

1. How well does the model work in theory and how well in practice?

2. What are possible reasons for it to work/not work?

3. What are possible improvements to the system in the future?

Methods

1. Literature Review
2. Dataset Curation
3. Model Training
4. Optimization
5. Deployment
6. Evaluation

Prototype Design: Requirements

Prototype Design: Requirements

- Detect and Classify

Prototype Design: Requirements

- Detect and Classify
- Publish Results via REST-API

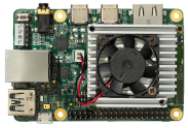
Prototype Design: Requirements

- Detect and Classify
- Publish Results via REST-API
- Reasonable Inference Time

Prototype Design: Requirements

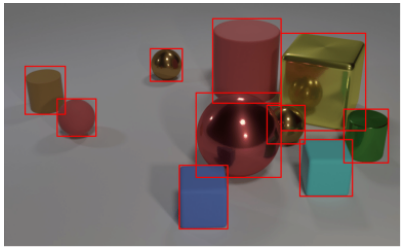
- Detect and Classify
- Publish Results via REST-API
- Reasonable Inference Time
- Reasonable Model Performance

Prototype Design



Input Images

Object Detection

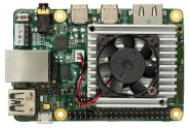


No

Water Stress?

Yes

State Classification



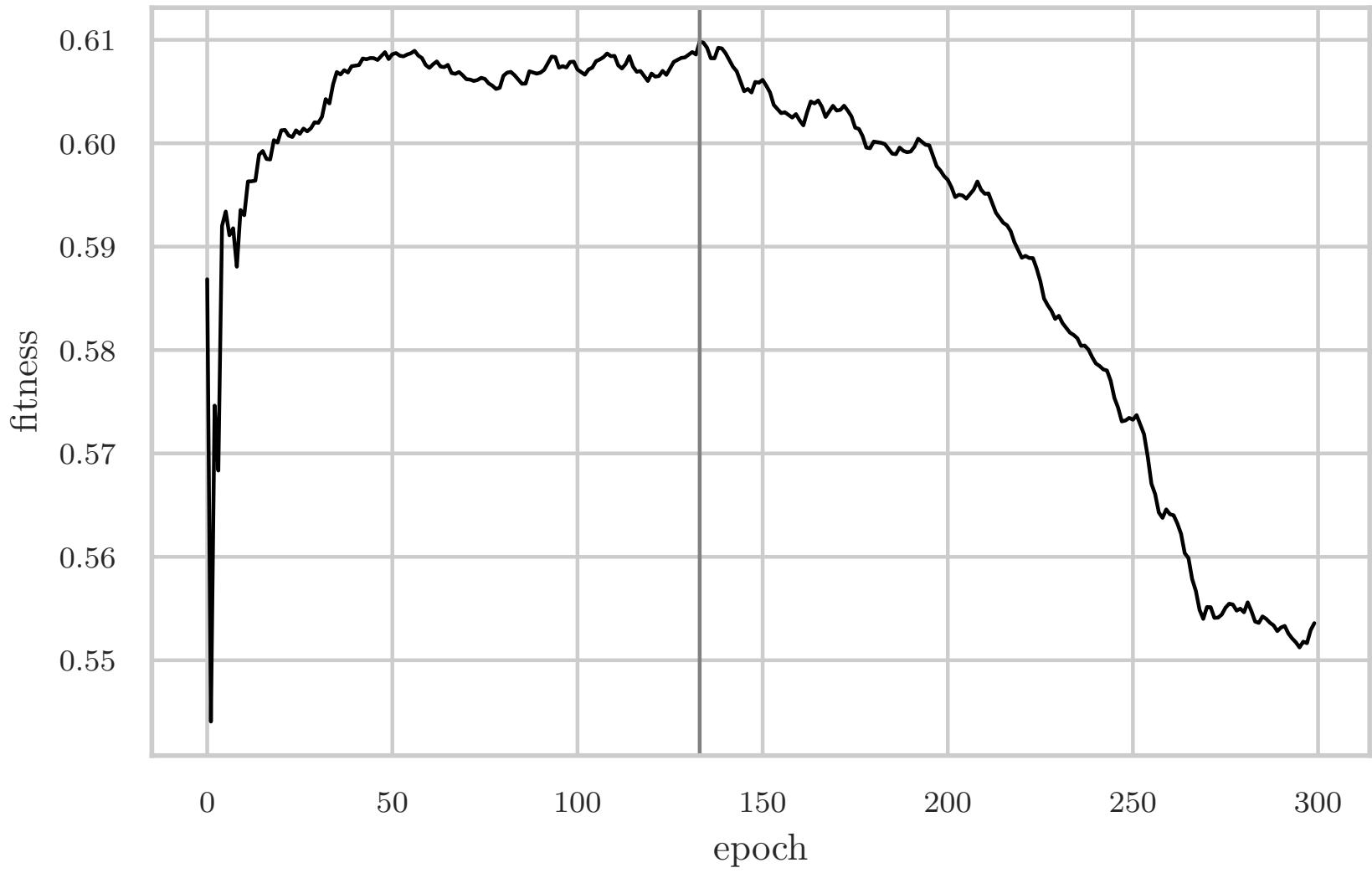
Recommend Watering

Prototype Implementation: YOLOv7n

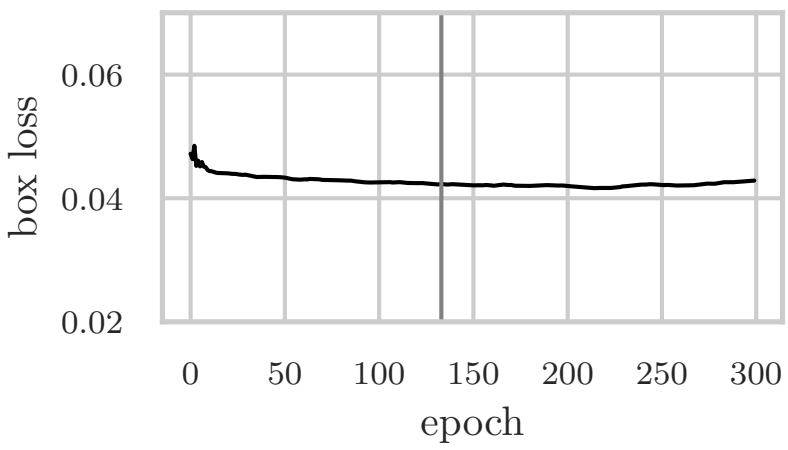
- Pretrained on COCO
- OID classes *Houseplant* and *Plant*
- Training Set
  - 79 204 images
  - 284 130 bounding boxes
- Validation Set
  - 3091 images
  - 4092 bounding boxes

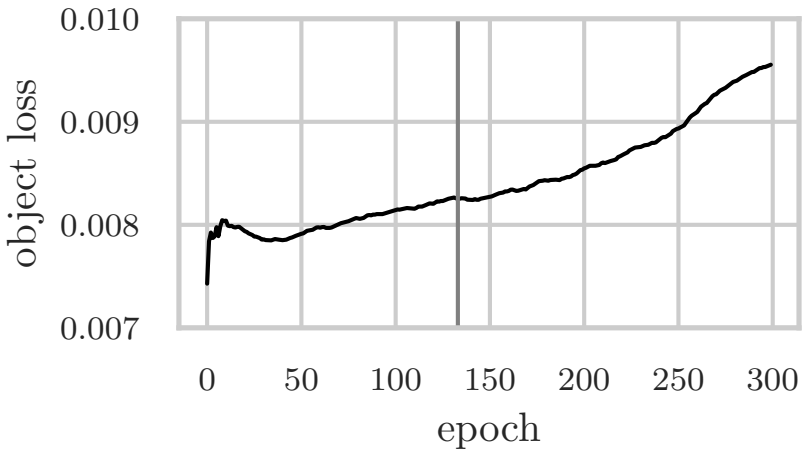
Earthy Tones For Fallsurlevif by Flickr User decor8 under CC BY 2.0

Prototype Implementation: YOLOv7n



Prototype Implementation: YOLOv7n





YOLOv7n Hyperparameter Optimization

YOLOv7n Hyperparameter Optimization

- Mutate 26 out of 30 hyperparameters

YOLOv7n Hyperparameter Optimization

- Mutate 26 out of 30 hyperparameters
- Parent chosen randomly from top five previous generations

YOLOv7n Hyperparameter Optimization

- Mutate 26 out of 30 hyperparameters
- Parent chosen randomly from top five previous generations
- 3 epochs per iteration

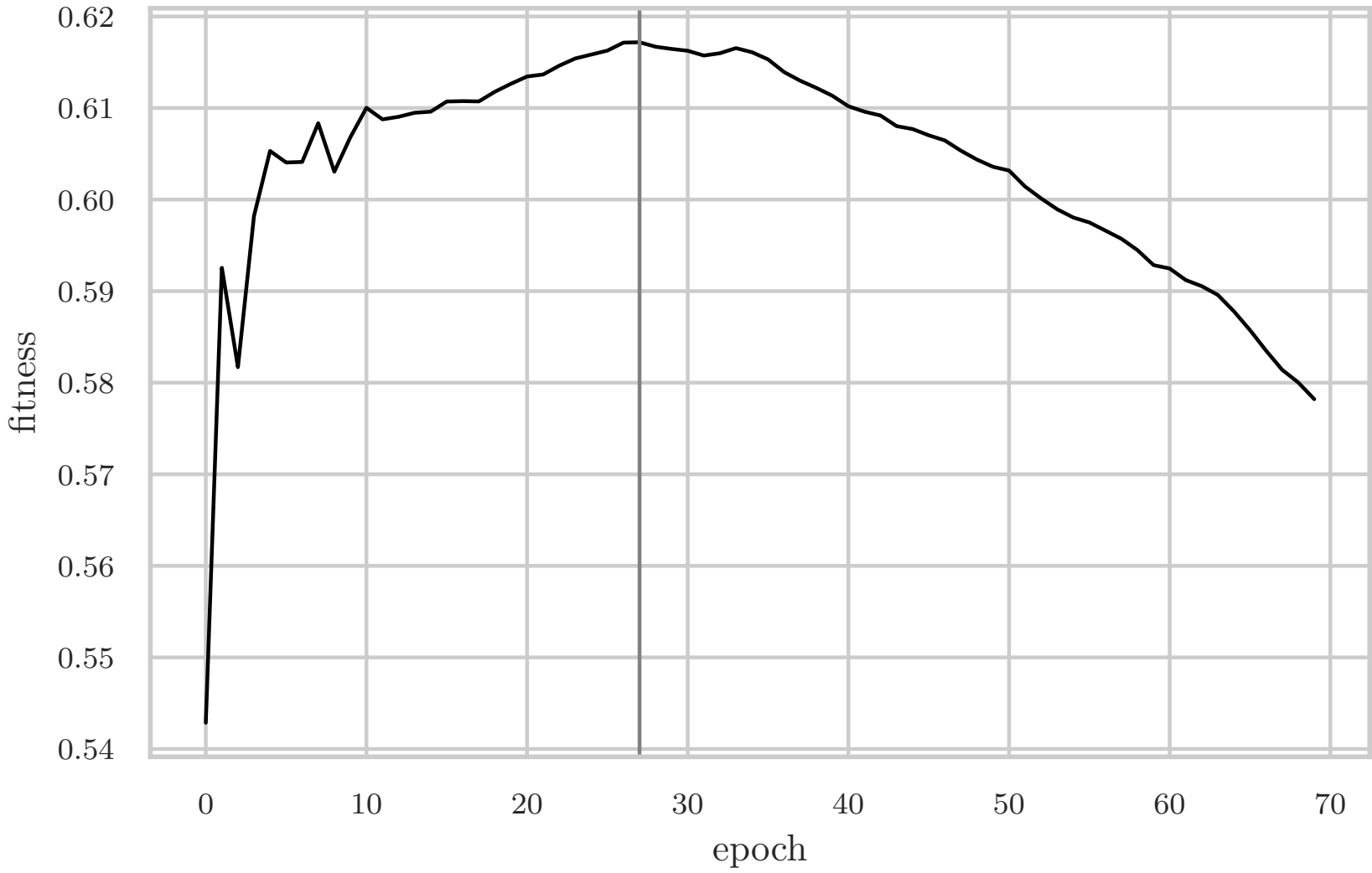
YOLOv7n Hyperparameter Optimization

- Mutate 26 out of 30 hyperparameters
- Parent chosen randomly from top five previous generations
- 3 epochs per iteration
- 87 iterations

YOLOv7n Hyperparameter Optimization

- Mutate 26 out of 30 hyperparameters
- Parent chosen randomly from top five previous generations
- 3 epochs per iteration
- 87 iterations
- Best with 0.6076 fitness

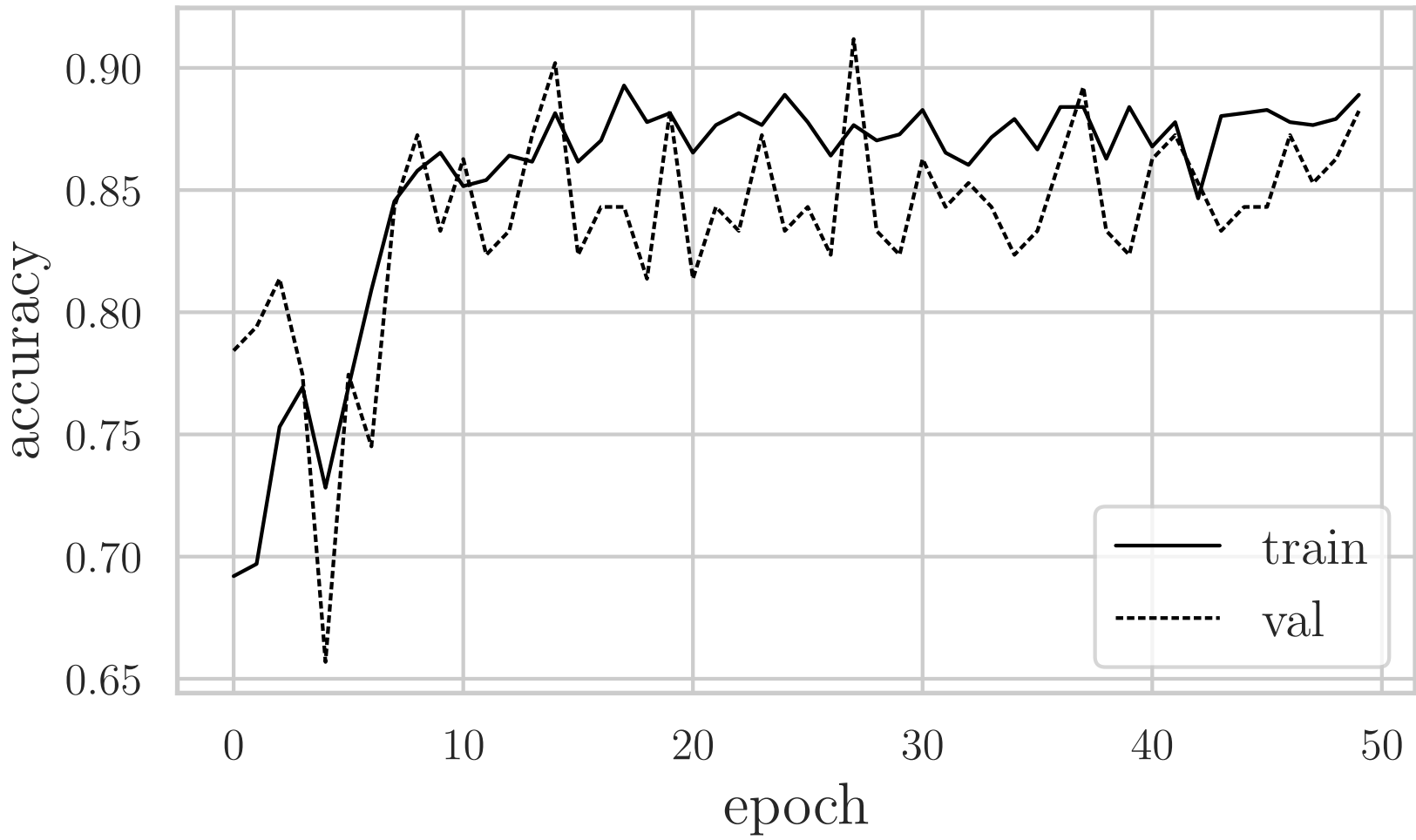
YOLOv7n Hyperparameter Optimization



Prototype Implementation: ResNet-50

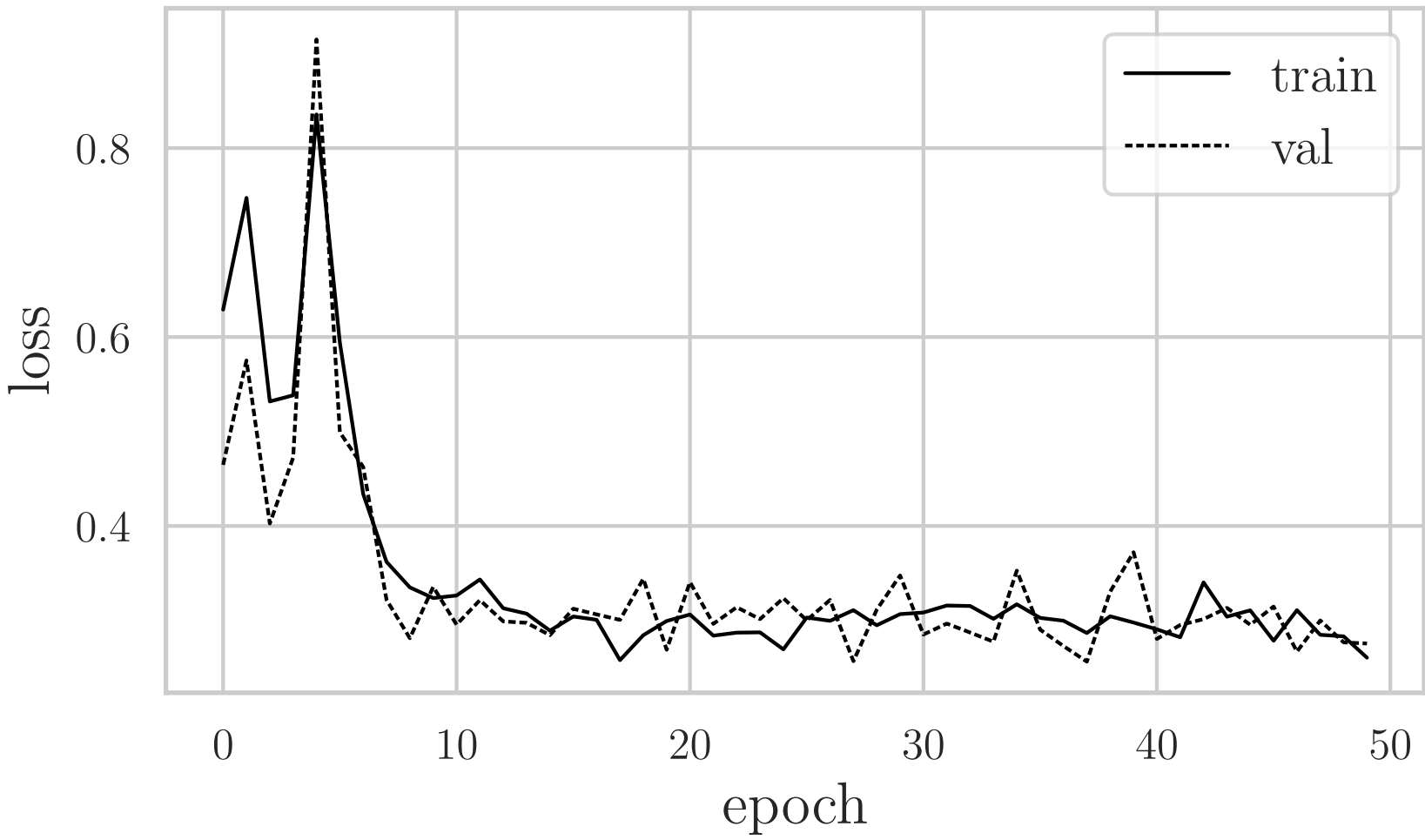
- Pretrained on ImageNet
- Training Set
  - 384 healthy
  - 384 stressed
- Validation Set
  - 68 healthy
  - 68 stressed

Prototype Implementation: ResNet-50 Accuracy



Maximum validation accuracy of 0.9118 at epoch 27

Prototype Implementation: ResNet-50 Loss



ResNet-50 Hyperparameter Optimization

- Random search

ResNet-50 Hyperparameter Optimization

- Random search
- 10 epochs per iteration

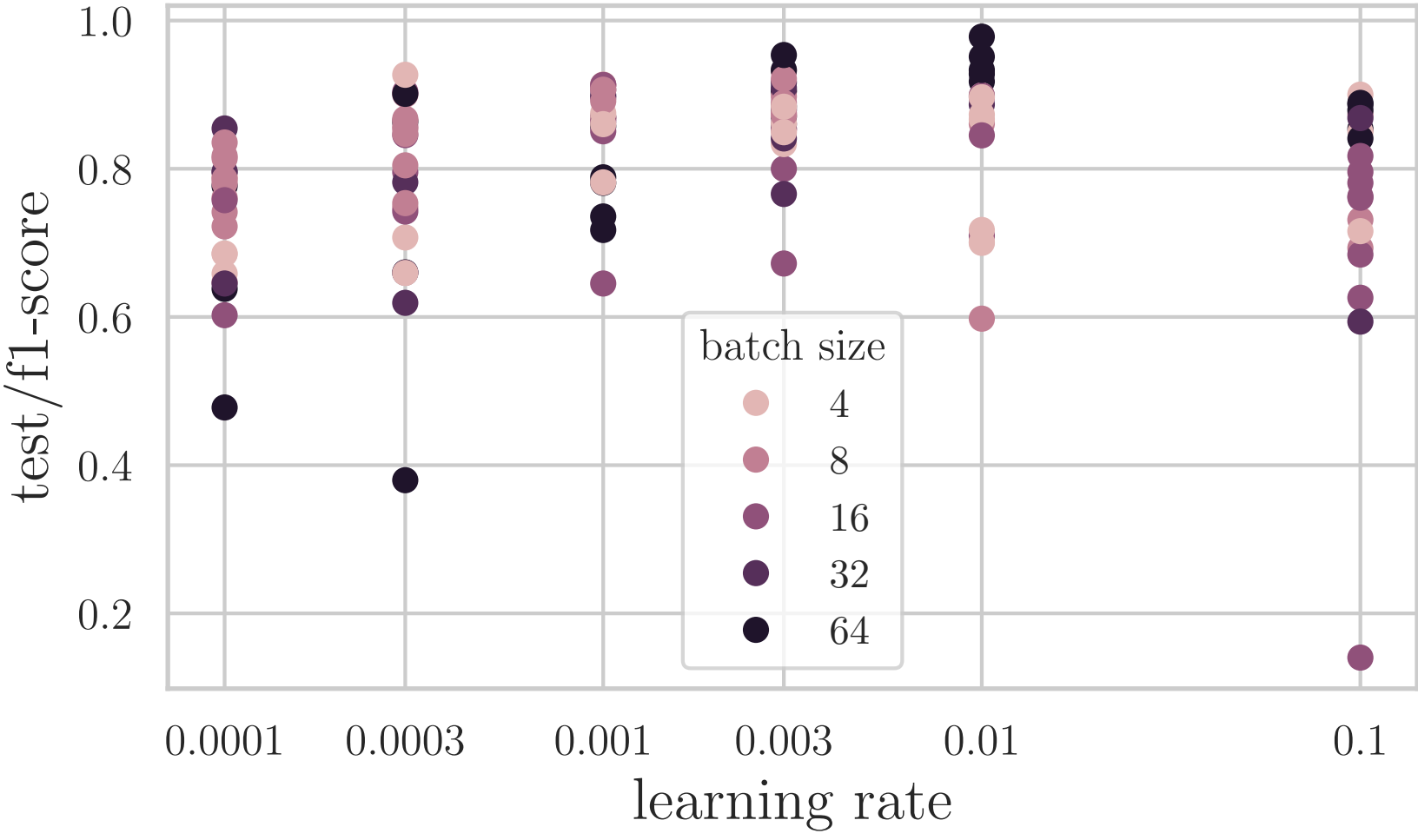
ResNet-50 Hyperparameter Optimization

- Random search
- 10 epochs per iteration
- 138 iterations

ResNet-50 Hyperparameter Optimization

- Random search
- 10 epochs per iteration
- 138 iterations
- Best with 0.9783 F_1-score

ResNet-50 Hyperparameter Optimization



Learning rate and batch size effect on F_1-score

YOLOv7n Evaluation

- Test Set
  - 9000 images
  - 12 238 bounding boxes

YOLOv7n Evaluation

- Test Set
  - 9000 images
  - 12 238 bounding boxes

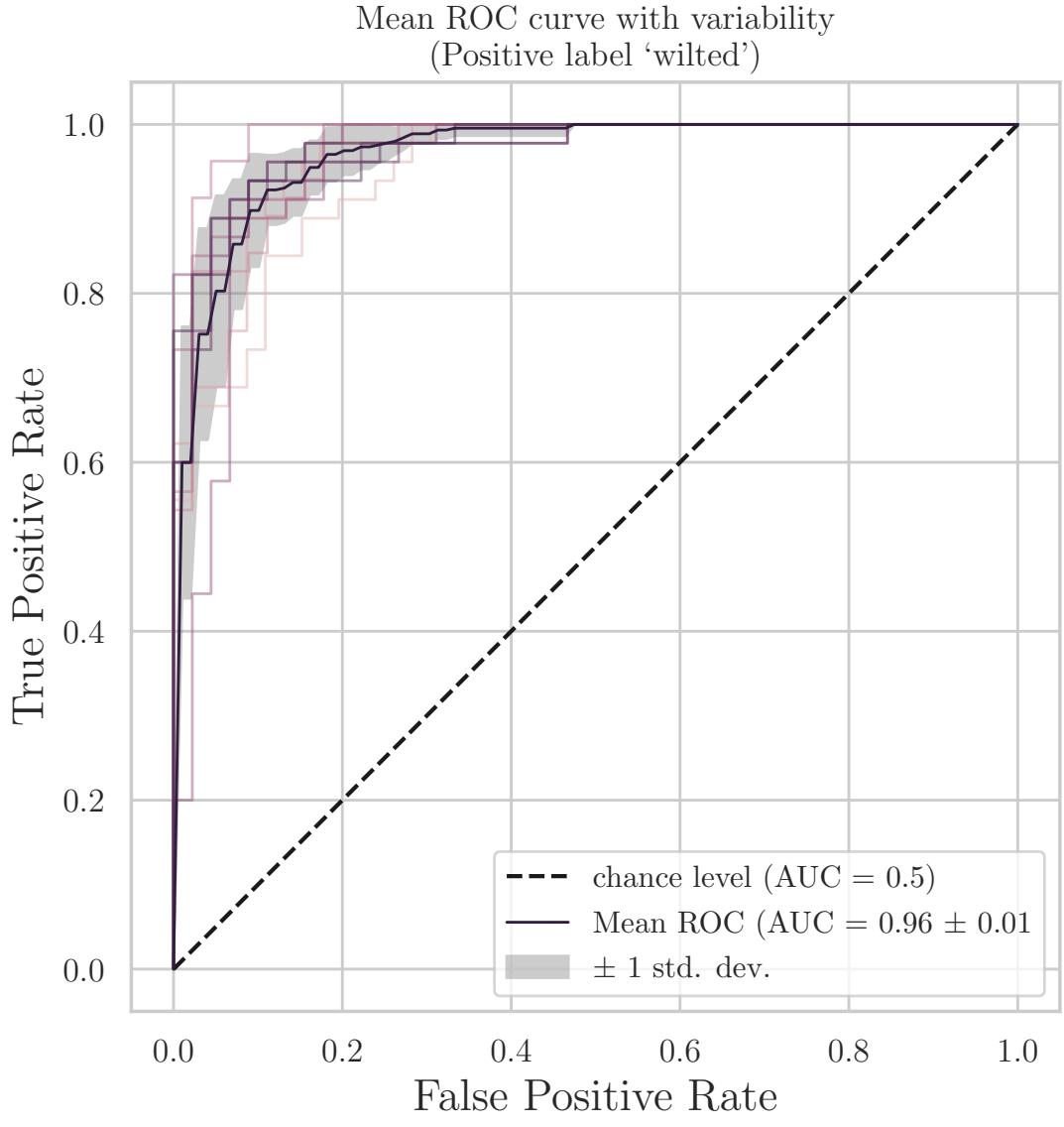
	Precision	Recall	F_1-score	Support
Plant	0.5476	0.7379	0.6286	12 238

Results for the non-optimized object detection model

	Precision	Recall	F_1-score	Support
Plant	0.6334	0.7028	0.6663	12 238

Results for the optimized object detection model

ResNet-50 Evaluation



ROC curves and AUC for classifier 10-fold cross-validation

Aggregate Model Evaluation

- Pre-annotated Test Set
  - 640 images
  - 766 bounding boxes healthy
  - 494 bounding boxes stressed

Aggregate Model Evaluation

- Pre-annotated Test Set
  - 640 images
  - 766 bounding boxes healthy
  - 494 bounding boxes stressed
- Non-optimized model $\mathrm{mAP} = 0.3581$

Aggregate Model Evaluation

- Pre-annotated Test Set
  - 640 images
  - 766 bounding boxes healthy
  - 494 bounding boxes stressed
- Non-optimized model $\mathrm{mAP} = 0.3581$
- Optimized model $\mathrm{mAP} = 0.3838$

Limitations and Conclusions

- I am *not* an expert labeler!

Limitations and Conclusions

- I am *not* an expert labeler!
- Object detection performs well (mAP 0.5727)

Limitations and Conclusions

- I am *not* an expert labeler!
- Object detection performs well (mAP 0.5727)
- Optimized detector worse than non-optimized

Limitations and Conclusions

- I am *not* an expert labeler!
- Object detection performs well (mAP 0.5727)
- Optimized detector worse than non-optimized
- Inconsistent ground truth

Limitations and Conclusions

- I am *not* an expert labeler!
- Object detection performs well (mAP 0.5727)
- Optimized detector worse than non-optimized
- Inconsistent ground truth
- Robust classification

Research Questions Revisited

1. How well does the model work in theory and how well in practice?

Research Questions Revisited

1. How well does the model work in theory and how well in practice?
   - Plant detection comparable to benchmarks

Research Questions Revisited

1. How well does the model work in theory and how well in practice?
   - Plant detection comparable to benchmarks
   - Impressive stress classification

Research Questions Revisited

1. How well does the model work in theory and how well in practice?
   - Plant detection comparable to benchmarks
   - Impressive stress classification

2. What are possible reasons for it to work/not work?

Research Questions Revisited

1. How well does the model work in theory and how well in practice?
   - Plant detection comparable to benchmarks
   - Impressive stress classification

2. What are possible reasons for it to work/not work?
   - Dataset curation

Research Questions Revisited

1. How well does the model work in theory and how well in practice?
   - Plant detection comparable to benchmarks
   - Impressive stress classification
2. What are possible reasons for it to work/not work?
   - Dataset curation
3. What are possible improvements to the system in the future?

Research Questions Revisited

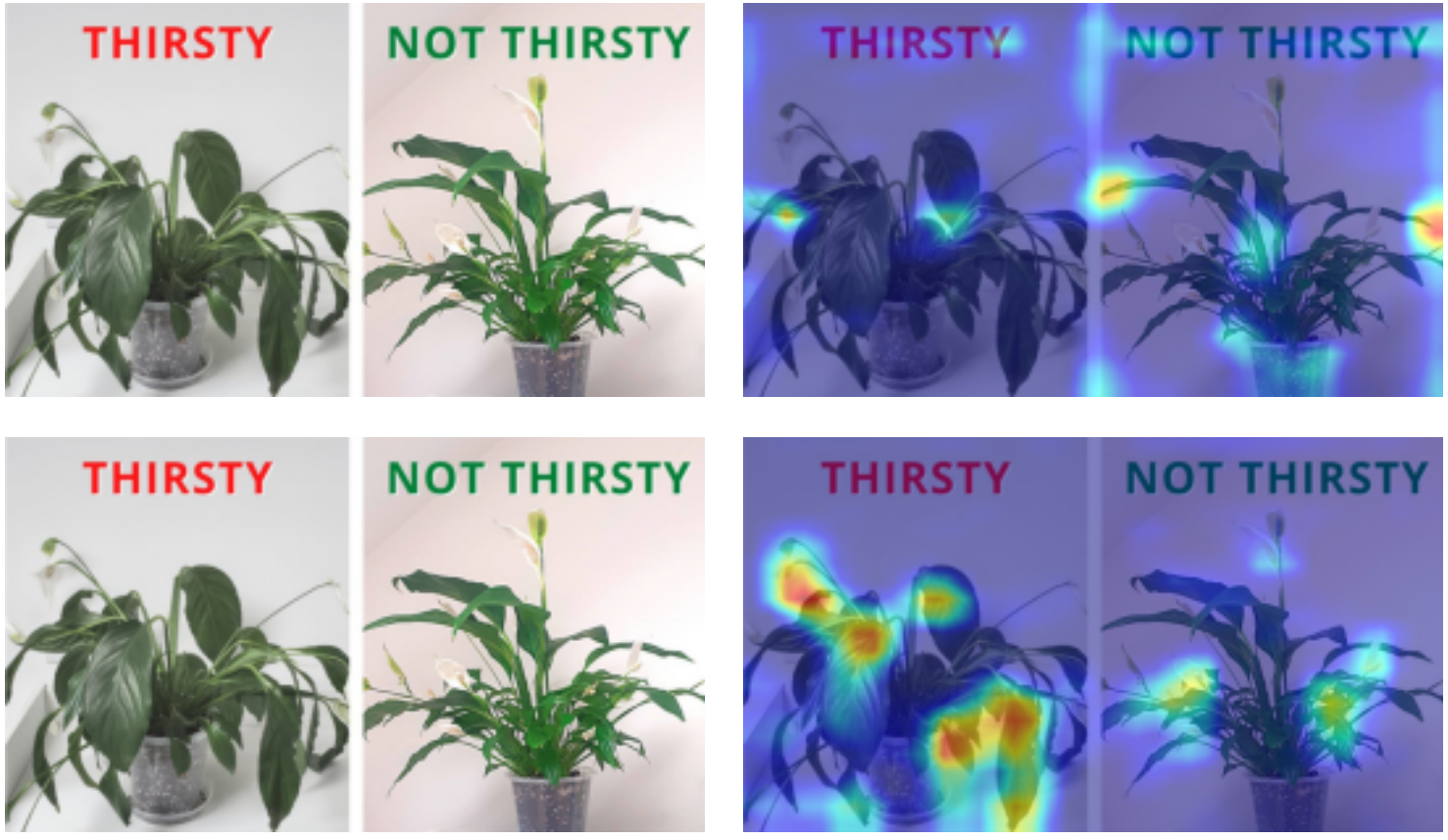
1. How well does the model work in theory and how well in practice?
   - Plant detection comparable to benchmarks
   - Impressive stress classification
2. What are possible reasons for it to work/not work?
   - Dataset curation
3. What are possible improvements to the system in the future?
   - Use more computational resources

Research Questions Revisited

1. How well does the model work in theory and how well in practice?
   - Plant detection comparable to benchmarks
   - Impressive stress classification
2. What are possible reasons for it to work/not work?
   - Dataset curation
3. What are possible improvements to the system in the future?
   - Use more computational resources
   - Expert labeling

Thank you for your attention!

ResNet-50 CAM



Top-right: CAM for *healthy*. Bot-right: CAM for *stressed*

Aggregate Model Evaluation

	Precision	Recall	F_1-score	Support
Healthy	0.665	0.554	0.604	766
Stressed	0.639	0.502	0.562	494
Weighted Avg	0.655	0.533	0.588	1260

Metrics for the non-optimized aggregate model

	Precision	Recall	F_1-score	Support
Healthy	0.711	0.555	0.623	766
Stressed	0.570	0.623	0.596	494
Weighted Avg	0.656	0.582	0.612	1260

Metrics for the optimized aggregate model