

# ML Methods for Localization and Classification of Insects in Images

Philipp Zettl, 841523

BHT Berlin, Berliner Hochschule für Technik



# Content

- ① Intro
- ② Available data sets
- ③ Machine Learning Methods
- ④ Application
- ⑤ Conclusion

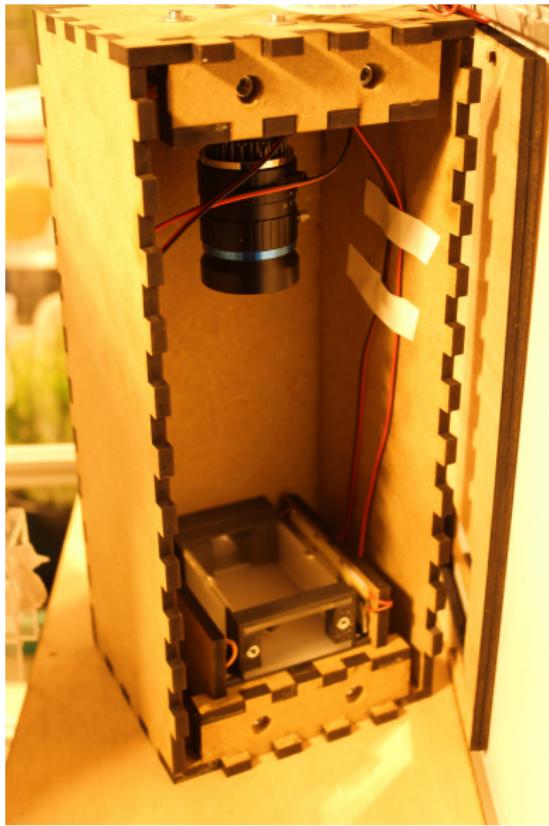
# Part I

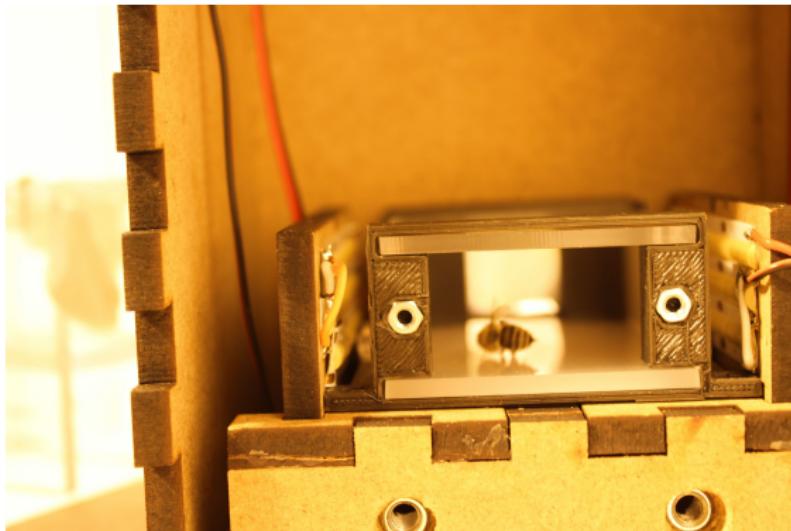
## Intro

# Motivation

- paper[1] from 2017 displays 75-80% decrease in insect biomass over the last 27 years
- KInsecta citizen science project of BHT
- Monitoring of insects
- required to run on Raspberry Pi

# Monitoring Device





# Problem description

Localization and classification of insects in images.



Input

# Problem description

Localization and classification of insects in images.



→ Method

Input

# Problem description

Localization and classification of insects in images.



Input

→ Method →



Output

# Problem description

Localization and classification of insects in images.



Input

Class label → Coleoptera

→ Method →

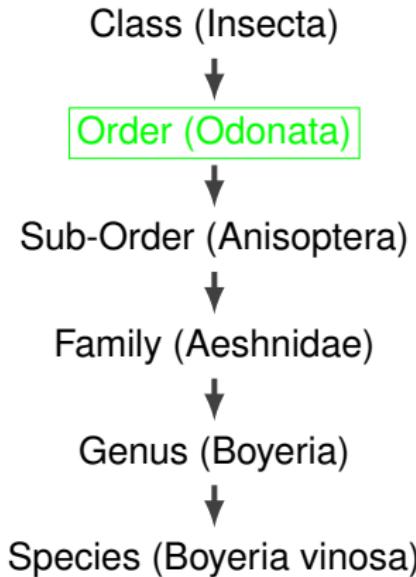


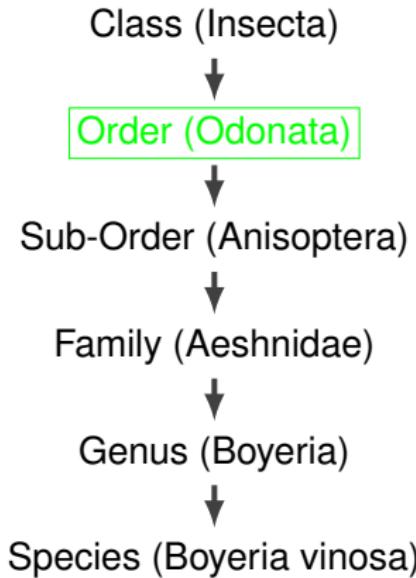
Output

Bounding Box

# Part II

Available data sets

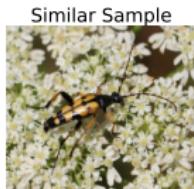




Here used orders:

- Coleoptera (Beetles)
- Lepidoptera (Butterflies)
- Hemiptera (True bugs)
- Hymenoptera
- Odonata (Dragon flies)

## Samples for optical Cross-Order similarity



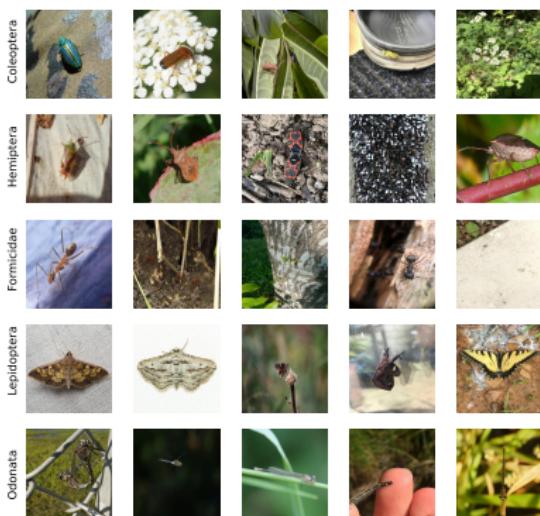
Here used orders:

- Coleoptera (Beetles)
- Lepidoptera (Butterflies)
- Hemiptera (True bugs)
- **Hymenoptera** Formicidae (Ants)
- Odonata (Dragon flies)

# iNaturalist

Representation of reduced *iNat* data set

- original iNaturalist (*iNat*) data set contains all sorts of organisms
- subset *iNat* data set
- JPEG image files
- ≈600.000 insect images



Classification

Regression

## Classification

- Labels given by data set
- No further data generation required

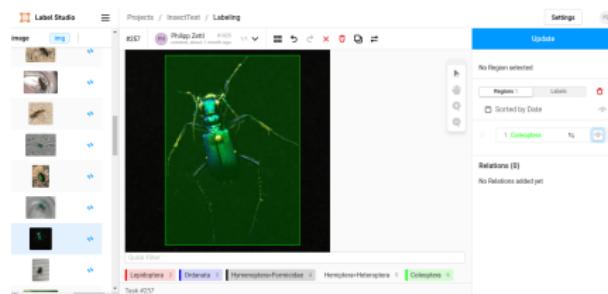
## Regression

## Classification

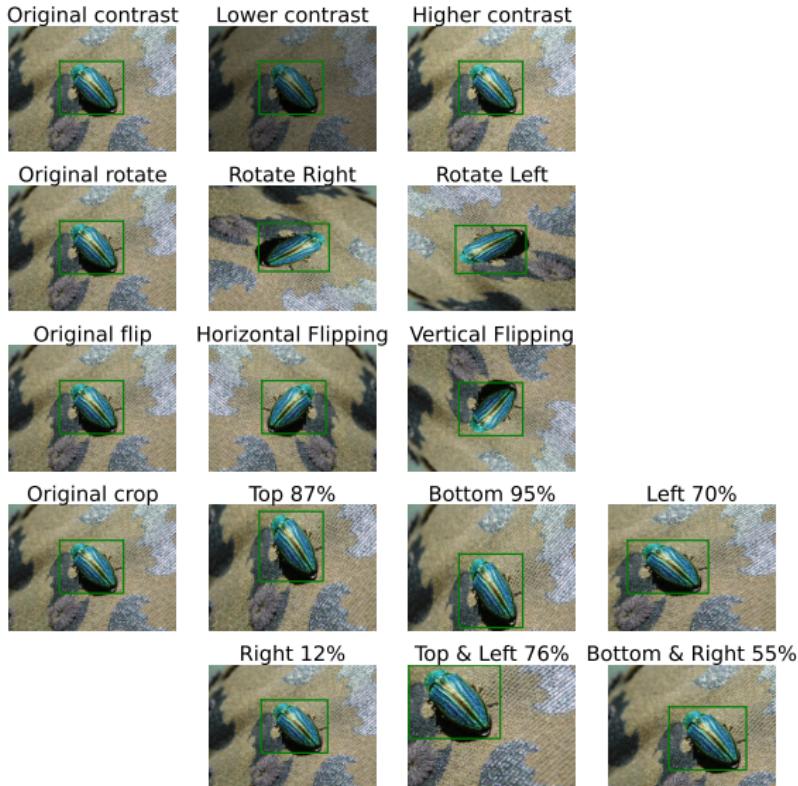
- Labels given by data set
- No further data generation required

## Regression

- No labels available
- Manual generation required



# Augmentation



Applied augmentation techniques visualized on one sample.

# Part III

## Machine Learning Methods

- Conventional Methods

- 
- 

- Artificial Neural Networks

- 
-

- Conventional Methods
  - Linear/Ridge Regression
  - Support Vector Machines (SVMs)
- Artificial Neural Networks
  - 
  -

- Conventional Methods
  - Linear/Ridge Regression
  - SVMs
- Artificial Neural Networks
  - Neural Networks
  - Convolutional Neural Networks

- Conventional Methods
  - Linear/Ridge Regression (**Regression**)
  - SVMs
- Artificial Neural Networks (**Regression**)
  - Neural Networks
  - Convolutional Neural Networks

- Conventional Methods
  - Linear/Ridge Regression (**Regression**)
  - SVMs (**Classification**)
- Artificial Neural Networks (**Regression & Classification**)
  - Neural Networks
  - Convolutional Neural Networks

## Classification

- Cross-Entropy
- Categorical-Cross-Entropy

## Regression

- Vector distances
  - Mean Average Error (MAE)
  - Mean Squared Error (MSE)
- Object oriented
  - Intersection over Union (IoU) loss
  - Generalized IoU loss

## Classification

- F1
- Accuracy

## Regression

- RMSE
- (G)IoU

## Classification

- F1 ( $\uparrow 1.0$ )
- Accuracy ( $\uparrow 1.0$ )

## Regression

- RMSE ( $\downarrow$ )
- (G)IoU ( $\uparrow 1.0$ )

Desired outcome

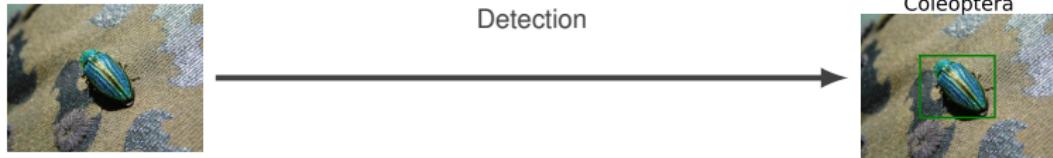
## 2-Stage-Method



## 2-Stage-Method



## Single-Stage-Method



## Part IV

### Application

# Conventional Methods

## SVMs

- Classification
- **One vs. Rest** for 5 classes

## Ridge Regression

- Bounding Box Regression
- OvR for 4 BBox coordinates

Input: Data transformed using sklearn's [2] PCA, to reduce input features.

# Artificial Neural Networks

## Custom architectures

- Split into CNN-backbone and NN-head
- backbones: Custom CNN (INet), MobileNet, VGG-16
- heads: NNs with regularization, target of HPO

## YOLOv5

- implementation of YOLO[3] in PyTorch  
(<https://github.com/ultralytics/yolov5> [4])
- predicts (multiple) BBs and class labels at once  
(single-stage-method)

### Conventional methods:

- Ind.: Independently trained SMVs and RidgeReg models
- Seq.: Sequentially trained, first RidgeReg models then SVMs based on predictions from RidgeReg models

## Conventional methods:

<b>Method</b>	<b>GloU(↑)</b>	<b>RMSE(↓)</b>	<b>Accuracy(↑)</b>	<b>F1(↑)</b>
<i>PCA<sub>100</sub></i>				
Ind.	0.3481	21.2985	0.4806	0.4725
Seq.	0.3481	21.2985	0.2840	0.2685
<i>PCA<sub>400</sub></i>				
Ind.	0.3525	20.9326	0.5339	0.5200
Seq.	0.3525	20.9326	0.3040	0.2704

## Conventional methods:

<b>Method</b>	<b>GloU(↑)</b>	<b>RMSE(↓)</b>	<b>Accuracy(↑)</b>	<b>F1(↑)</b>
<i>PCA<sub>100</sub></i>				
Ind.	0.3481	21.2985	0.4806	0.4725
Seq.	0.3481	21.2985	0.2840	0.2685
<i>PCA<sub>400</sub></i>				
Ind.	0.3525	20.9326	0.5339	0.5200
Seq.	0.3525	20.9326	0.3040	0.2704

# Conventional methods

Ind. using input transformed by PCA<sub>400</sub>



Normalized confusion matrix

	Coleoptera	Hymenoptera>Formicidae	Hemiptera	Odonata	Lepidoptera
Coleoptera	0.50	0.19	0.19	0.02	0.11
Hymenoptera>Formicidae	0.20	0.34	0.15	0.20	0.10
Hemiptera	0.24	0.05	0.29	0.33	0.10
Odonata	0.16	0.12	0.12	0.44	0.16
Lepidoptera	0.10	0.14	0.19	0.12	0.46

Coleoptera  
Hymenoptera>Formicidae  
Hemiptera  
Odonata  
Lepidoptera

Predicted label

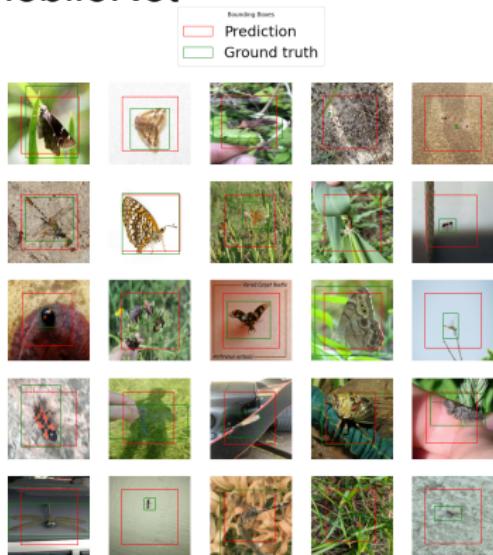
# Regression

<b>Backbone</b>	<b>Data set</b>	<b>GloU(↑)</b>	<b>RMSE(↓)</b>
INet	Raw	0.3838	25.9176
MobileNet	Raw	0.3877	25.3727
VGG-16	Raw	0.0525	24.2184
INet	Augmented	0.3670	26.3442
<b>MobileNet</b>	<b>Augmented</b>	0.5602	17.0061
VGG-16	Augmented	-0.4000	40.7438

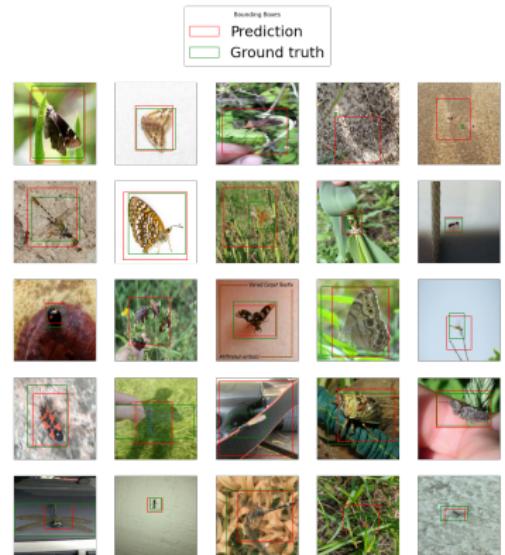
# Regression

<b>Backbone</b>	<b>Data set</b>	<b>GloU(↑)</b>	<b>RMSE(↓)</b>
INet	Raw	0.3838	25.9176
MobileNet	Raw	0.3877	25.3727
VGG-16	Raw	0.0525	24.2184
INet	Augmented	0.3670	26.3442
<b>MobileNet</b>	<b>Augmented</b>	0.5602	17.0061
VGG-16	Augmented	-0.4000	40.7438

## Regression MobileNet

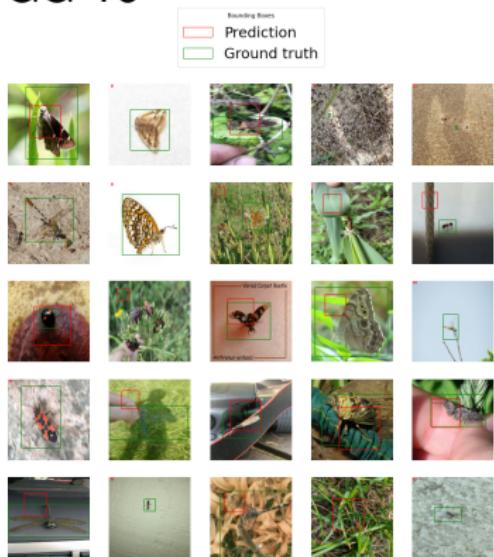


Trained using unaugmented data

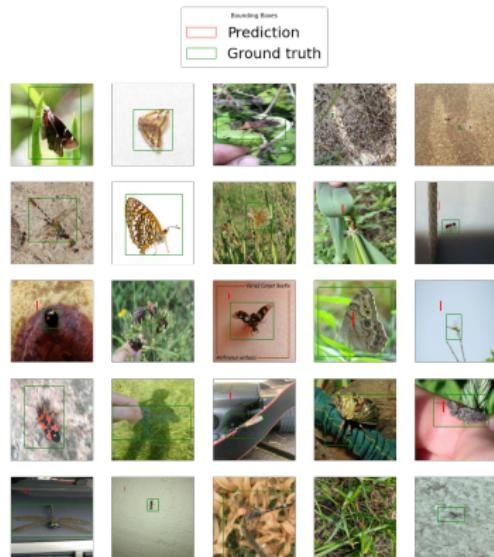


Trained using augmented data

## Regression VGG-16



Trained using unaugmented data



Trained using augmented data

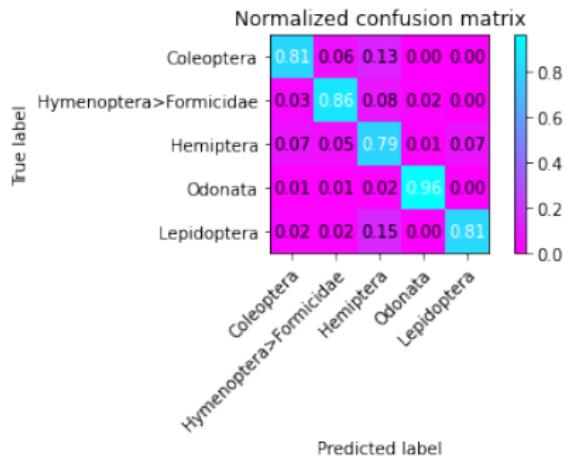
## Classification

Backbone	Data set	Accuracy(↑)	F1(↑)
INet	Raw	0.6422	0.7793
<b>MobileNet</b>	<b>Raw</b>	0.8511	0.8462
VGG-16	Raw	0.8467	0.8439
INet	Augmented	0.6422	0.7793
MobileNet	Augmented	0.8333	0.826
VGG-16	Augmented	0.8467	0.8439
MobileNet	Uncropped, Raw	0.7844	0.7793
MobileNet	Predicted, Raw	0.5289	0.5097

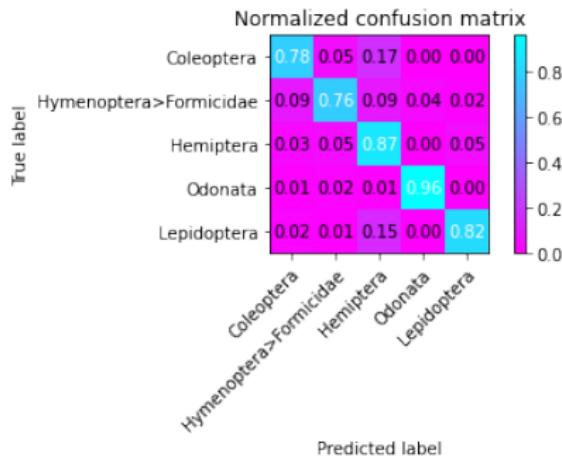
## Classification

Backbone	Data set	Accuracy(↑)	F1(↑)
INet	Raw	0.6422	0.7793
<b>MobileNet</b>	<b>Raw</b>	<b>0.8511</b>	<b>0.8462</b>
VGG-16	Raw	0.8467	0.8439
INet	Augmented	0.6422	0.7793
MobileNet	Augmented	0.8333	0.826
VGG-16	Augmented	0.8467	0.8439
MobileNet	Uncropped, Raw	0.7844	0.7793
MobileNet	Predicted, Raw	0.5289	0.5097

# Classification MobileNet



Trained using unaugmented data



Trained using augmented data

# Two-Stage-Methods: Training results

- Sequential & Independent assembled of:
  - BB Regression: MobileNet trained on augmented data
  - Classification: MobileNet trained on unaugmented data
- Results of BB Regression are equal due to equal models

## Two-Stage-Methods: Training results

Method	GIoU( $\uparrow$ )	RMSE( $\downarrow$ )	Accuracy( $\uparrow$ )	F1( $\uparrow$ )
<b>Independent</b>	0.5602	17.0061	0.8511	0.8462
Sequential	0.5602	17.0061	0.5666	0.5661

## Two-Stage-Methods: Training results

Method	GloU(↑)	RMSE(↓)	Accuracy(↑)	F1(↑)
Independent	0.5602	17.0061	0.8511	0.8462
Sequential	0.5602	17.0061	0.5666	0.5661

# Single-Stage-Methods: Training results

<b>Backbone</b>	<b>Data set</b>	<b>GloU(↑)</b>	<b>RMSE(↓)</b>	<b>Accuracy(↑)</b>	<b>F1(↑)</b>
INet	Raw	0.3566	24.2111	0.4844	0.4625
MobileNet	Raw	-0.4751	39.1935	0.7311	0.7793
VGG-16	Raw	-0.5731	39.1935	0.7311	0.7793
INet	Aug.	0.3042	24.2111	0.4844	0.4625
<b>MobileNet</b>	<b>Aug.</b>	0.3604	20.5186	0.7555	0.7496
VGG-16	Aug.	-0.4624	38.7296	0.7800	0.7779

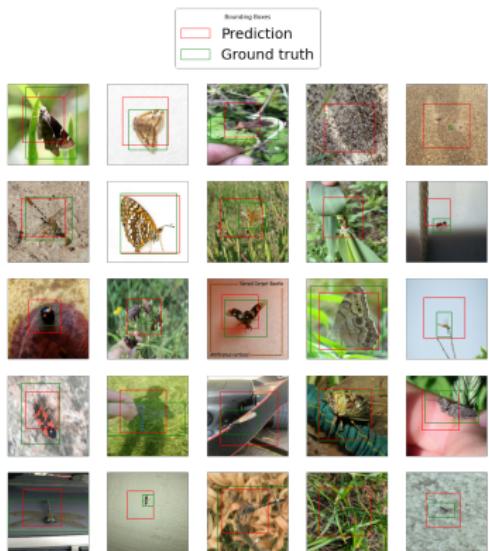
# Single-Stage-Methods: Training results

<b>Backbone</b>	<b>Data set</b>	<b>GloU(↑)</b>	<b>RMSE(↓)</b>	<b>Accuracy(↑)</b>	<b>F1(↑)</b>
INet	Raw	0.3566	24.2111	0.4844	0.4625
MobileNet	Raw	-0.4751	39.1935	0.7311	0.7793
VGG-16	Raw	-0.5731	39.1935	0.7311	0.7793
INet	Aug.	0.3042	24.2111	0.4844	0.4625
<b>MobileNet</b>	<b>Aug.</b>	0.3604	20.5186	0.7555	0.7496
VGG-16	Aug.	-0.4624	38.7296	0.7800	0.7779

Unfortunately, it was not possible to generate metrics during training for YOLOv5

# MobileNet

Trained using augmented data



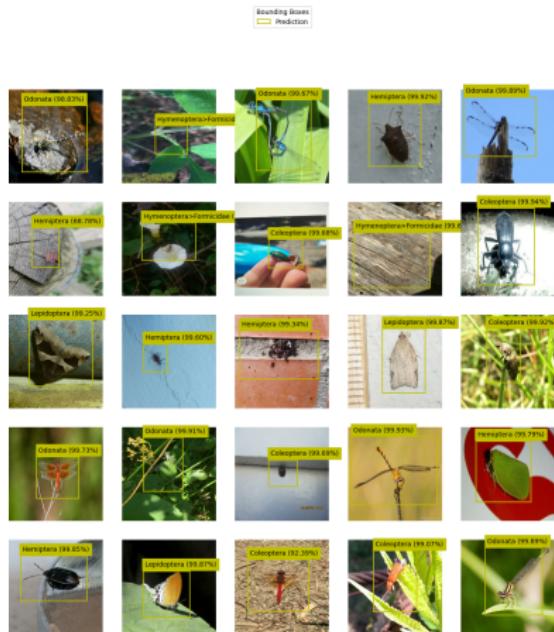
## Validation of trained models on test set

Method	GloU(↑)	RMSE(↓)	Accuracy(↑)	F1(↑)	Inf. time [s]
Independent	0.56	17.35	0.92	0.92	0.60
Sequential	0.55	18.03	0.52	0.53	0.79
Single-Stage	0.37	19.67	0.92	0.92	0.22
<b>YOLOv5</b>	<b>0.69</b>	<b>18.78</b>	<b>0.85</b>	<b>0.84</b>	<b>0.20</b>

## Validation of trained models on test set

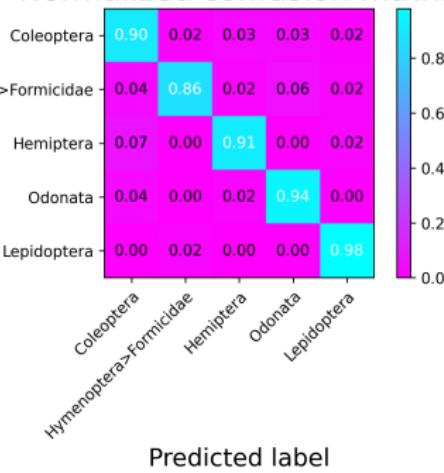
<b>Method</b>	<b>GloU(↑)</b>	<b>RMSE(↓)</b>	<b>Accuracy(↑)</b>	<b>F1(↑)</b>	<b>Inf. time [s]</b>
Independent	0.56	17.35	0.92	0.92	0.60
Sequential	0.55	18.03	0.52	0.53	0.79
Single-Stage	0.37	19.67	0.92	0.92	0.22
<b>YOLOv5</b>	<b>0.69</b>	<b>18.78</b>	<b>0.85</b>	<b>0.84</b>	<b>0.20</b>

# Independent

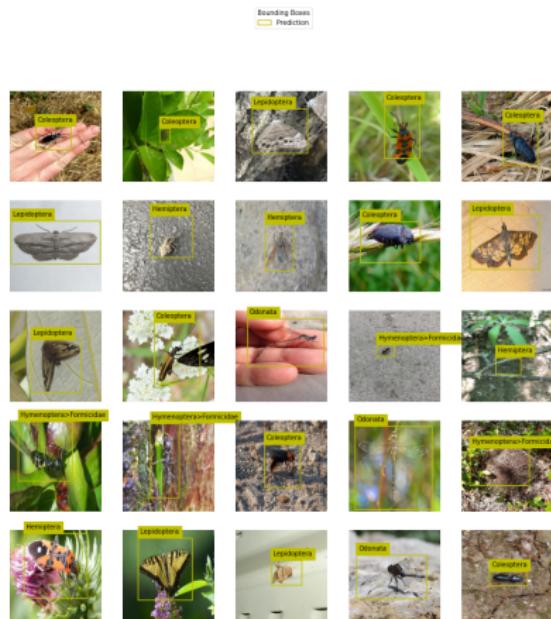


True label

Normalized confusion matrix

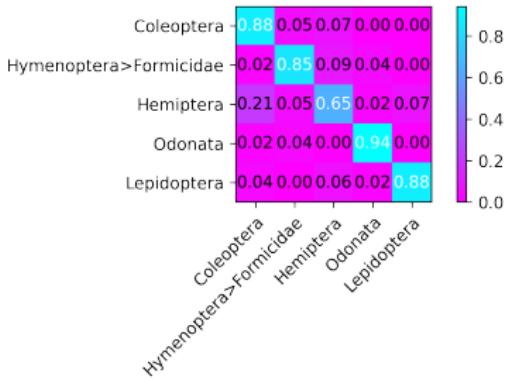


## YOLOv5



True label

Normalized confusion matrix



Predicted label

## Model reduction

- MobileNet for classification: 14.2 MB → 13.7 MB using post training quantization
- MobileNet for regression: 40.6 MB → 13.3 MB using weight clustering
- Single-Stage model with MobileNet backbone: 42.2 MB → 13.9 MB using weight clustering

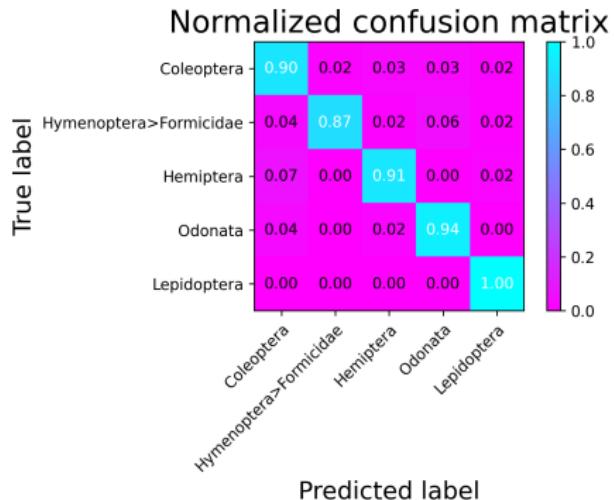
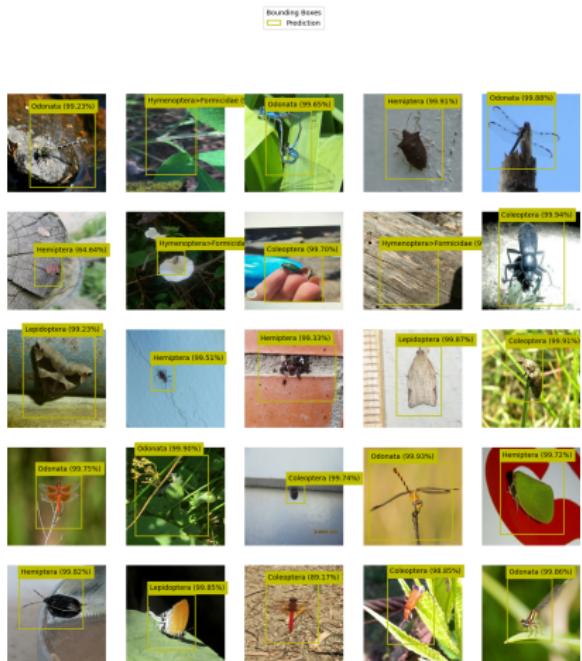
## Final inference tests

Method	GloU(↑)	RMSE(↓)	Accuracy(↑)	F1(↑)	Inf. time [s]
Independent	0.5540	18.0264	0.9200	0.9210	1.3140
Sequential	0.5540	18.0262	0.5200	0.5285	2.2799
Single-Stage	0.1833	21.7866	0.6080	0.5804	0.7272
<b>YOLOv5</b>	0.6847	22.7424	0.8200	0.8091	2.0840

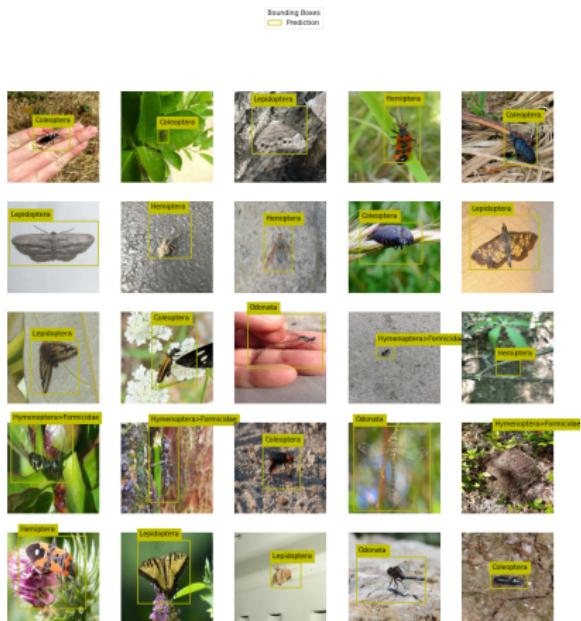
## Final inference tests

Method	GloU(↑)	RMSE(↓)	Accuracy(↑)	F1(↑)	Inf. time [s]
Independent	0.5540	18.0264	0.9200	0.9210	1.3140
Sequential	0.5540	18.0262	0.5200	0.5285	2.2799
Single-Stage	0.1833	21.7866	0.6080	0.5804	0.7272
<b>YOLOv5</b>	0.6847	22.7424	0.8200	0.8091	2.0840

# Independent

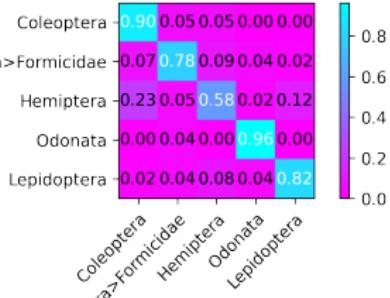


## YOLOv5



True label

Normalized confusion matrix



Coleoptera  
Hymenoptera>Formicidae  
Hemiptera  
Odonata  
Lepidoptera

Predicted label

# Part V

## Conclusion

# Conclusion

- INet and VGG-16 perform mediocre on all tasks
- using MobileNet generates good starting results
- YOLOv5 works out of the box better than regular CNN architectures

A short presentation of manual labelling of 2.500 image files.

<https://youtu.be/twjyfQ7sXk4?t=43>

## Tasks

- General supervised learning task

$$T : \mathbb{R}^N \mapsto \mathbb{R}^K, \mathbf{x}^{(m)} \mapsto \hat{\mathbf{y}}^{(m)}$$

## Tasks

- General supervised learning task

$$T : \mathbb{R}^N \mapsto \mathbb{R}^K, \mathbf{x}^{(m)} \mapsto \hat{\mathbf{y}}^{(m)}$$

- Bounding Box Regression:

$$T_r : \mathbb{R}^N \mapsto \mathbb{R}^4, \mathbf{x}^{(m)} \mapsto \hat{\mathbf{c}}^{(m)}$$

## Tasks

- General supervised learning task

$$T : \mathbb{R}^N \mapsto \mathbb{R}^K, \mathbf{x}^{(m)} \mapsto \hat{\mathbf{y}}^{(m)}$$

- Bounding Box Regression:

$$T_r : \mathbb{R}^N \mapsto \mathbb{R}^4, \mathbf{x}^{(m)} \mapsto \hat{\mathbf{c}}^{(m)}$$

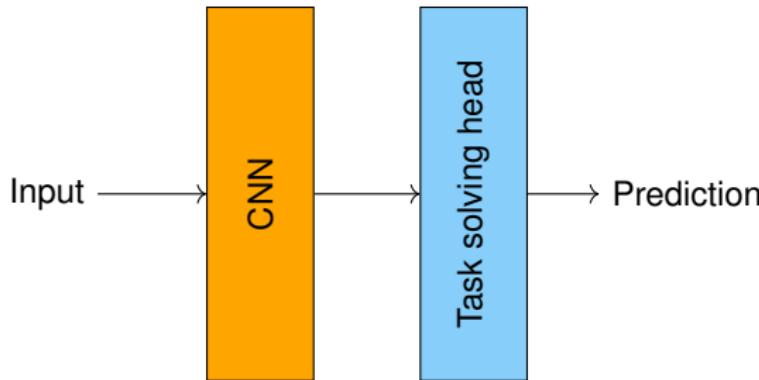
- Classification:

$$T_c : \mathbb{R}^N \mapsto [0, 1]^5, \mathbf{x}^{(m)} \mapsto \hat{\mathbf{y}}^{(m)}$$

# Artificial Neural Networks

## CNN-Backbones

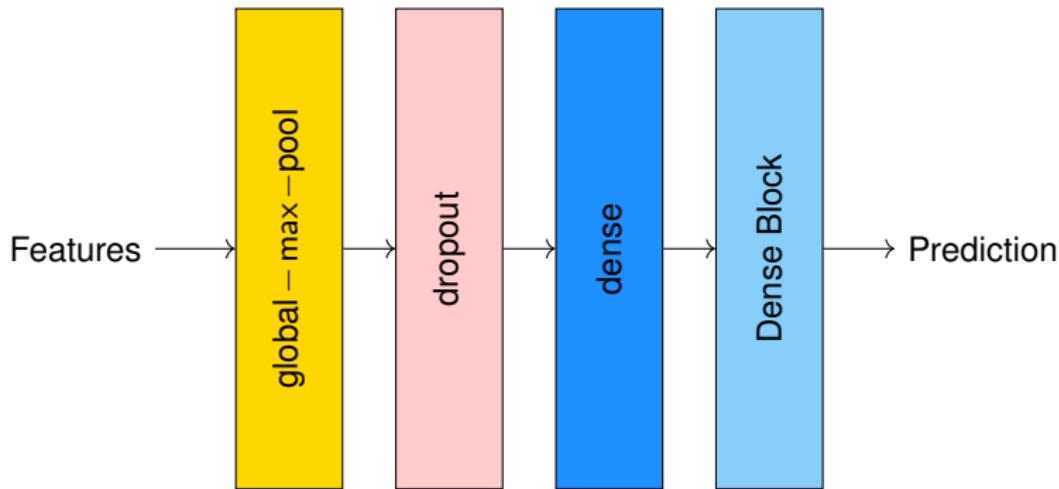
- Extract features from input
- Require additional "Task solving head" to solve one or more tasks
- INet, MobileNet, VGG-16
- Input: normalized rescaled image data



# Artificial Neural Networks

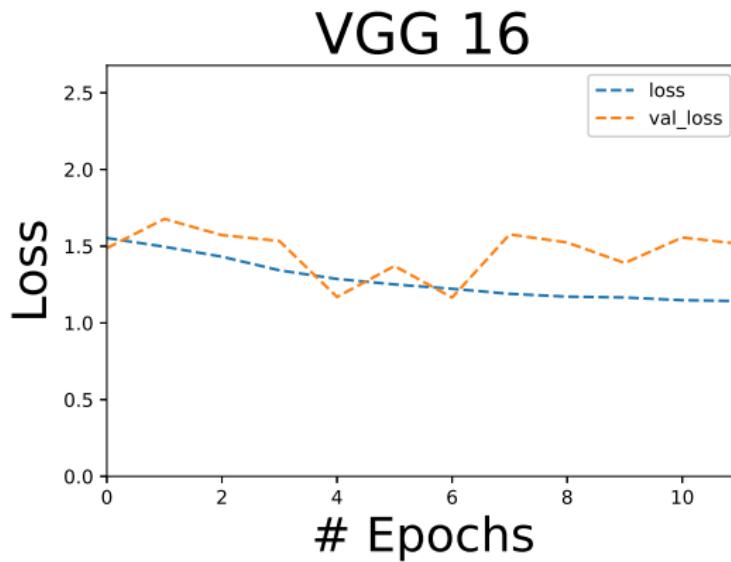
## NN-Task solving heads

- can perform predictions for single task
- 4 elements: GlobalMaxPooling, Dropout, Dense, Dense-Block



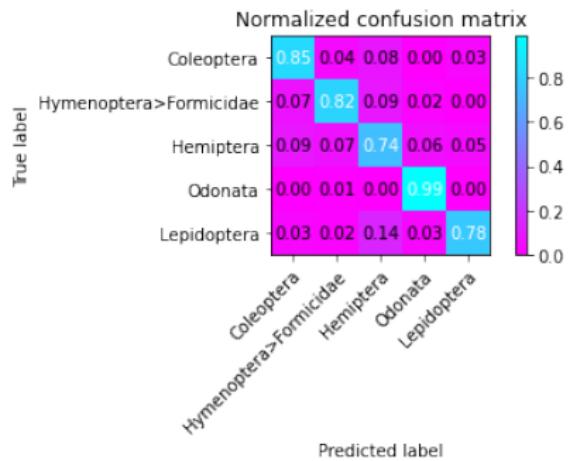
# Regression

VGG-16 trained with augmented training data.

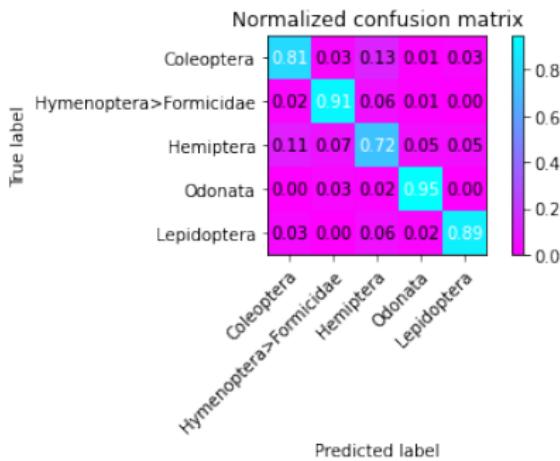


# Classification

## VGG-16



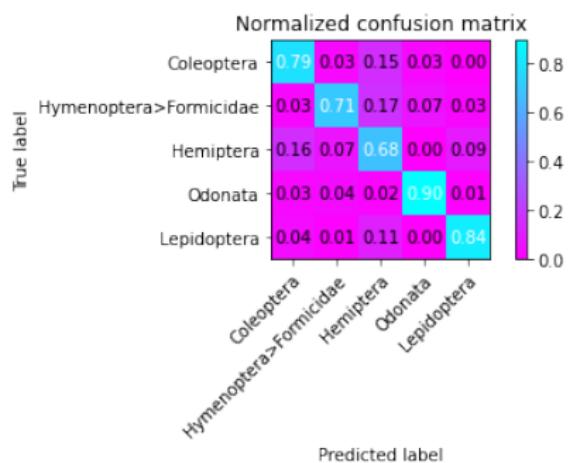
Trained using unaugmented data



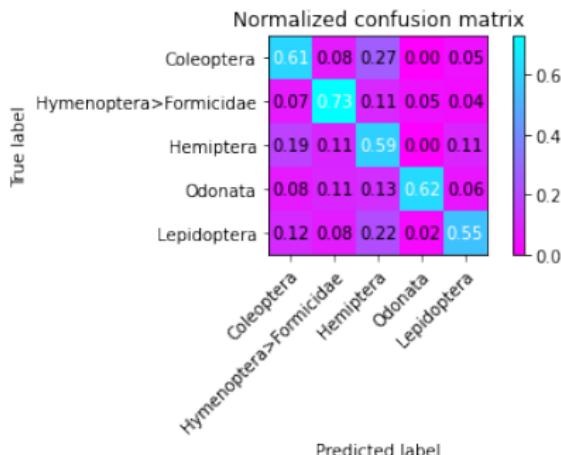
Trained using augmented data

# Classification

## MobileNet

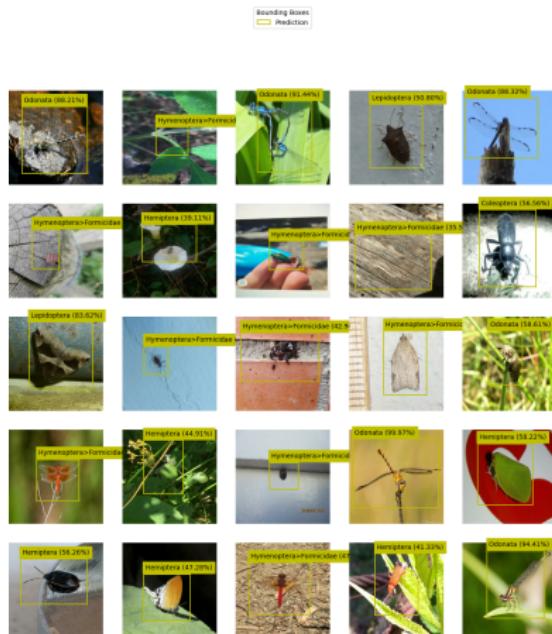


Trained using uncropped images



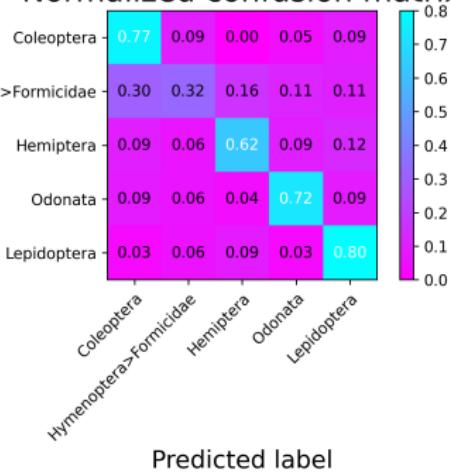
Trained using cropped images, based on BB predictions

# Sequential

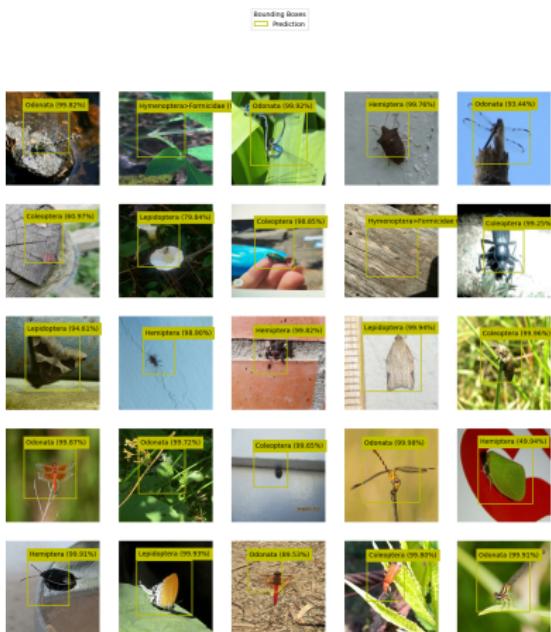


True label

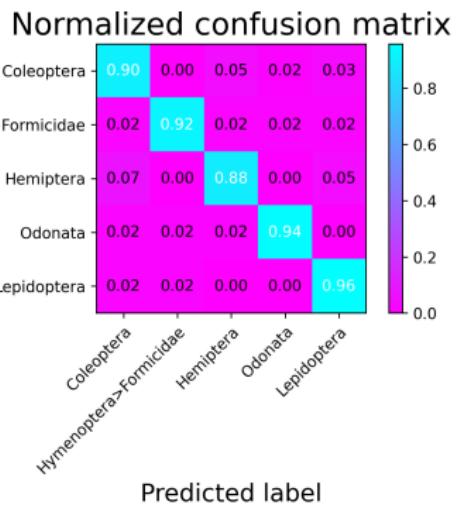
Normalized confusion matrix



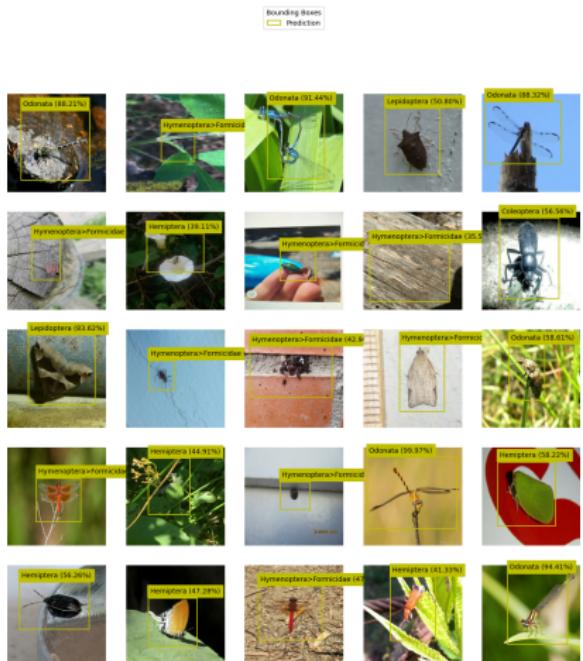
## Single-Stage



True label

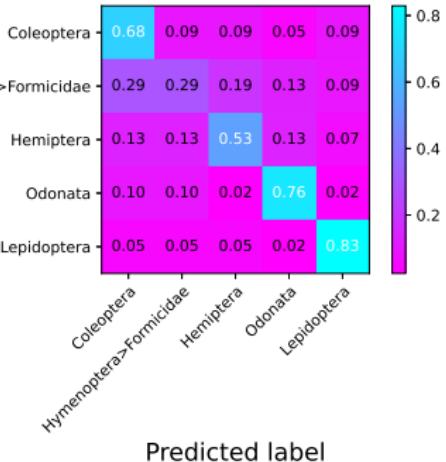


# Sequential

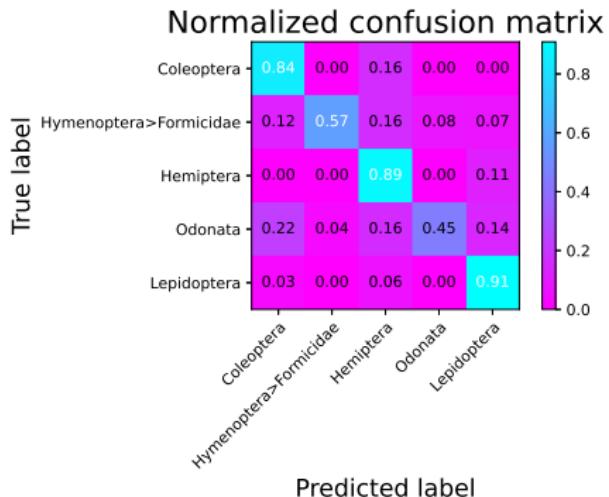
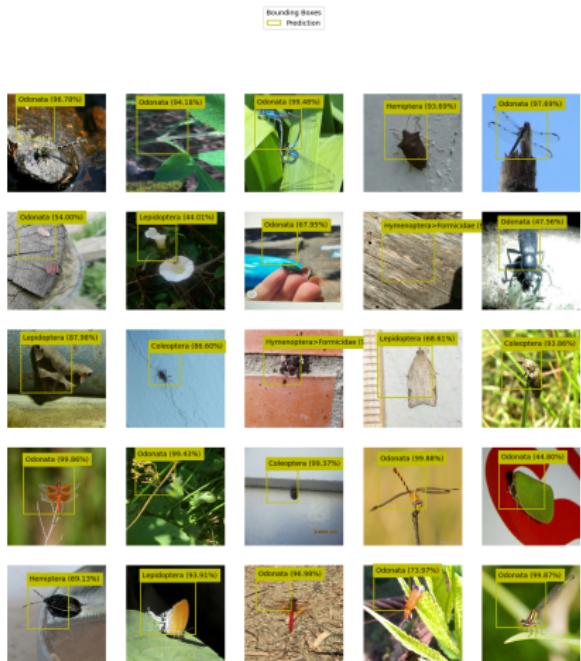


True label

Normalized confusion matrix



# Single-Stage



-  C. A. Hallmann, M. Sorg, E. Jongejans, H. Siepel, N. Hofland, H. Schwan, W. Stenmans, A. Müller, H. Sumser, T. Hörren, D. Goulson, and H. de Kroon, "More than 75 percent decline over 27 years in total flying insect biomass in protected areas," *PLOS ONE*, vol. 12, pp. 1–21, 10 2017.
-  F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
-  J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016.
-  A. Kathuria, "How to Train YOLO v5 on a Custom Dataset," 2021.