This CVPR Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

End-to-end Solution for Tenebrio Molitor Rearing Monitoring with Uncertainty Estimation and Domain Shift Detection

Paweł Majewski*, Piotr Lampa, Robert Burduk, Jacek Reiner

Wrocław University of Science and Technology, Poland

*corresponding author

pawel.majewski@pwr.edu.pl

Abstract

The large-scale rearing of edible insects, of which Tenebrio Molitor is a representative, requires monitoring using vision systems to control the process and to detect anomalies. Previously proposed solutions by researchers relied on multiple modules related to specific tasks (calculated coefficients) and specific types of models (instance segmentation, semantic segmentation). Long processing times and difficulties in maintaining and updating modules encourage the search for a more condensed solution as an end-to-end model. This paper proposed a modified YOLOv8 architecture extended with additional heads related to specific tasks. Heads were trained on problem-oriented small datasets, which significantly reduced the time spent on sample annotation. The proposed solution also included estimation of prediction uncertainty based on variation among predictions in model ensemble and detection of domain shift phenomenon. Quantitative results from the conducted experiments confirmed the potential of the developed solution.

1. Introduction

Increasing demands on the quantity and quality of food produced worldwide are necessitating the search for new food sources and alternative approaches to food production [9]. Insect rearing (including edible insects) for feed and food purposes is becoming an increasingly important part of the agri-food industry. Among the most popular insect species reared for feed purposes are Hermetia Ilucens (HI) [3] and Tenebrio Molitor (TM) [12]. The distinguishing factor of farming the mentioned insects is the possibility of obtaining a product rich in protein, fat and minerals at much lower environmental costs (greenhouse gas emissions, water consumption) as in the case of traditional farming (pigs, cattle) [21, 24]. The profitability of HI and TM insect farming is closely related to its large-scale nature, which necessitates the automation of basic farming operations (e.g. feeding, harvesting) [22] on the one hand and the need for its monitoring on the other. Information obtained from data analysis is also needed to control rearing and make critical decisions, e.g. to end rearing and to change the feeding strategy.

Researchers have already addressed the problem of monitoring insect rearing on the example of the TM using a vision system and computer vision methods [16, 20]. The proposed solutions allowed (1) detection and counting of the TM growth stages (larva, pupa, beetle, (2) detection and counting of anomalies (dead larva), (3) estimation of the amount of chitinous moults and feed, and (4) estimation of size indicators of larvae (referred to as phenotyping). The developed methods were based on the following models: Mask R-CNN [13] for instance segmentation, U-Net [23] for semantic segmentation, and YOLOv5 [14] for object detection and classical image processing methods.

An undeniable disadvantage of existing solutions for TM rearing monitoring is their multi-module nature, i.e. many separate models related to a specific task. With such an approach, the problems of long image processing time (difficulty of achieving real-time inference), maintenance and updating of specific modules are of great importance. Researchers have also addressed the problem of simplifying some parts of processing, e.g., the phenotyping module, by proposing custom regression deep convolutional neural network instead of multistage image processing. However, the problem of developing a comprehensive solution should be considered still open [18].

With the above in mind, we propose an end-to-end solution for monitoring the rearing of the TM based on the YOLOv8 [15] object detection model extended with additional heads associated with a specific task (calculated indicators). To reduce labelling efforts, an approach of training individual heads on problem-oriented small datasets was proposed. Given the importance of some of the calculated

indicators in terms of rearing control and in order to increase the reliability of the solution, a method for estimating prediction uncertainties using an ensemble of models was also proposed. The calculated prediction uncertainties were also used to detect the domain shift phenomenon, which, considering the changeable conditions on the farm, is a significant problem.

2. Related Work

The problem of reducing multiple tasks to a single model architecture (with a shared backbone) and developing end-to-end solutions is eagerly addressed in many application areas of computer vision [2, 32], including agriculture and phenotyping of biosystems [6, 30]. Many approaches to developing condensed model architectures can be distinguished. In this section, three selected ones will be discussed, namely (1) multioutput regression models, (2) extending basic models with new heads (branches), and (3) multi-task learning.

Multioutput regression models. With this approach, all defined tasks are implementable by calculating a certain number of numerical values representing specific indicators. In [31], an architecture based on a backbone pretrained on ImageNet [8] was proposed for the simultaneous calculation of six physical indicators that characterize cattle, namely the length and width of specific body parts (shoulder, hip, body) along with the estimated weight. The input to the model was recorded depth images. A combined loss based on MSE (mean squared error) was used for training, consisting of parts corresponding to prediction errors for a specific indicator. In [19], different fruit traits, i.e. moisture content (MC) and soluble solids content (SSC), were predicted simultaneously based on spectral signals from NIR spectroscopy. The proposed custom architectures consisted of a certain number of convolution layers and fully connected layers. The combined loss MSE for different coefficients was used for training as in [31].

Extending basic models with new heads (branches). A common approach to extend the functionality of the solution with new tasks is to extend the basic architecture with additional heads. Applying this approach, the Faster R-CNN [11] architecture was extended in [4] to include an additional branch for weight estimation. In [29], an additional block for direct counting of soybean pods was proposed as a modification to the YOLOv5 [14] model.

Multi-task learning. For some types of tasks, there is a need for output in the form of predictions of different types, for example, returning simultaneously bounding boxes for an object detection problem along with a predicted map for a semantic segmentation problem. For these types of issues, multi-task learning methods are helpful. The challenge in multi-task learning is to propose a suitable loss function that takes into account predictions in different formats, often with fine-tuning the weights of specific parts in the loss function. In [5], the problem of detection and determination of cherry tomato maturity was extended to the task of detection and determination of maturity of the whole bunch. For this purpose, additional improved heads to the YOLOv7 [26] model and a combined loss function for the tasks posed were proposed. In [27], inspired by the YOLOP [28] model, a solution was proposed for the simultaneous detection of peppers, pepper segmentation and stem segmentation. The minimized loss during training consisted of three parts related to the defined tasks.

3. Problem Definition

The problem addressed in this paper is the calculation of multiple indicators that characterize the current status of TM rearing based on RGB images of TM rearing boxes (shown in Fig. 1).

Figure 1. Example image of a rearing box with Tenebrio Molitor.

The tasks undertaken include: (1) counting TM states (beetles, dead larvae and pupae), (2) estimating indicators of box coverage with chitinous moults and feed, and (3) calculating size indicators (width, length) of larvae.

Compared to the nomenclature in [16], the presented article combines object classes from the 'growth stages' and 'anomalies' groups into a single group called 'states' due to the possibility of counting objects from all classes related to TM using a single object detection model.

The counting of live larvae was abandoned from the tasks undertaken since the number of live larvae in the rearing box should be constant under normal conditions. The estimated number of live larvae will also strongly depend on the growth stage of the larvae, which is related to the influence of occlusion on the results and the tendency of larvae to hide in the substrate. With these problems present, interpreting the change in the number of live larvae over time can be problematic for the farmer.

4. Dataset

Multiple datasets were developed for the experiments, and each dataset was associated with a specific task. The defined datasets contained tiles of a certain size extracted from the whole image (with the size of 4096x3000 pixels) of a rearing box with Tenebrio Molitor as in Fig. 1.

The base dataset contained 640x640 images for training the basic YOLOv8 model to detect objects from three classes: beetles (B), dead larvae (DL) and pupae (P). Sample images with objects from the classes under consideration are shown in Fig. 2. The base dataset contained 373 images with a total number of annotations of 3442 (367 for beetles, 1781 for dead larvae and 1294 for pupae). For the base dataset, the annotations were bounding boxes.

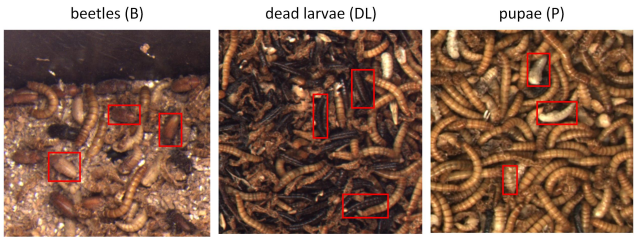

Figure 2. Classes of detected and counted objects with example bounding boxes.

For the task of estimating the chitin coverage index (CCI) and feed coverage index (FCI), labelling was performed for 150 images with 640x640 size. Labelling consisted of marking all areas in the image representing chitin or feed. Based on the annotated images, CCI and FCI coefficients were calculated as target values. Selected samples with assigned values of CCI and FCI coefficients are shown in Fig. 3.

For the larvae phenotyping task related to calculating the three quartiles of larvae width (lower, median, upper), a dataset described in [18] consisting of 739 images of 1024x1024 size was used. Sample images from this dataset are also shown in Fig. 3.

To conduct experiments for the detection of the domain shift effect, a separate dataset was developed, consisting of images from three domains related to image registration by different vision systems (different cameras, lighting). The developed dataset contained 640x640 images, respectively 87 from the base domain, 29 from domain A and 15 from domain B. Sample images from the three considered domains are presented in Fig. 4. Details on the defined domains can also be found in [17] (data source 'JA' is the base domain, 'LU' is domain A, 'CA' is domain B).

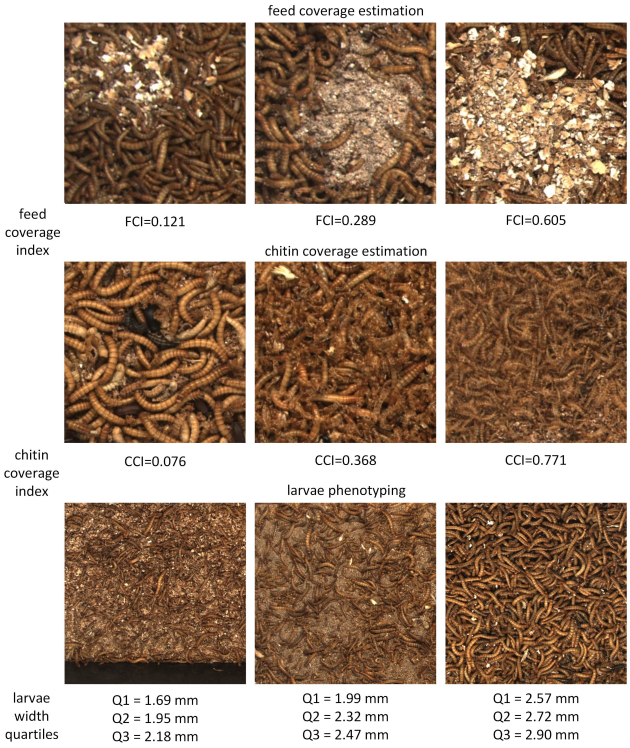

Figure 3. Examples of samples from problem-oriented datasets for training machine learning models to proposed additional heads in YOLO architecture.

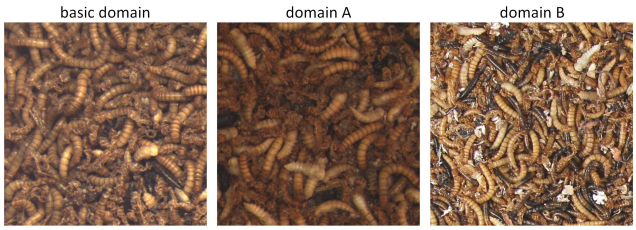

Figure 4. Examples of images from defined domains.

5. Proposed Approach

The proposed approach to calculating informative indicators for monitoring the rearing of TM is a modified architecture of the YOLOv8 model for object detection, which has been extended with additional problem-oriented heads, namely (1) feed coverage estimation head, (2) chitin coverage estimation head and (3) larvae phenotyping head. At the training stage, each head was separately fine-tuned using a different dataset prepared for a specific problem, saving considerable annotation time. The YOLOv8 base model allowed the detection of objects from the beetle, dead larvae and pupae classes and their counting. Feed and chitin coverage estimation heads calculated image coverage indices for feed or chitin, respectively. Coverage indices should

be understood as the number of pixels associated with the considered classes (feed or chitin) divided by the total number of pixels. In the case of the larvae phenotyping head, the output was three quartiles (lower, median, upper) of the width of the larvae as proposed in [18].

5.1. Model Architecture

The developed model architecture is shown in Fig. 5. The proposed three heads used features extracted from specific backbone layers of the YOLOv8 model (these are the layers with indexes from 0 to 9 shown in Fig. 5). In the case of the YOLOv8 model under consideration, the backbone is the CSPDarknet53 [25] feature extractor. The indexes of the layers used for feature extraction for the problems posed were determined experimentally, and the procedure is described in the following sections of the paper. Based on the extracted features, the selected classical machine learning models calculated the specified indices. Fig. 5 places the heads in specific locations associated with the results obtained in the conducted experiments.

5.2. Model Ensemble and Uncertainty Estimation

To increase the estimation accuracy of the proposed indices and enable the estimation of prediction uncertainty, an ensemble of models was considered for prediction. A bootstrap method was used to train successive models, which involved training successive YOLOv8n models on different subsets of samples determined from the basic object detection dataset. The prediction of an ensemble of models was the unweighted average of single model predictions. Uncertainty was calculated as the standard deviation among single-model predictions.

5.3. Domain Shift Detection

The possibility of a domain shift effect is associated with a change in the nature of the registered images. The first source of changes can be different acquisition conditions, for example, due to significant dust or contamination of the elements of the vision system. Changes can also be associated with a variation in the type of feed used or the type of rearing box used. The domain shift effect can also occur when implementing a monitoring system for a new large-scale farm.

The method for detecting the domain shift effect was based on calculated prediction uncertainties. A logistic regression model was used for the binary classification task. In addition to the standard approach of detecting domain shift for single samples, the detection of this phenomenon was also considered when averaging the uncertainty values from a subset of samples of a specific size. It was justified from the point of view of the problem addressed (registration of multiple images under large-scale rearing conditions).

6. Experiments

6.1. Selection of YOLO Core Architecture

The first stage of the conducted experiments was training YOLOv8 models using architectures with different complexity and number of parameters (n, s, m, l and x versions). The training was repeated in 5 iterations of cross-validation, where the whole dataset was divided into train/val and test parts. Model training was performed on the training set. Based on the validation set, the best training epoch was selected. On the test set, an evaluation was carried out. The best architecture was selected for further experiments based on the averaged results (metrics) obtained on the test set.

6.2. Selection of Best Settings for Proposed Heads

The next experiment aimed to determine the optimal settings (layer ID for feature extraction and the type of classical machine learning model for the regression task) for the proposed heads. Using the GridSearch approach, further combinations of settings were examined, whereby layers for feature extraction with indexes from 0 to 9 and the following machine learning models for regression were considered: linear regression (LR), k-nearest neighbours regression (KNN), support vector regression (SVR) [7] and gradient boosting regression (GBR) [10]. As in the first experiment, training was repeated for different iterations of cross-validation, and the results were averaged. The search for the best settings was conducted for each defined head separately. The selected best settings of each head were used for further experiments.

6.3. Model Ensemble and Uncertainty Estimation

The next experiment involved developing an ensemble of YOLOv8 models. For this task, the train/val and test splits from the cross-validation from the first experiment were used. Training of subsequent models was carried out on sets determined using bootstrapping. Each determined training set was extracted from the train/val part, with about 70% of the unique samples from the train/val part in the training set. To check the effect of the number of single models in the ensemble on the results, the prediction was performed in ensemble mode, averaging the single model predictions using an unweighted average. Prediction uncertainty was also determined based on the standard deviation among single model predictions in the ensemble.

6.4. Domain Shift Detection

The last experiment was developing a model for detecting the domain shift effect based on estimated prediction uncertainties. The Logistic Regression model was used for this task. The cases of two domains (A and B) that differed from the basic domain were considered. The obtained values of the metrics in the stratified cross-validation were referred

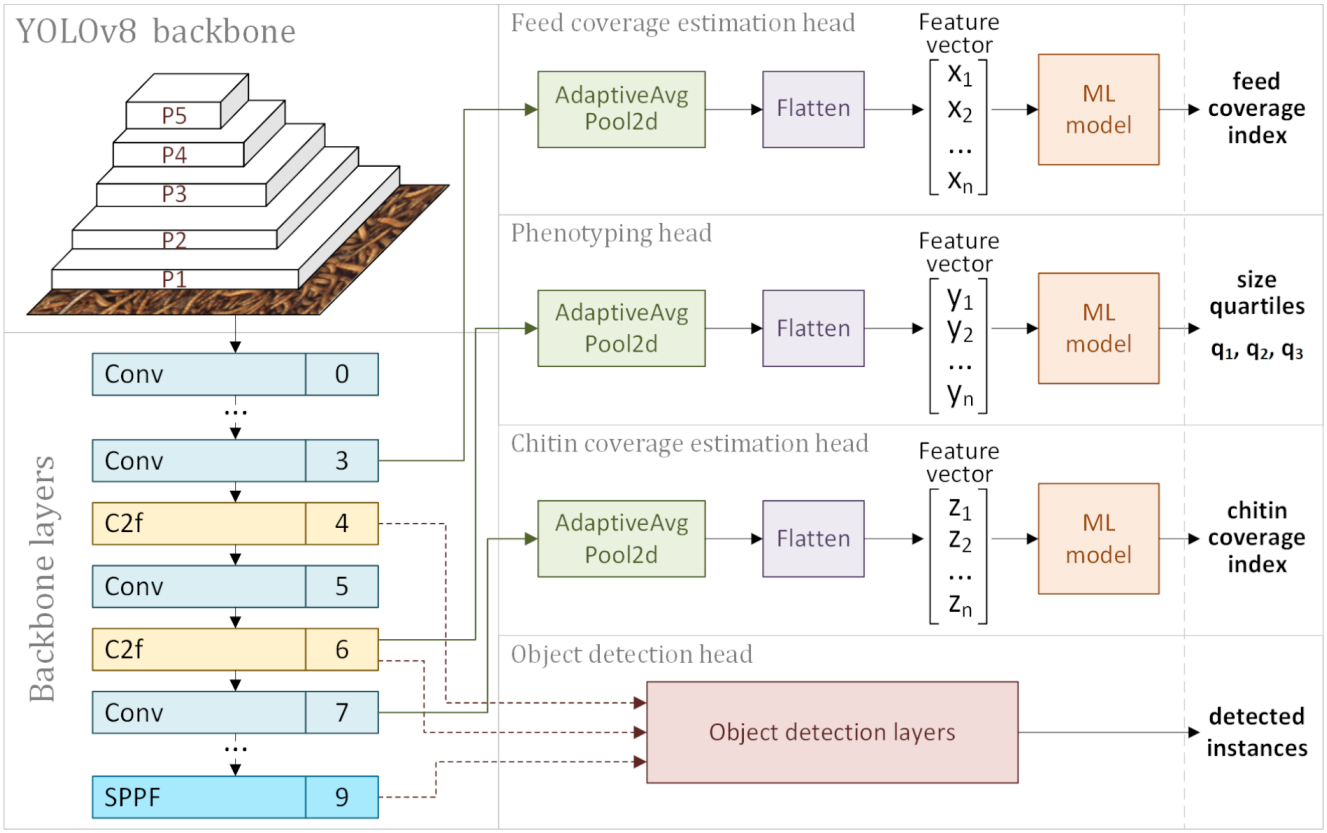

Figure 5. Modified YOLOv8 architecture with proposed additional heads: feed coverage estimation head, chitin coverage estimation head and phenotyping head.

to as the results of this experiment. The study also tested the hypothesis of the possibility of increasing the detection accuracy of the domain shift effect by averaging the prediction uncertainty over several samples. Experiments were conducted with different numbers of samples considered for averaging.

7. Evaluation

The proposed methods were evaluated using standard metrics for a specific problem. The referred averaged values of the specified metrics with standard deviation were based on the results obtained in successive cross-validation iterations. Consistently, the number of splits in cross-validation was set at five for all problems posed.

7.1. Metrics for Regression Problems

For the evaluation of regression tasks (TM states counting, estimation of chitin and feed coverage indexes), three metrics were used: mean absolute error (MAE), coefficient of determination (R^2) and Pearson correlation coefficient (r), which can be calculated using formulas Eq. (1), Eq. (2), and Eq. (3).

$$MAE = \frac{1}{n_{sample}} \sum_{i=1}^{n_{sample}} |g_i - p_i| \tag{1}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n_{sample}} (g_i - p_i)^2}{\sum_{i=1}^{n_{sample}} (g_i - \overline{g})^2} \tag{2}$$

$$r = \frac{\sum_{i=1}^{n_{sample}} (g_i - \overline{g})(p_i - \overline{p})}{\sqrt{\sum_{i=1}^{n_{sample}} (g_i - \overline{g})^2} \sqrt{\sum_{i=1}^{n_{sample}} (p_i - \overline{p})^2}} \tag{3}$$

Where n_{sample} is the number of samples, p_i - prediction for the i-th sample, g_i - target value (true) for the i-th sample, $\overline{p}$ - averaged prediction values, $\overline{g}$ - averaged target values.

7.2. Metrics for Uncertainty Estimation

Evaluation of the prediction uncertainty estimation method was carried out as follows. Using a specified number of predictions in an ensemble of models, the 95 percent prediction uncertainties interval (95 PPU) was determined, calculating the lower (X_i^L) and upper (X_i^U) bounds of the interval being

the 2.5th and 97.5th percentiles, as in the article [1]. Having the limits of the interval, it was checked what part of the predictions fell within the determined uncertainty interval, which was referenced under the metric called *pred. in 95 PPU*. Based on the X_i^L and X_i^U values, the degree of uncertainty $\overline{d_x}$ was also determined from the formula Eq. (4) and then the d-factor metric from the formula Eq. (5).

$$\overline{d_x} = \frac{1}{n_{sample}} \sum_{i=1}^{n_{sample}} (X_i^U - X_i^L) \tag{4}$$

$$d - factor = \frac{\overline{d_x}}{\sigma_x} \tag{5}$$

Where σ_x is the standard deviation among the target values for the selected problem

7.3. Metrics for Domain Shift Detection

To evaluate domain shift detection models, precision, recall and F1-score metrics were used, whose formulas can be found in Eq. (6), Eq. (7) and Eq. (8).

$$precision = \frac{TP}{TP + FP} \tag{6}$$

$$recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{8}$$

Where TP, TN, FP and FN represent the number of true positive, true negative, false positive and false negative predictions, respectively.

7.4. Inference Time

The referenced inference time values were based on predictions made using hardware with the following specifications: GeForce RTX 2060 SUPER 8 GB (GPU) and AMD Ryzen 7 1700 3 GHz (CPU). When referencing inference times for the entire rearing box (size of 4096x3000), the inference was assumed for 54 individual tiles (dividing the entire image into 640x640 tiles with 25% overlap).

8. Results and Discussion

8.1. Selection of YOLO Core Architecture

In the first step of developing the proposed solution, the appropriate architecture of the YOLOv8 model was selected, the results of which are presented in Tab. 1.

Based on the results obtained in Tab. 1, it was decided to select the YOLOv8n architecture for further experiments. The YOLOv8n architecture had the highest metrics for the counting task and the highest throughput, which is particularly important for inference in model ensemble mode.

model	class	MAE	R^2	r
YOLOv8n	B	**0.20**$_{\pm 0.04}$	**0.959**$_{\pm 0.011}$	**0.985**$_{\pm 0.007}$
YOLOv8n	DL	**1.07**$_{\pm 0.21}$	**0.908**$_{\pm 0.020}$	**0.962**$_{\pm 0.005}$
YOLOv8n	P	**0.82**$_{\pm 0.05}$	0.969$_{\pm 0.012}$	0.988$_{\pm 0.006}$
YOLOv8s	B	0.21$_{\pm 0.06}$	0.955$_{\pm 0.009}$	0.982$_{\pm 0.004}$
YOLOv8s	DL	1.19$_{\pm 0.09}$	0.893$_{\pm 0.019}$	0.955$_{\pm 0.002}$
YOLOv8s	P	0.88$_{\pm 0.10}$	0.966$_{\pm 0.011}$	0.988$_{\pm 0.006}$
YOLOv8m	B	0.21$_{\pm 0.08}$	0.949$_{\pm 0.020}$	0.977$_{\pm 0.012}$
YOLOv8m	DL	1.14$_{\pm 0.12}$	0.897$_{\pm 0.022}$	0.955$_{\pm 0.009}$
YOLOv8m	P	0.85$_{\pm 0.14}$	0.966$_{\pm 0.017}$	**0.989**$_{\pm 0.005}$
YOLOv8l	B	0.21$_{\pm 0.07}$	0.956$_{\pm 0.032}$	0.979$_{\pm 0.017}$
YOLOv8l	DL	1.26$_{\pm 0.15}$	0.888$_{\pm 0.022}$	0.951$_{\pm 0.008}$
YOLOv8l	P	0.83$_{\pm 0.09}$	0.969$_{\pm 0.015}$	0.987$_{\pm 0.008}$
YOLOv8x	B	0.23$_{\pm 0.08}$	0.954$_{\pm 0.012}$	0.982$_{\pm 0.007}$
YOLOv8x	DL	1.22$_{\pm 0.22}$	0.889$_{\pm 0.012}$	0.953$_{\pm 0.010}$
YOLOv8x	P	0.85$_{\pm 0.05}$	**0.973**$_{\pm 0.010}$	0.988$_{\pm 0.004}$

Table 1. Results for Tenebrio Molitor states (beetle/B, dead larva/DL, pupa/P) counting for different types of YOLO models.

It is noteworthy that already at the level of the object detection model, it was possible to achieve a significant reduction in computation time compared to [16], where the YOLOv5x model characterized by an inference time of 40 ms/tile was used. In the case of the YOLOv8n model, the inference time was 7.9 ms/tile. The reduction in computation time would be even greater assuming batch inference (the throughput for YOLOv8n was 395 tiles/s). This results in a computation time of about 0.14s for the entire rearing box (composed of 54 tiles). With such values of processing times, even ensemble mode inference, with a reasonable number of single models, is reasonable.

8.2. Selection of Best Settings for Proposed Heads

In the next step, the best settings (machine learning model for regression and layer ID for feature extraction) were searched for the proposed additional heads. The results from this step are presented in Tab. 2.

Based on the results in Tab. 2, it can be concluded that different models and features extracted from different layers were the best choice for different tasks. Finally, for the chitin coverage estimation head, the GBR model based on features extracted from the 7th layer was chosen; for the feed coverage estimation head - the LR model and features from the 3rd layer; and for the phenotyping head - the GBR model along with features from the 6th layer. The relatively high results ($R^2 > 0.78$) confirmed the validity of the proposed solution based on attaching additional heads to the base YOLOv8n model. The lowest results were achieved for the estimation of the chitin coverage index. This may be due to the high similarity between live larvae and chitinous moults. It is noteworthy that the results obtained for the

head	model	layer	MAE	R^2	r
chitin	LR	3	$0.082_{\pm 0.024}$	$0.752_{\pm 0.128}$	$0.919_{\pm 0.031}$
chitin	KNN	9	$0.075_{\pm 0.023}$	$0.716_{\pm 0.185}$	$0.911_{\pm 0.036}$
chitin	SVR	4	$0.070_{\pm 0.018}$	$\mathbf{0.786}_{\pm 0.130}$	$0.947_{\pm 0.022}$
chitin	GBR	7	$\mathbf{0.062}_{\pm 0.025}$	$0.785_{\pm 0.169}$	$\mathbf{0.948}_{\pm 0.030}$
feed	LR	3	$\mathbf{0.042}_{\pm 0.008}$	$\mathbf{0.949}_{\pm 0.025}$	$\mathbf{0.983}_{\pm 0.007}$
feed	KNN	6	$0.065_{\pm 0.017}$	$0.857_{\pm 0.113}$	$0.938_{\pm 0.042}$
feed	SVR	0	$0.046_{\pm 0.009}$	$0.941_{\pm 0.026}$	$0.976_{\pm 0.012}$
feed	GBR	6	$0.065_{\pm 0.021}$	$0.850_{\pm 0.137}$	$0.945_{\pm 0.038}$
pheno	LR	6	$0.106_{\pm 0.005}$	$0.863_{\pm 0.012}$	$0.930_{\pm 0.007}$
pheno	KNN	9	$0.119_{\pm 0.008}$	$0.829_{\pm 0.023}$	$0.917_{\pm 0.010}$
pheno	SVR	6	$\mathbf{0.101}_{\pm 0.002}$	$0.868_{\pm 0.008}$	$0.932_{\pm 0.004}$
pheno	GBR	6	$0.103_{\pm 0.004}$	$\mathbf{0.869}_{\pm 0.012}$	$\mathbf{0.935}_{\pm 0.006}$

Table 2. Results for the tasks related to the proposed heads, i.e., chitin coverage estimation (chitin), feed coverage estimation (feed) and larvae phenotyping (pheno) using different settings (chosen machine learning models for prediction based on embeddings from a specific layer of the YOLO model).

phenotyping head are comparable with the results reported in [18], where a special architecture was used for the task of phenotyping larvae based on the ResNet18 model with fine-tuning of all model parameters. In the approach considered in this article, we assume frozen weights for the backbone.

8.3. Predictions with Proposed Heads

The evaluation results in the form of true versus predicted charts for the regression tasks of estimating feed coverage index and larvae phenotyping are shown in Fig. 6

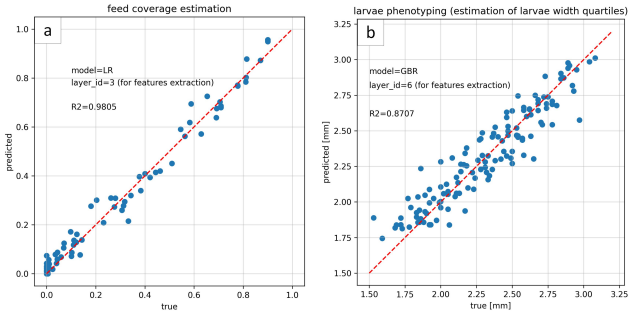

Figure 6. Comparative analysis of true vs. predicted values for selected regression tasks: (a) feed coverage estimation and (b) phenotyping based on the results from the selected cross-validation iteration.

The results in Fig. 6 confirm the validity of the developed approach to calculating the proposed indices and indicate the great potential of extracted features from the chosen backbone layers of the YOLOv8n model.

8.4. Model Ensemble and Uncertainty Estimation

The results for prediction in model ensemble mode for the Tenebrio Molitor state counting task are shown in Tab. 3 in the context of prediction efficiency and Tab. 4 for estimation of prediction uncertainty.

mode	class	MAE	R^2	r
single model	B	$0.212_{\pm 0.082}$	$0.949_{\pm 0.040}$	$0.981_{\pm 0.011}$
single model	DL	$1.134_{\pm 0.155}$	$0.898_{\pm 0.021}$	$0.955_{\pm 0.009}$
single model	P	$0.906_{\pm 0.166}$	$0.965_{\pm 0.023}$	$0.987_{\pm 0.008}$
ensemble(n=5)	B	$0.194_{\pm 0.055}$	$0.967_{\pm 0.013}$	$0.987_{\pm 0.006}$
ensemble(n=5)	DL	$1.004_{\pm 0.120}$	$0.924_{\pm 0.013}$	$0.965_{\pm 0.006}$
ensemble(n=5)	P	$0.759_{\pm 0.087}$	$0.979_{\pm 0.012}$	$0.991_{\pm 0.006}$
ensemble(n=10)	B	$0.189_{\pm 0.054}$	$0.970_{\pm 0.011}$	$0.988_{\pm 0.005}$
ensemble(n=10)	DL	$0.989_{\pm 0.114}$	$0.927_{\pm 0.012}$	$0.966_{\pm 0.006}$
ensemble(n=10)	P	$0.732_{\pm 0.08}$	$0.981_{\pm 0.011}$	$0.991_{\pm 0.006}$
ensemble(n=20)	B	$0.188_{\pm 0.054}$	$0.971_{\pm 0.011}$	$0.988_{\pm 0.005}$
ensemble(n=20)	DL	$0.982_{\pm 0.114}$	$0.929_{\pm 0.010}$	$0.966_{\pm 0.005}$
ensemble(n=20)	P	$0.718_{\pm 0.075}$	$0.981_{\pm 0.012}$	$0.991_{\pm 0.006}$

Table 3. Comparison of results for single-model and ensemble of models approaches for Tenebrio Molitor states counting.

mode	class	pred. in 95 PPU	d-factor
ensemble(n=5)	B	$0.928_{\pm 0.029}$	$0.119_{\pm 0.032}$
ensemble(n=5)	DL	$0.714_{\pm 0.040}$	$0.251_{\pm 0.045}$
ensemble(n=5)	P	$0.777_{\pm 0.046}$	$0.138_{\pm 0.019}$
ensemble(n=10)	B	$0.956_{\pm 0.026}$	$0.156_{\pm 0.034}$
ensemble(n=10)	DL	$0.811_{\pm 0.030}$	$0.328_{\pm 0.049}$
ensemble(n=10)	P	$0.859_{\pm 0.037}$	$0.180_{\pm 0.023}$
ensemble(n=20)	B	$0.975_{\pm 0.017}$	$0.185_{\pm 0.044}$
ensemble(n=20)	DL	$0.865_{\pm 0.010}$	$0.381_{\pm 0.053}$
ensemble(n=20)	P	$0.920_{\pm 0.019}$	$0.214_{\pm 0.023}$

Table 4. Results for prediction uncertainty estimation using model ensemble for Tenebrio Molitor states counting.

Based on the results in Tab. 3, it can be concluded that, as expected, using an ensemble of YOLOv8 models resulted in a significant increase in counting performance compared to the results achieved by single models. The optimal number of models for the ensemble is not obvious. On the one hand, increasing the number of models in the ensemble from 10 to 20 no longer resulted in a significant increase in prediction accuracy. On the other hand, based on the results in Tab. 4, we can see that using more models in an ensemble results in more accurate uncertainty estimation (a larger proportion of predictions bracketed by 95 PPU). Of course, this is also related to the larger d-factor associated with the size of the uncertainty interval. The final decision on the number of models for the ensemble should be made, taking into ac-

count the characteristics of the problems, that is, the cost of potential FP and FN errors.

8.5. Domain Shift Detection

The distributions of prediction uncertainties for samples from the defined domains are shown in Fig. 7

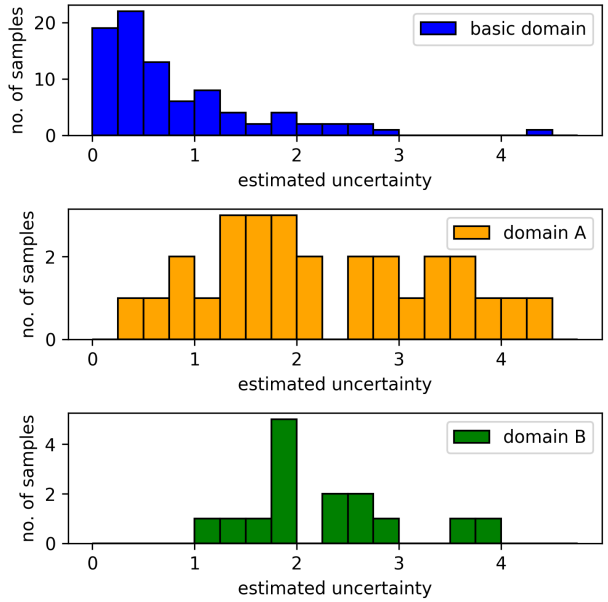

Figure 7. Comparison of the distributions of estimated uncertainty for samples from the base domain and samples from other domains (A and B).

In Fig. 7, we can see that considering the uncertainties for single samples, there is a noticeable overlap between the distributions for the defined domains. Considering the significant difference between the average uncertainty for the considered distributions, averaging the uncertainty values for a subset of samples of a certain size can significantly improve the separability of the distributions. We can find confirmation of this hypothesis in Tab. 5, where quantitative results are presented for detecting the domain shift phenomenon at a certain size of the subset of samples used to calculate the averaged uncertainty.

With 10 samples taken for averaged uncertainty, $F1 > 0.94$ was achieved for the two domains considered. Quantitative indicators confirm the validity of detecting the phenomenon of domain shift in the proposed way based on an increase in the value of prediction uncertainty. Carrying out a procedure for detecting the phenomenon of domain shift in production conditions for the problem posed also does not seem complicated, given the thousands of images recorded daily (which may represent a subset for averaging uncertainty).

set size	new domain	F1	precision	recall
1	A	$0.622_{\pm 0.071}$	$0.573_{\pm 0.067}$	$0.693_{\pm 0.118}$
1	B	$0.620_{\pm 0.104}$	$0.487_{\pm 0.086}$	$0.867_{\pm 0.163}$
5	A	$0.833_{\pm 0.140}$	$0.787_{\pm 0.181}$	$0.893_{\pm 0.088}$
5	B	$0.876_{\pm 0.123}$	$0.800_{\pm 0.187}$	$1.000_{\pm 0.000}$
10	A	$0.945_{\pm 0.078}$	$0.971_{\pm 0.057}$	$0.933_{\pm 0.133}$
10	B	$0.971_{\pm 0.057}$	$0.950_{\pm 0.100}$	$1.000_{\pm 0.000}$

Table 5. Results for domain shift detection for different sizes of the subset of samples used for averaged uncertainty calculation.

9. Conclusion and Future Work

The research proposed an end-to-end solution for calculating indicators to support the monitoring of Tenebrio Molitor rearing. Compared to previous approaches, multiple (separate) models trained for specific tasks were reduced to a single architecture based on a shared backbone. The extended YOLOv8 architecture with three problem-oriented heads made it possible to perform predictions for specific regression tasks. Training for each head was done separately, which made it possible to develop smaller datasets focused on defined object classes and significantly reduce the time spent on labelling. The proposed solution is flexible and allows rapid architecture extension for new problems by adding the following heads. Using an ensemble of models made it possible to increase the accuracy of prediction and estimate the uncertainty of prediction, which will increase the reliability of the developed solution and facilitate critical decision-making on the farm.

Future work should focus on multi-task learning problems, making it possible to jointly learn the separated heads of the architecture. Given the dependencies between the calculated indicators (e.g., the occurrence of pupae is related to a certain size of larvae), this approach seems reasonable.

Acknowledgements

We wish to thank Professor Ta-Te Lin (Department of Biomechatronics Engineering, National Taiwan University, Taiwan, ROC) for his valuable comments on the developed solution and review. We wish to thank Paweł Górzyński and Dawid Biedrzycki from Tenebria (Lubawa, Poland) for providing a data source of boxes with Tenebrio molitor. The work presented in this publication was carried out within the project "Automatic mealworm breeding system with the development of feeding technology" under Sub-measure 1.1.1 of the Smart Growth Operational Program 2014–2020 co-financed from the European Regional Development Fund on the basis of a co-financing agreement concluded with the National Center for Research and Development (NCBiR, Poland); grant POIR.01.01.01-00-0903/20.

References

[1] Chetan Badgujar, Daniel Flippo, and Stephen Welch. Artificial neural network to predict traction performance of autonomous ground vehicle on a sloped soil bin and uncertainty analysis. *Computers and Electronics in Agriculture*, 196: 106867, 2022. 6

[2] Ayan Banerjee, Palaiahnakote Shivakumara, Saumik Bhattacharya, Umapada Pal, and Cheng-Lin Liu. An end-to-end model for multi-view scene text recognition. *Pattern Recognition*, 149:110206, 2024. 2

[3] Karol B Barragan-Fonseca, Marcel Dicke, and Joop JA van Loon. Nutritional value of the black soldier fly (hermetia illucens l.) and its suitability as animal feed–a review. *Journal of Insects as Food and Feed*, 3(2):105–120, 2017. 1

[4] Yan Cang, Hengxiang He, and Yulong Qiao. An intelligent pig weights estimate method based on deep learning in sow stall environments. *IEEE Access*, 7:164867–164875, 2019. 2

[5] Wenbai Chen, Mengchen Liu, ChunJiang Zhao, Xingxu Li, and Yiqun Wang. Mtd-yolo: Multi-task deep convolutional neural network for cherry tomato fruit bunch maturity detection. *Computers and Electronics in Agriculture*, 216:108533, 2024. 2

[6] Zheng Chu and Jiong Yu. An end-to-end model for rice yield prediction using deep learning fusion. *Computers and Electronics in Agriculture*, 174:105471, 2020. 2

[7] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995. 4

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 2

[9] Jonathan A Foley, Navin Ramankutty, Kate A Brauman, Emily S Cassidy, James S Gerber, Matt Johnston, Nathaniel D Mueller, Christine O'Connell, Deepak K Ray, Paul C West, et al. Solutions for a cultivated planet. *Nature*, 478(7369):337–342, 2011. 1

[10] Jerome H Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001. 4

[11] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2

[12] Thorben Grau, Andreas Vilcinskas, and Gerrit Joop. Sustainable farming of the mealworm tenebrio molitor for the production of food and feed. *Zeitschrift für Naturforschung C*, 72(9-10):337–349, 2017. 1

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[14] Glenn Jocher, K Nishimura, T Mineeva, and R Vilariño. yolov5. *Code repository https://github.com/ultralytics/yolov5*, page 9, 2020. 1, 2

[15] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8, 2023. 1

[16] Paweł Majewski, Piotr Zapotoczny, Piotr Lampa, Robert Burduk, and Jacek Reiner. Multipurpose monitoring system for edible insect breeding based on machine learning. *Scientific Reports*, 12(1):7892, 2022. 1, 2, 6

[17] Paweł Majewski., Piotr Lampa., Robert Burduk., and Jacek Reiner. Mixing augmentation and knowledge-based techniques in unsupervised domain adaptation for segmentation of edible insect states. In *Proceedings of the 18th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2023) - Volume 5: VISAPP*, pages 380–387. INSTICC, SciTePress, 2023. 3

[18] Paweł Majewski, Mariusz Mrzygłód, Piotr Lampa, Robert Burduk, and Jacek Reiner. Monitoring the growth of insect larvae using a regression convolutional neural network and knowledge transfer. *Engineering Applications of Artificial Intelligence*, 127:107358, 2024. 1, 3, 4, 7

[19] Puneet Mishra and Dário Passos. Multi-output 1-dimensional convolutional neural networks for simultaneous prediction of different traits of fruit based on near-infrared spectroscopy. *Postharvest Biology and Technology*, 183: 111741, 2022. 2

[20] Sarah Nawoya, Frank Ssemakula, Roseline Akol, Quentin Geissmann, Henrik Karstoft, Kim Bjerge, Cosmas Mwikirize, Andrew Katumba, and Grum Gebreyesus. Computer vision and deep learning in insects for food and feed production: A review. *Computers and Electronics in Agriculture*, 216:108503, 2024. 1

[21] Dennis GAB Oonincx and Imke JM De Boer. Environmental impact of the production of mealworms as a protein source for humans–a life cycle assessment. *PloS one*, 7(12):e51145, 2012. 1

[22] JA Cortes Ortiz, A Torres Ruiz, JA Morales-Ramos, M Thomas, MG Rojas, JK Tomberlin, L Yi, R Han, L Giroud, and RL Jullien. Insect mass production technologies. In *Insects as sustainable food ingredients*, pages 153–201. Elsevier, 2016. 1

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 1

[24] Arnold Van Huis, Joost Van Itterbeeck, Harmke Klunder, Esther Mertens, Afton Halloran, Giulia Muir, and Paul Vantomme. *Edible insects: future prospects for food and feed security*. Number 171. Food and agriculture organization of the United Nations, 2013. 1

[25] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-yolov4: Scaling cross stage partial network. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pages 13029–13038, 2021. 4

[26] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7464–7475, 2023. 2

[27] Yihan Wang, Xinglong Deng, Jianqiao Luo, Bailin Li, and Shide Xiao. Cross-task feature enhancement strategy in

multi-task learning for harvesting sichuan pepper. *Computers and Electronics in Agriculture*, 207:107726, 2023. 2

[28] Dong Wu, Man-Wen Liao, Wei-Tian Zhang, Xing-Gang Wang, Xiang Bai, Wen-Qing Cheng, and Wen-Yu Liu. Yolop: You only look once for panoptic driving perception. *Machine Intelligence Research*, 19(6):550–562, 2022. 2

[29] Shuai Xiang, Siyu Wang, Mei Xu, Wenyan Wang, and Weiguo Liu. Yolo pod: a fast and accurate multi-task model for dense soybean pod counting. *Plant methods*, 19(1):8, 2023. 2

[30] Fan Zhang, Jin Gao, Chaoyu Song, Hang Zhou, Kunlin Zou, Jinyi Xie, Ting Yuan, and Junxiong Zhang. Tpmv2: An end-to-end tomato pose method based on 3d key points detection. *Computers and Electronics in Agriculture*, 210:107878, 2023. 2

[31] Jianlong Zhang, Yanrong Zhuang, Hengyi Ji, and Guanghui Teng. Pig weight and body size estimation using a multiple output regression convolutional neural network: A fast and fully automatic method. *Sensors*, 21(9):3218, 2021. 2

[32] Lei Zhang, Haisheng Li, Ruijun Liu, Xiaochuan Wang, and Xiaoqun Wu. Weakly supervised end-to-end domain adaptation for person re-identification. *Computers and Electrical Engineering*, 113:109055, 2024. 2