Journal of Cleaner Production 468 (2024) 143024

Contents lists available at ScienceDirect

Journal of Cleaner Production

journal homepage: www.elsevier.com/locate/jclepro

Robust deep learning based shrimp counting in an industrial farm setting

Christina Bukas [a,1], Frauke Albrecht [b,1], Muhammad Saeed Ur- Rehman [c,1], Daniel Popek [d], Mikołaj Patalan [d], Jarosław Pawłowski [d,e], Bert Wecker [f], Kilian Landsch [f], Tomasz Golan [d], Tomasz Kowalczyk [d], Marie Piraud [a], Stephan S.W. Ende [c,*]

[a] Helmholtz AI, Helmholtz Zentrum München, Ingolstädter Landstr. 1, 85764, Neuherberg, Germany
[b] German Climate Computing Center DKRZ, Hamburg, Germany
[c] Alfred-Wegener-Institut Helmholtz-Zentrum für Polar- und Meeresforschung, Bremerhaven, Germany
[d] NeuroSYS, Rybacka 7, 53-656, Wrocław, Poland
[e] Faculty of Fundamental Problems of Technology, Wroclaw University of Science and Technology, Wybrzeże S. Wyspiańskiego 27, 50-372, Wrocław, Poland
[f] Oceanloop Kiel GmbH & Co. KG, Bülker Huk, 24229, Strande, Germany

ARTICLE INFO

Handling Editor: Cecilia Maria Villas Bôas de Almeida

Keywords:
Aquaculture
Artificial intelligence
Shrimp
Counting
Welfare

ABSTRACT

Shrimp production is one of the fastest growing sectors in the aquaculture industry. Despite extensive research in recent years, stocking densities in shrimp systems still depend on manual sampling which is neither time nor cost efficient and additionally challenges shrimp welfare. This paper compares the performance of automatic shrimp counting solutions for commercial Recirculating Aquaculture System (RAS) based farming systems, using eight Deep Learning based methods. The entire dataset includes 1379 images of shrimps in RAS farming tanks, taken at a distance using an iPhone 11 mini. These were manually annotated, with bounding boxes for every clearly visible shrimp. The dataset was partitioned into training (60 %, 828 samples), validation (20 %, 276 samples) and test (20 %, 275 samples) splits for purposes of training and evaluating the models. The present work demonstrates that state-of-the-art object detection models outperform manual counting and achieve high performance across the entire production range and at various circumstances known to be challenging for object detection (dim light, overlapping and small animals, various acquisition devices and image resolutions and camera distance to object). Highest counting performance was obtained with models based on YOLOv5m6 and Faster R–CNN (as opposed to neural network autoencoder architecture to estimate a density map). The best model generalizes well on an independent test set and even shows promising performance when tested with different taxa. The model performs best at densities below 200 shrimps per image with an overall error of 5.97 %. It is assumed that this performance can be improved by increasing the dataset size, especially with images at high shrimp stocking density, and it is strongly believed that a performance below the 5 % error threshold is close to being achieved, which will allow for deployment of the model in an industrial setting.

1. Introduction

Despite a substantial increase in intensive Recirculating Aquaculture System (RAS) based farming in Europe, animal welfare and financial issues remain. Indeed, the monitoring of important production parameters, such as biomass, health status, feeding rates and growth of the animals, is still largely manual. The farm manager invests about 2 h per week in manpower (coinciding with 300 €/month). Beyond being extremely time consuming and error prone, it causes stress and sometimes physical damage to the animals. In a recent review entitled on Welfare of Decapod Crustaceans the authors summarize that behaviour of the animals suggests that decapod crustaceans experience nociception and there are also several indications of pain perception Wuertz et al. (2023) .

Monitoring relies primarily on the number of individuals per tank and their respective lengths (which correlates to weights). However, efforts to automatically estimate these numbers in shrimp farming have failed in commercial settings. In the meantime, counting shrimps using image processing algorithms is not a recent concept. Khantuwan and Khiripet (2012) introduced an automatic method for counting shrimp

[*] Corresponding author.
E-mail address: sende@awi.de (S.S.W. Ende).
[1] Joint-first authorship with equal contribution.

https://doi.org/10.1016/j.jclepro.2024.143024
Received 21 May 2024; Accepted 24 June 2024
Available online 4 July 2024

larvae using histogram and template matching. This method was able to achieve 97 % accuracy. A similar counting performance is reported in other shrimp larvae studies based upon image processing techniques (Awalludin et al., 2019; Boksuwan et al., 2018; Kaewchote et al., 2018). Unfortunately, these methods were based upon classical image processing, whose performance drastically decreases in overlapping scenarios. For example, in Kaewchote et al. (2018), a large root mean square error of 14.43 shrimps per image (showing on average 164 shrimp larvae, data recalculated from manual counting numbers provided in Table 2 by authors) is reported, which is mainly due to the overlapping of shrimps, even after transferring the animals to counting buckets. Additionally, all the above approaches required manual thresholds to achieve satisfactory results.

Deep Learning (DL) models can be more resilient in uncontrolled conditions, when provided with enough annotated images. Such models are therefore better suited for industrial scenarios, as seen for example in applied microbiology (Majchrowska et al., 2021, 2022; Pawłowski et al., 2022). YOLOv3 based models used for counting shrimp larvae in a controlled lab environment obtained a test F1-score of 94.38 % on a test set of 30 images, but failed to detect shrimps in overlapping scenarios (Armalivia et al., 2021). A two-phase Mask R–CNN for instance segmentation of shrimp larvae achieved a counting accuracy of 72.9 % in high overlapping scenarios (Nguyen et al., 2020). Similarly, in another study, Faster R–CNN based on shrimp key body part detection, i.e., head and stomach, had a mean percentage error of 23.8 % in overlapping scenarios (Hashisho et al., 2021). The model was evaluated on a test set containing 30 images.

However, all the above-mentioned methods, based on standard image processing techniques and DL applications alike, count shrimps in images taken in a controlled environment with ideal settings, such as light, contrast (e.g., white background), water level or numbers of shrimps, in order to optimize image quality and avoid overlapping scenarios. For this purpose, shrimp larvae needed to be removed from the cultivation system and placed in separate buckets. In Kaewchote et al. (2018), for example, images were taken in small beakers with low water level, a high background contrast against the shrimp and additional lighting, at a camera distance of a few centimeters (Fig. 1a). The estimated number would then need to be extrapolated to the whole tank, thus introducing additional errors. Moreover, most methods focus on a specific shrimp age and size, thereby not covering the entire production range. Such limited conditions do not represent the industrial environment and cannot be met in on-growing shrimp systems using clear water technology and different lighting conditions, where acquired images of the tank have lower quality and the contrast of shrimps against the background is reduced (Fig. 1b). An additional challenge on farms, where tanks have a high-water level, is that the distance of the shrimps to the camera changes and therefore shrimps swimming on the water surface appear bigger, while those sitting on the net appear smaller. In this scenario, thresholding techniques completely fail because swimming shrimps could be considered outliers compared to sitting shrimps and vice versa. Finally, all methods reported increasing error rates with increasing overlapping. In real farm systems, the number of shrimps on an image and their overlap cannot be controlled and algorithms need to be developed to overcome such complications. In such a setup, shrimp detection and counting remains a challenging task.

To summarize, all the methods presented in the literature counted shrimp in the lab environment with ideal settings that are impossible to achieve in an industrial environment. The following conditions were identified to be most challenging for automatic object detection in a farm environment:

1. Overlapping shrimps due to high stocking densities and high-water level
2. Diffused background color and reduced contrast to shrimp object
3. Non-homogeneous lighting conditions, and farm light reflection on the water surface

In the present work eight DL based methods trained on images acquired in an industrial RAS farm were tailored and benchmarked, and performances of object detection vs Density Map (DM) based models at different environmental settings were compared (blue and white light, 80 and 120 cm camera height). A solution is proposed that tackles shrimp counting in an industrial environment on shrimp sizes representing the entire production range (from 3.4 g to 29 g). It is shown that all but one of the models clearly outperforms the current manual standard for counting, estimated by aqua farming experts at an error of approximately 20 % (pers. comm. B. Wecker, Oceanloop Kiel GmbH & Co. KG). Furthermore, a 5 % error target was set for automatic counting approaches to be accepted and integrated in farm setups and it is shown that the best models achieve a performance very close to it. The 5 % error target is based on several observations, which are largely derived from practical farm experience. Firstly, the current methods used for determining biomass consist of weighing individual shrimps and manually counting images and are therefore only carried out about once a month due to the high effort involved. Moreover, the number of measured data collected in these cases does not adequately reflect the high variance observed in the farms. Finally, the accuracy of the manually determined number of shrimps can be verified by the final number of shrimps harvested. Some of the results show a deviation in the data of up to 20–25 %. A 5 % error target is therefore already a significant improvement to existing techniques and sufficient from an economic point of view. This is especially true for the setting of an optimized feeding rate, which accepts an under- or overestimation of the biomass of 5 %, since the underfeeding is usually in the range of 10–20 % to optimize the feed conversion, and the error of 5 % plays only a minor role. It is assumed that this current 5 % error threshold can be overcome in future work by collecting a larger dataset to retrain the present model. In addition, it is demonstrated that the present model generalizes to an independently collected dataset, displaying reduced performance only for images out of the distribution of the original training set.

2. Materials and methods

In the sections below the process of data acquisition, annotation and preparation is described in order to collect an imaging dataset depicting

Table 1
Comparison of overall best performing models in terms of shrimp counting error. In bold the best results per metric for which the Wilcoxon signed-rank tests showed no significant statistical difference.

No.	approach	model	variant	MAE (counts)	MAPE (%)	sMAPE (%)
Model 1	*detection-based*	*Faster R–CNN*	*globally tuned NMS and confidence thresholds*	**5.48**	**6.46**	**6.59**
Model 2	*detection-based*	*Faster R–CNN*	*adjusted NMS and confidence thresholds per category*	**5.03**	**6.37**	**6.47**
Model 3	*detection-based*	*YOLOv5m6*	*globally tuned NMS and confidence thresholds*	**4.70**	**6.03**	**6.05**
Model 4	*detection-based*	*YOLOv5m6*	*adjusted NMS and confidence thresholds per category*	***4.66***	***5.97***	***6.01***
Model 5	*DM-based*	*U^2-Net*	*bare model*	12.97	22.34	18.15
Model 6	*DM-based*	*U^2-Net*	*with self-normalization*	9.86	15.19	13.45
Model 7	*DM-based*	*U^2-Net*	*with blobs counting*	8.33	10.42	10.27
Model 8	*DM-based*	*U^2-Net*	*with regression CNN network*	6.65	9.77	9.35

Table 2
Comparison of MAPE error for Model 1 (Faster R–CNN) and Model 3 (YOLOv5m6) on the independent test set. In bold the statistically significant better performing model for each category ($P < 0.01$).

distance	images #	avg. shrimp number/image	Faster R–CNN error [MAPE]	YOLO error [MAPE]
all categories	*90*	*164*	19.14	**11.59**
100 cm	*30*	*100*	7.33	7.39
125 cm	*30*	*182*	20.12	**10.44**
185 cm	*30*	*209*	29.98	**16.94**

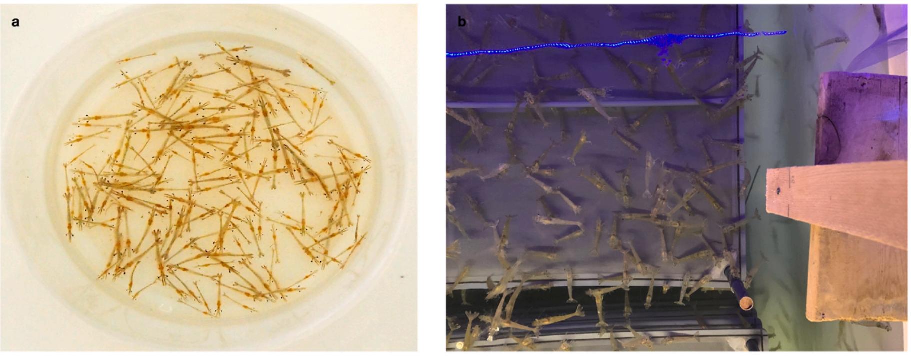

Fig. 1. a) Images taken under experimental conditions with camera (type unspecified) placed above a small beaker, with white background (Source: Kaewchote et al. (2018)). b) Image taken at a commercial shrimp farm with a smartphone (iPhone 11 mini), at realistic light conditions, in the presence of horizontal ‘mangroves’ nets, and at commercial stocking density. The image was taken at 80 cm above water (Source: Oceanloop Kiel GmbH & Co. KG).

shrimps in RAS farming tanks which can be used to train a deep learning model to automatically estimate the number of shrimps per tank. The models training process, hyperparameter tuning and model selection are outlined, followed by the testing and evaluation of the different approaches.

2.1. Dataset

The entire dataset includes 1379 images of shrimps in RAS farming tanks, taken at a distance using an iPhone 11 mini. These were manually annotated, with bounding boxes for every clearly visible shrimp using the open-source Label Studio tool (Tkachenko et al., 2020). The Label Studio was self-hosted on a local server. The annotation setup was configured as a rectangular annotation with a single class ‘shrimp’. The whole dataset was manually annotated by a single annotator, however, the samples with high annotation density (above 100 bounding boxes) were further validated and corrected by another two annotators. The data was acquired under four different light-distance categories: 80 cm and 120 cm distance to the tank, with either white or blue light (See Fig. S1). The average number of shrimps per image for the entire dataset is 73 (min: 14, max: 292), while the average shrimp count for category 80 cm/blue is 42 (min: 14, max: 98), 56 (min: 23, max: 133) for 120 cm/blue, 61 (min: 16, max: 150) for 80 cm/white and 120 (min: 35, max: 292) for 120 cm/white. A higher density is observed in the 120 cm/white category, which can be partially explained by the 120 cm camera distance, meaning that the images cover a larger area of the tank (See Fig. S2 for the distribution of bounding boxes per light-distance category). The dataset was partitioned into training (60 %, 828 samples), validation (20 %, 276 samples) and test (20 %, 275 samples) splits for purposes of training and evaluating the models. Stratified sampling was applied for the partitioning to preserve the original distribution of light-distance categories (Fig. 2a). Based on the bounding box annotations, density maps were prepared according to the following procedure: mask image was created with ones in the locations of bounding box centers and zeros elsewhere. In order to obtain the density map, the mask was convolved with a 2D Gaussian filter with a sigma parameter (σ) equal to eight. The sum of a single Gaussian distribution was equal to one, therefore the sum of pixels on the density map was equal to the number of shrimps. In this way, two types of ground truth annotations were generated for each image (Fig. 2b, c), which are independently

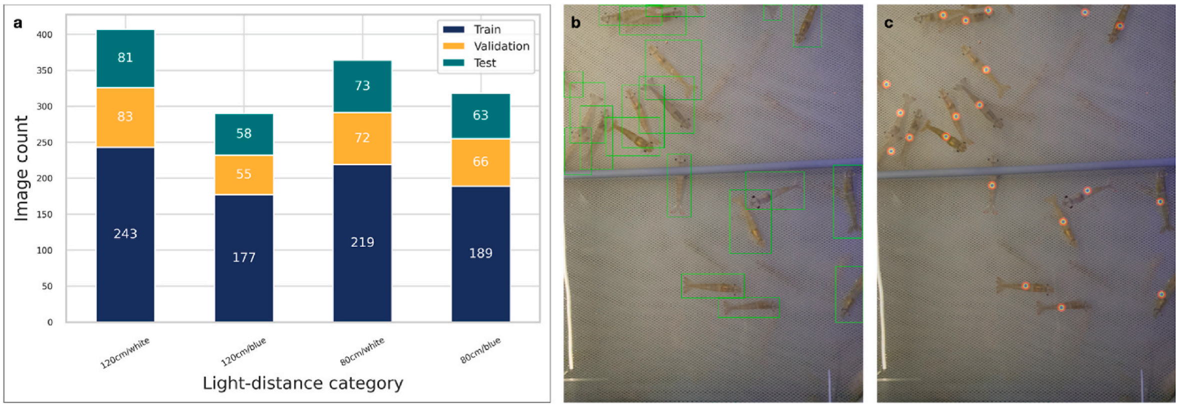

Fig. 2. a) The number of images divided into four light-distance categories with stacked train, validation and test splits. Numbers seen on the individual bar plots indicate the number of images in each dataset partition. b) Visualization of exemplary annotations of images with bounding boxes and c) Density maps.

used to train our various models. For details on the acquisition, labeling and more statistics please refer to the Supplementary Information A.

Independent test set: An additional dataset was used to test the generalizability of the models, containing 90 samples collected with an iPhone 11 mini and with three distance categories under both light conditions (100 cm, 125 cm and 185 cm, with 30 samples per category). The images were extracted from videos, with a sufficient window between frames to ensure that the shrimps had changed location, signifying that these images have lower resolution compared to the main dataset. Additionally, this dataset differs significantly from the main dataset in terms of average number of shrimps per image (164 shrimps with min: 66 and max: 266 vs. 73 shrimps). To generate the ground truth for this dataset, the CVAT platform was used (Sekachev et al., 2020), which provides the option for single-click annotations.

The images used in this study were collected at Oceanloop Kiel GmbH & Co. KG farm (Strande, Germany). Animals were photographed at a distance in a holding system registered for animal production (Registering body SH plus Zulassungsnummer) and no handling or stress was encountered or possible during the acquisition. The authors and the experimental image collection comply fully with ARRIVE guidelines.

2.2. Models

A two-pronged approach was used for determining the number of shrimps in an image and trained two types of bounding box detection algorithms, which was named detection-based approaches, as well as DM-based models. Counting via object detection was realized with two main architecture types, namely one-stage and two-stage models, as depicted in Fig. 3a and 3b. In contrast to one-stage models, which perform a straightway classification of objects and regression of bounding boxes within a single feed-forward pass of the network, two-stage models break down the generation into two phases: first the generation of candidate regions is realized via a Region Proposal Network (RPN) and then classification and regression is performed for each proposed region. Faster R–CNN (Ren et al., 2016), deriving from the R–CNN (Girshick et al., 2014) family, was trained as a representative of the two-stage models group and YOLOv5m6 as a well-performing one-stage model was used (Jocher et al., 2021). YOLOv5m6 is a version of YOLO and was chosen here due to its augmentation techniques, namely mosaic and mix-up image transformations, which greatly contributes to model generalization (Zhang et al., 2018). For details on the architectures please refer to the Supplementary Information.

Another approach for object counting is to use a neural network autoencoder architecture to estimate a DM that represents objects on a given image. The main idea is to transform an input image to the normalized map of the density of the objects, so that the number of objects can be established as the integral over the map. This approach is most commonly used for crowd counting (Lempitsky and Zisserman, 2010; Zhang et al., 2015), but also for biological objects (Arteta et al., 2014; Graczyk et al., 2022; Xie et al., 2018). Several approaches were applied for predicting the number of shrimps with DMs: bare Gaussian-kernel-based DMs (Jiang and Yu, 2020), self-normalizing DMs (SNDMs) (Graczyk et al., 2022), DMs with blob-based counting and DMs with an additional custom CNN-based regression network (Fig. 3d). All the aforementioned models used the U^2-Net backbone (Qin et al., 2020), which is depicted in Fig. 3c.

The models were initially trained on the training set and determined the hyperparameters of the models on the validation set. The tuned hyperparameters are different for detection and DM-based models. In the case of detection-based models the following hyperparameters were tuned: the threshold of Non-Maximum Suppression (NMS) and the bounding boxes acceptance threshold, later referred to as the confidence threshold. Tuning was performed of the thresholds both: globally (with no distinction of light-distance categories) and individually for each light-distance category, resulting in a total of four detection-based models. In the case of DM models, the tuned hyperparameters were the Gaussian-kernel's standard deviation (σ) and the normalization factor in the case of SNDMs. Having established the hyperparameters values, the final models were retrained on the combined training and validation sets (80 % of the dataset). For training the detection and DM-based models, the same loss functions were used as in the original publications. In order to track the training progress on the prediction of bounding boxes (Faster R–CNN and YOLOv5m6), the mean Average Precision (mAP) was used at 0.5:0.95 IoU thresholds, mAP@.5:.95 (Everingham et al., 2010) (see also Supplementary Information B). The final models were evaluated on the 20 % test set, unseen during training and validation. In the following sections the training configurations that were used for the various models are briefly summarized. For a more detailed description of the variants of the models and their

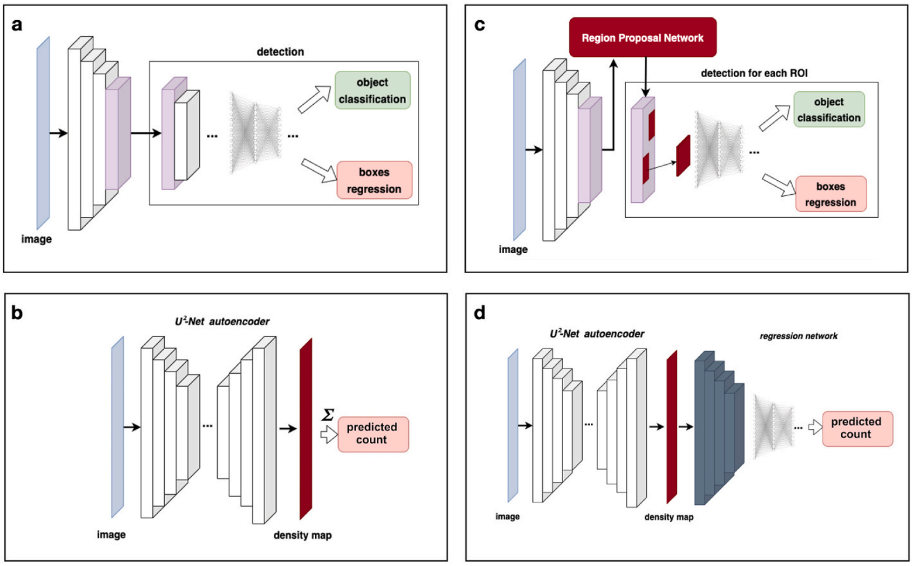

Fig. 3. Schematic representation of the detection-based (A, B) and DM-based models (C, D). a) One-stage object detection models perform direct classification and regression of bounding-boxes b) Two-stage detection models use additional RPN for generation of candidate regions. c) Autoencoder neural network (U^2-Net) predicts DMs which can be later integrated (Σ symbol) to count the number of objects. d) The custom approach uses an additional CNN-based regression network for integration of the maps.

hyperparameters please refer to the Supplementary Information and to the cited literature.

Faster R–CNN*:* Both Faster R–CNN models (with global and per category NMS thresholds) were trained based on the Res-Next-101 backbone (Xie et al., 2017) and pre-trained on MS-COCO (Lin et al., 2014). At the preprocessing stage input images were resized to 1333x800x3 and were padded to keep the original dimensions ratio. Additionally, RGB-color-space normalization was used and images were flipped with a 0.5 probability. The models were trained for 12 epochs with stochastic gradient descent (SGD), a batch size of two, 2.5e-3 learning rate, 0.9 momentum and 0.0001 wt decay. A single NVIDIA GeForce RTX Titan GPU with 24 GB of VRAM was used for training.

YOLOv5m6: The YOLOv5m6 model (with global and per category NMS thresholds), a variant of YOLOv5 pre-trained on the COCO dataset, was used in this study. The model was trained for 80 epochs with a batch size of eight. Input images were resized and padded to a size of 2048x2048x3. The following data augmentations were applied: mixup and mosaic data enhancement, HSV color varying, image scaling and translation, left-right flip. SGD was used as an optimizer with 0.01 learning rate, 0.937 momentum and 0.0005 wt decay. The models were trained on a single NVIDIA Tesla A40 GPU**.**

DM-based models: As the baseline for the DM-based models an autoencoder network was trained. Among several types of neural networks used for DM estimation, the U-Net architecture (Ronneberger et al., 2015) is most commonly applied (Jiang and Yu, 2020). However, it was observed that the U^2-Net architecture performed better than the U-Net in this case. The U^2-Net was trained for 100 epochs using the SGD optimizer with 2e-3 learning rate and a batch size of one. The models were trained on a single NVIDIA GeForce RTX Titan GPU with 24 GB of VRAM.

Self-normalization variant: During the experiments it was often found that the network predicted the position of objects correctly, but after taking the integral of the DM, the predicted count was underestimated. To remedy this, the self-normalization mechanism to the U^2-Net, with the same training hyperparameters as in the base autoencoder was applied (see Supplementary Information F for more details).

Blob detection variant: Another solution in which, instead of simple integration of the output DM, blob detection was performed on the DM using the Difference-of-Gaussian-based algorithm was tested (Lowe, 2004). The final prediction of the objects' number can be obtained by counting the detected blobs. This approach was applied to the trained baseline autoencoder.

Regression network variant: A custom regression network was developed with the aim to perform better summation of the DM instead of using the simple pixel-wise sum (Fig. 3D). For this purpose, a shallow CNN network was applied, followed by fully connected layers that were trained to perform regression on DMs predicted by the U^2-Net model. The regression part and autoencoder part were trained separately. The final regression network's architecture had a simple yet effective design (see Supplementary Information F). The Mean Squared Error was used as the regression loss function, which proved to perform better than the Mean Absolute Error (MAE). The model was trained with the Adam optimizer, for 10 epochs, with 1e-3 learning rate and a batch size of one.

2.3. Evaluation

The final validation of the approaches, either detection or DM-based, was performed on the test set using evaluation metrics for the task of counting objects, i.e. shrimps. In addition to the standard counting metric, MAE, also relative errors were used to account for the distribution of ground truth counts: the Mean Absolute Percentage Error (MAPE) and the symmetric Mean Absolute Percentage Error (sMAPE, see Supplementary Material B).

To select the optimal model for shrimp detection a non-parametric statistical tests was used. Each model's performance was compared with respect to the Absolute Error, the Absolute Percentage Error (APE) and the symmetric Absolute Error between the true and predicted number of shrimps in each image of the test set. First, a Fligner-Killeen test was performed to ensure that the distributions have similar variances (Fligner and Killeen, 1976), and then applied a pairwise Wilcoxon signed-rank test for each model pair (Wilcoxon, 1945). The choice of test was based upon the observation that: i. Comparing paired samples of error rates deriving from the same test set and ii. The distributions were skewed, which violates the normality assumption of the one-way ANOVA. A threshold value of $P = 0.01$ was set to determine whether the p-value between the pair's distribution is statistically significant.

3. Results

Table 1 shows results on the test set for the eight models which were benchmarked. The best performing model is Model 4, YOLOv5m6 with adjusted NMS and confidence thresholds per scenario, achieving the lowest MAPE of 5.97 %. Model 3, again a YOLOv5m6 but with global thresholds, performs slightly worse with an MAPE of 6.03 %. Similarly, for the Faster R–CNN models, Model 2 achieves a slightly lower error compared to Model 1 (MAPE of 6.37 % vs. 6.46 %). DM-based models obtained results on the level of 9.77 % MAPE.

Statistical significance tests between model pairs showed no significant difference between any of the detection-based models ($P > 0.01$). However, the pairwise Wilcoxon signed-rank tests showed that the difference between each detection-based model and each DM-based model were statistically significant (detection models were always superior, $P < 0.01$). Additionally, from the DM-based approaches, Model 7 and Model 8 were significantly better than the two remaining DM-based models, while Model 6 performed significantly better than Model 5. Table S1 in the Supplementary Information gives detailed P values for each test performed on the APE distributions.

3.1. Best performing models

In this section the counting performance of the two best models based on YOLOv5m6 and Faster R–CNN is presented (Models 4 and 2 respectively). It was observed that both architectures, YOLOv5m6 and Faster R–CNN, perform comparably well. All models had test-time NMS and confidence thresholds tuned on the validation dataset (see Supplementary Information C). Models with adjusted NMS and confidence thresholds per category proved to perform better than the globally adjusted hyperparameters. However, the difference between model performance with globally adjusted thresholds and with per light-distance category thresholds for the YOLOv5m6 models (Models 3 and 4) was relatively small, around 0.06 % MAPE.

The counting performance of Model 2 and Model 4 is visible in Fig. 4. As can be seen, both models perform well on the majority of images, predicting them within the ±10 % error margin. Both models output more predictions that are outside this margin for higher object density images (120 cm/white category) than for the rest of the data.

The best performing models were also compared regarding inference time on the whole test dataset (275 images with dimensions 1536 × 2048 and 73 objects on average). The test was run on both an NVIDIA A40 GPU and a AMD EPYC 7252 8-Core CPU Processor. For Faster R–CNN the following performance was achieved: 270 ms per image on a single GPU (2.5 GB VRAM allocated) and 4.2 s per image on the CPU. For YOLOv5m6 the performance was significantly better: 15 ms per image on a single GPU (2 GB VRAM allocated) and 130 ms per image on CPU. Further model optimizations are possible for these models' architectures (e.g. conversion to TensorRT) that should allow for real-time inference with satisfactory FPS (frames per second).

3.2. Models performance on the independent test set

In order to assess the ability of the models to generalize to changing environmental conditions, Faster R–CNN (Model 1) and YOLOv5m6

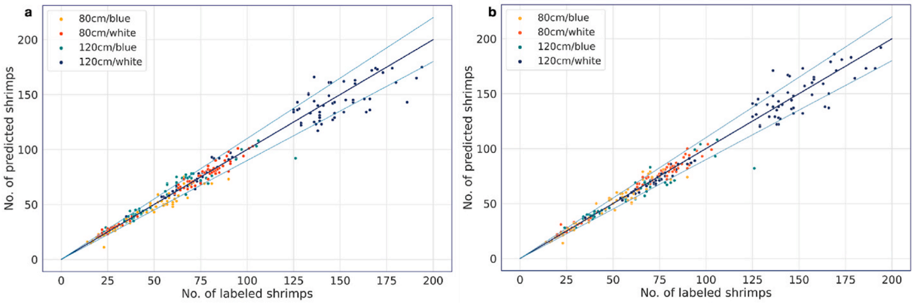

Fig. 4. The relation between ground truth and prediction count for each image in the test set for a) Model 2 (Faster R–CNN) and b) Model 4 (YOLOv5m6). The light blue lines mark the ±10 % error margin. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

(Model 3) were tested on an independent test set (see Section 2.1.). These models were chosen, as opposed to Models 2 and 4, because the external dataset was acquired at different heights than the main dataset.

It is clearly visible (Fig. 5) that both models perform reasonably well on images of the independent test set with a shrimp count below 200 (62 samples with an average of 131 shrimps per image, see Table S2), which is closer to the average shrimp count of the original test set. However, at greater camera heights, i.e., more shrimps per image, Faster R–CNN's results are visibly worse due to the higher average shrimp density when compared to the training set. Overall, the YOLOv5m6 performs significantly better than Faster R–CNN across all distance categories as can be seen on Table 2. Especially noticeable is the difference at higher distance categories, where YOLOv5m6 outperforms Faster R–CNN by about 10 % on MAPE. An example of YOLOv5m6 predictions on an image of this dataset is visualized in Fig. 6.

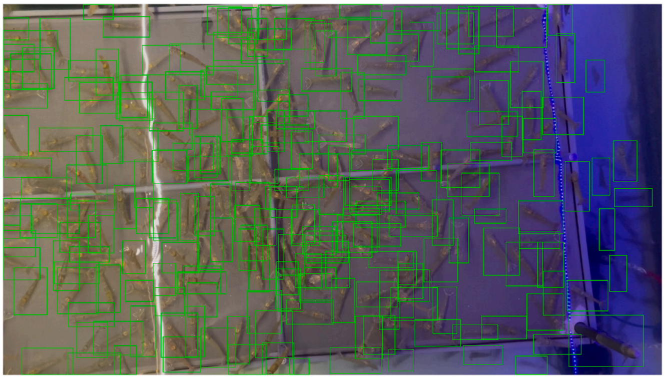

Fig. 6. Visualization of Model 3 (YOLOv5m6) predictions on a representative image of the independent test set acquired at a 185 cm camera distance (predicted value is 237, ground truth is 255 APE of 7.06 %).

4. Discussion

This study provides the first evidence of automatic shrimp counting with high performance under real farm conditions. All models showed low error rates and fast processing times of only a few seconds, critical for fast decision making and intervention, for example when a decreasing number of animals is detected. In addition, shrimps can hereby remain in their original environment, which was not the case in previous studies. Moreover, this application does not require any complicated setup, merely a smartphone with a camera and a user who follows the provided guidelines when taking images (height, angle, image in focus).

In the field of shrimp biomass determination, there are currently only a few companies that can determine both the number of individuals and the average weight. XpertSea (2024), a Canadian company, focuses primarily on determining length, average weight and density. Shrimps are photographed in a predefined volume against a white background and their length is automatically determined. The weight can then be surmised from the length-weight ratio. The disadvantage of this method is that it cannot be used directly in the pond or tank, but the shrimp must be transferred to a measuring device. The measured sample will accurately reflect the real biomass if the sample is taken from a homogeneous population, such as in larval tanks. However, the technology cannot be used in larger tanks or for larger individuals. The start-up company Sincere Aquaculture (2024), based in Denmark, has developed a shrimp counting machine similar to those used for fish. The shrimp flow through a tube where they are detected and counted by a high-speed camera. As this method requires shrimp to be actively pumped out of the system through the counter, it is particularly suitable for measurements during stocking, transfer or harvesting. Currently, shrimp from approximately 1.5 cm–10 cm in length can be detected. Sincere Aquaculture claims 90% accuracy in counting. Ideally, one would aim to further increase accuracy and be able to count both smaller and larger shrimps.

The results and statistical tests presented in Section 3 show that detection-based approaches perform significantly better than DM-based methods. One possible explanation for the lower performance with DM-based models is that the density estimation approach is sensitive to

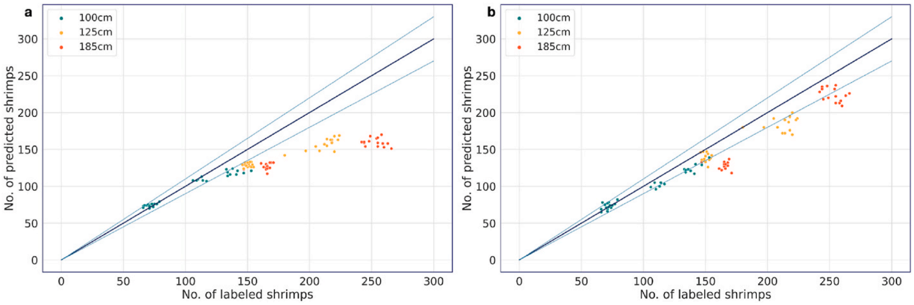

Fig. 5. The relation between ground truth and prediction count for each of the 90 images of the independent test set using a) Model 1 (Faster R–CNN) and b) Model 3 (YOLOv5m6).

isolated object clusters (Bai et al., 2019). Moreover, from the detection-based models, using a YOLOv5m6 architecture achieved a slightly lower MAPE compared to Faster R–CNN, however this improvement is not significant. All models except Model 5, a bare U^2-Net, proved to be satisfactory for the task of counting shrimps, as their MAPE falls well below the estimated 20 % error achieved with manual counting. Additionally, the performance of the best model, Model 4, almost achieves the 5 % threshold set for deployment in an industrial farm which would enable automatic monitoring of shrimp parameters. It was observed that the accuracy of the predictions depends on the density of shrimps within particular tanks. The model is more accurate in the range of shrimps counts up to 120. YOLOv5m6 tends to slightly underestimate shrimp counts in images taken from a 120 cm distance, but copes well with the 80 cm distance category for both conditions: white and blue light. Individual NMS test-time thresholds per each category and individual confidence thresholds slightly improve the performance of both YOLOv5m6 and Faster R–CNN, however this improvement is not significant.

4.1. Generalization of YOLO detection model to other environments

This study is the first of its kind developing and accessing DL models in shrimp real aquaculture farm environments. The two detection-based architectures achieved comparable results on the original test set, however, their value to the industry needs to be assessed by additionally testing it on different data. For this reason, experiments using an independent test set were performed. Here, YOLOv5m6 with global thresholds (Model 3) proved to be more transferable to different environments compared to Faster R–CNN. Fig. 5 shows that YOLOv5m6 is clearly superior. It achieved satisfactory performance (MAPE<20 %), even for images with higher density and at a new camera height (185 cm, Fig. 6) and low resolution, which shows that the YOLOv5m6 model generalizes better for this use case. Most likely, YOLOv5m6 outperforms Faster R–CNN at this task due to the heavier augmentation techniques applied during training, and in particular the mix-up which artificially creates training samples with higher object density and is especially useful when multiple overlapping objects are present on an image (see Fig. S8). Therefore, the model learns during training to correctly detect objects with large overlap. This could also explain why the improvement when applying per category thresholds in Table 1 is smaller for YOLOv5m6 compared to Faster R–CNN, indicating that it generalizes better and doesn't require the NMS and confidence thresholds to be tuned to a specific dataset to achieve good results. However, to further prove the generalization power of YOLOv5m6, it should be additionally tested on a larger dataset containing samples from different farms and even with different taxa.

In a first attempt, the model was tested on a different taxon, specifically rainbow trout in a RAS system at the Center for Aquaculture Research (ZAF), AWI. Impressively it yielded an APE of 11.76 %, despite never being trained with such animals (Fig. 7a). This performance, which is comparable to the average error on the independent test set, is promising and suggests that the proposed model has indeed good generalization potential which could further be explored and effectively used for counting fish in RAS or systems allowing clear vision.

4.2. Future studies to improve shrimp counting

The present study used 1379 images for training and validation, a relatively small dataset, with an average shrimp number of 73 per image. Increased error rates in images of high shrimp density (independent test set with more than 200 shrimps per image) were observed, as DL models normally do not perform well on out-of-distribution samples. Hence, it is likely that more training data in this regime would improve the model performance in tanks with a higher density. In general, concentrating on the sources of error and trying to eliminate them by gathering more problematic scenarios in the training data, would be a recommended approach for future improvement of the model. Additionally, preliminary data obtained with a new camera system (Keyence CV-X490F with a 64 MP color camera, CA-HF6400C, and an 18 mm objective, CA-LHT18) indicate that a higher image resolution allows for shrimp detection even at extremely low contrast of the animals from the background (Fig. 7b), which deserves further investigation.

Clear water shrimp farming systems play a minor role in shrimp farming. The majority of shrimps are farmed in turbid water that exceed 1 m in depth. The present method for shrimp detection obviously does not perform as well in RAS with turbid water (e.g., Fig. 7c), and even worse visibility is expected in ponds which produce the bulk of shrimps worldwide. Companies are now deploying sonar technology to effectively visualize shrimps in turbid ponds. For example, Minnowtech (2024) uses sonar technology claiming a 95% accuracy in determining biomass. The device, called the BRS1, emits sonar that is reflected by the shrimp and provides biomass information. The exact number of individuals and average weight are ignored and only the total biomass is considered. However, this alone can significantly improve feed management, with the additional advantage that this method can be used to determine biomass in turbid water. Hence, it would be of great interest to extend the current models for sonar images acquired in shrimp ponds.

Finally, as discussed in Section 1, the number of shrimps per tank as well as their respective lengths are important measures that need to be monitored in domestic shrimp farming, for example when feeding needs to be adjusted if the growth is lower than expected. So far, only the problem of counting the number of shrimps in a tank was tackled. However, this method could be easily extended and use the bounding

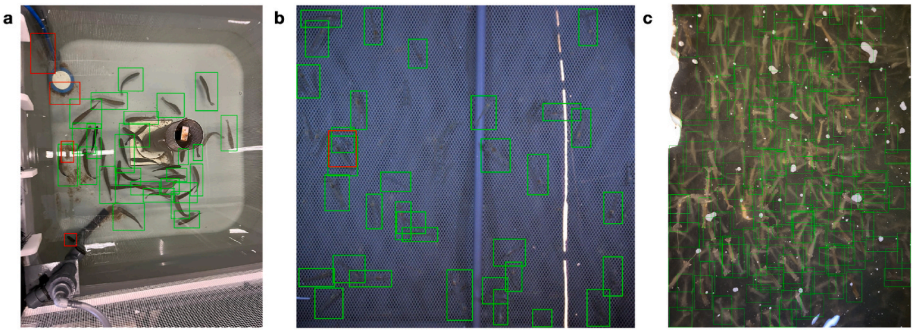

Fig. 7. The performance of Model 3 on out-of-distribution data. Green boxes represent true positive predictions and red boxes represent false positive predictions. a) Image of trout acquired with a smartphone at ZAF, AWI (predicted value is 30, ground truth is 34, APE is 11.76 %). b) Image of shrimps acquired with a Keyence camera system at Oceanloop Kiel GmbH & Co. KG farm (predicted value is 34, ground truth is 42, APE is 19.05 %). c) Image of shrimps acquired with a smartphone camera at the CrustaNova shrimp farm, Germany) - manual counting in this image was not possible. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

boxes to approximate the biomass per tank.

5. Conclusions

The present work demonstrates that high performance can be achieved with DL based counting of shrimps in commercial RAS based farming systems across the entire production range and at various circumstances known to be challenging for object detection algorithms (dim light, overlapping and/or small objects, various acquisition devices, image resolutions and camera distance to object). From present results, a camera lens distance from the water surface between 50 and 100 cm is considered optimal to ensure a sufficiently large image section, to be able to use fast camera lenses and to avoid the splash water area. Under normal lighting conditions in the range of 5–35 lux, this setup is sufficient to achieve adequate image quality with fast lenses. It is also possible to use a flash, which the shrimps cannot perceive due to the short duration (personal observations) and are therefore not stressed by it. The additional effort required to install an additional light source while avoiding disturbing reflections can be compensated for by improved image quality and the associated additional analyzable image information, such as health status.

The proposed method clearly overcomes the 20 % error of manual counting and achieves a promising performance close to the 5 % error threshold set for deployment of the method in an industrial setting. The chosen model performs best at a count below 200 shrimps per image. It is believed that improvements could be obtained by collecting a larger dataset, especially including images of high shrimp density. Remarkably, the model's performance on images with different conditions than those used during model training, only slightly decreases and stays within the 20 % acceptable error range. It is additionally observed that the model performs satisfactorily on images that are completely out of distribution (e.g. imaging of different taxa). This strongly suggests that real time animal counting in commercial RAS based farming environments can be achieved with the proposed models. Currently, RAS production accounts for 1 %–4 % of total aquaculture production worldwide (variation depending on author). However, according to a 2015 report from the Lux Research Water Intelligence service, it is expected that RAS production share will increase to 45 % by 2030. Even if this appears over optimistic, it is likely that RAS share will substantially increase in the next decade and hence computer vision-based counting models such as the present, suitable in clearwater RAS, will become increasingly relevant for the entire market.

Funding sources

The project was funded by the Federal Ministry of Food and Agriculture (BMEL) on the basis of a decision by the German Bundestag. The project was sponsored by the Federal Office for Agriculture and Food (BLE) as part of the programme to promote innovation. Funding code 281C213A19.

CRediT authorship contribution statement

Christina Bukas: Writing – original draft, Writing – review & editing. **Frauke Albrecht:** Conceptualization, Formal analysis, Methodology, Writing – original draft, Writing – review & editing. **Muhammad Saeed Ur- Rehman:** Conceptualization, Software, Writing – original draft, Writing – review & editing. **Daniel Popek:** Software, Writing – original draft, Writing – review & editing. **Mikołaj Patalan:** Data curation, Formal analysis, Software, Visualization, Writing – original draft, Writing – review & editing. **Jarosław Pawłowski:** Data curation, Software, Validation, Writing – original draft, Writing – review & editing. **Bert Wecker:** Conceptualization, Funding acquisition, Investigation, Resources, Writing – original draft, Writing – review & editing. **Kilian Landsch:** Data curation, Investigation, Methodology. **Tomasz Golan:** Data curation, Formal analysis, Software, Visualization, Writing – original draft. **Tomasz Kowalczyk:** Data curation, Formal analysis, Software, Validation, Writing – original draft, Writing – review & editing. **Marie Piraud:** Methodology, Writing – original draft, Writing – review & editing. **Stephan S.W. Ende:** Conceptualization, Funding acquisition, Resources, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was generated within the project 'Digitization of land-based shrimp farming in Germany using AI-based analysis of imaging and acoustic systems' funded by the Federal Office for Agriculture and Food (BLE) (FKZ 281C213A19). The authors would like to thank the professional expert for proofreading this article. This work is supported by the Helmholtz Association Initiative and Networking Fund under the Helmholtz AI platform grant. We highly acknowledge the work of Dr. Jaime Orellana (Erwin Sander Elektroapparatebau GmbH) in project acquisition and his valuable comments during project progress meetings.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jclepro.2024.143024.

References

Armalivia, S., Zainuddin, Z., Achmad, A., Wicaksono, M.A., 2021. Automatic counting shrimp larvae based you only look once (YOLO). AIMS 2021 - International Conference on Artificial Intelligence and Mechatronics Systems. https://doi.org/10.1109/AIMS52415.2021.9466058.

Arteta, C., Lempitsky, V., Noble, J.A., Zisserman, A., 2014. Interactive object counting. Lect. Notes Comput. Sci. https://doi.org/10.1007/978-3-319-10578-9_33 *(Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8691 LNCS (PART 3).

Awalludin, E.A., Mat Yaziz, M.Y., Abdul Rahman, N.R., Yussof, W.N.J.H.W., Hitam, M.S., T Arsad, T.N., 2019. Combination of canny edge detection and blob processing techniques for shrimp larvae counting. In: Proceedings of the 2019 IEEE International Conference on Signal and Image Processing Applications, ICSIPA 2019. https://doi.org/10.1109/ICSIPA45851.2019.8977746.

Bai, H., Wen, S., Chan, S.H.G., 2019. Crowd counting on images with scale variation and isolated clusters. Proceedings - 2019 International Conference on Computer Vision Workshop. https://doi.org/10.1109/ICCVW.2019.00009. *ICCVW 2019*.

Boksuwan, S., Panaudomsup, S., Cheypoca, T., 2018. A prototype system to count nursery pacific white shrimp using image processing. In: International Conference on Control, Automation and Systems, 2018-October.

Everingham, M., van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A., 2010. The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. 88 (2) https://doi.org/10.1007/s11263-009-0275-4.

Fligner, M.A., Killeen, T.J., 1976. Distribution-free two-sample tests for scale. J. Am. Stat. Assoc. 71 (353) https://doi.org/10.1080/01621459.1976.10481517.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/CVPR.2014.81.

Graczyk, K.M., Pawłowski, J., Majchrowska, S., Golan, T., 2022. Self-normalized density map (SNDM) for counting microbiological objects. Sci. Rep. 12 (1) https://doi.org/10.1038/s41598-022-14879-3.

Hashisho, Y., Dolereit, T., Segelken-Voigt, A., Bochert, R., Vahl, M., 2021. AI-assisted automated pipeline for length estimation, visual assessment of the digestive tract and counting of shrimp in aquaculture production. In: VISIGRAPP 2021 - Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, 4. https://doi.org/10.5220/0010342007100716.

Jiang, N., Yu, F., 2020. Multi-column network for cell counting. OSA Continuum 3 (7). https://doi.org/10.1364/osac.396603.

Jocher, G., Stoken, A., Borovec, J., NanoCode, Chaurasia, A., TaoXie, Changyu, L., Abhiram, V., Laughing, tkianai, yxNONG, Hogan, A., lorenzomammana, AlexWang, Hajek, J., Diaconu, L., Marc, Kwon, Y., oleg, et al., 2021. ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 Models, AWS, Supervise.Ly and YouTube Integrations, v5.0. https://doi.org/10.5281/zenodo.4679653. Zenodo.

Kaewchote, J., Janyong, S., Limprasert, W., 2018. Image recognition method using Local Binary Pattern and the Random forest classifier to count post larvae shrimp. Agric. Nat. Resour. 52 (4) https://doi.org/10.1016/j.anres.2018.10.007.

Khantuwan, W., Khiripet, N., 2012. Live shrimp larvae counting method using co-occurrence color histogram. 2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2012. https://doi.org/10.1109/ECTICon.2012.6254280.

Lempitsky, V., Zisserman, A., 2010. Learning to count objects in images. Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. *NIPS 2010*.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C. L., 2014. Microsoft COCO: Common Objects in Context. In: Lecture Notes in Computer Science. https://doi.org/10.1007/978-3-319-10602-1_48 (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8693 LNCS (PART 5).

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. 60 (2) https://doi.org/10.1023/B:VISI.0000029664.99615.94.

Majchrowska, S., Pawłowski, J., Guła, G., Bonus, T., Hanas, A., Loch, A., et al., 2021. AGAR a Microbial Colony Dataset for Deep Learning Detection arXiv preprint arXiv: 2108.01234.

Majchrowska, S., Pawłowski, J., Czerep, N., Górecki, A., Kuciński, J., Golan, T., 2022. Deep neural networks approach to microbial colony detection—a comparative analysis. Lect. Notes Netw. Syst. 440 https://doi.org/10.1007/978-3-031-11432-8_9. LNNS.

Nguyen, K.T., Nguyen, C.N., Wang, C.Y., Wang, J.C., 2020. Two-phase instance segmentation for whiteleg shrimp larvae counting. In: Digest of Technical Papers - IEEE International Conference on Consumer Electronics, 2020-January. https://doi.org/10.1109/ICCE46568.2020.9043075.

Pawłowski, J., Majchrowska, S., Golan, T., 2022. Generation of microbial colonies dataset with deep learning style transfer. Sci. Rep. 12 (1) https://doi.org/10.1038/s41598-022-09264-z.

Qin, X., Zhang, Z., Huang, C., Dehghan, M., Zaiane, O.R., Jagersand, M., 2020. U2-Net: going deeper with nested U-structure for salient object detection. Pattern Recogn. 106 https://doi.org/10.1016/j.patcog.2020.107404.

Ren, S., He, K., Girshick, R., Sun, J., 2016. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. 39 (6), 1137–1149.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: convolutional networks for biomedical image segmentation. Lect. Notes Comput. Sci. 9351 https://doi.org/10.1007/978-3-319-24574-4_28 (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).

Sekachev, B., Manovich, N., Zhiltsov, M., Zhavoronkov, A., Kalinin, D., Hoff, B., et al., 2020. Opencv/Cvat: v1.1.0. https://doi.org/10.5281/zenodo.4009388. Zenodo, Version v1.1.0.

Tkachenko, M., Malyuk, M., Shevchenko, N., Holmanyuk, A., Liubimov, N., 2020. Label Studio v0. Github 8.0, Version 0.8.0. https://github.com/heartexlabs/label-studio.

Wilcoxon, F., 1945. Individual comparisons by ranking methods. Biometrics Bull. 1 (6) https://doi.org/10.2307/3001968.

Wuertz, S., Bierbach, D., Bögner, M., 2023. Welfare of decapod Crustaceans with special emphasis on stress physiology. Aquacult. Res. 2023 https://doi.org/10.1155/2023/1307684.

Xie, W., Noble, J.A., Zisserman, A., 2018. Comput. Methods Biomech. Biomed. Eng. Imaging Vis. 6 (3) https://doi.org/10.1080/21681163.2016.1149104.

Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K., 2017. Aggregated residual transformations for deep neural networks. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. https://doi.org/10.1109/CVPR.2017.634, 2017-January.

Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D., 2018. MixUp: beyond empirical risk minimization. In: 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings.

Zhang, C., Li, H., Wang, X., Yang, X., 2015. Cross-scene crowd counting via deep convolutional neural networks. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 07-12-June-2015. https://doi.org/10.1109/CVPR.2015.7298684.

Web References

Xpertsea, 2024. Retrieved from. https://www.xpertsea.com.

Sincereaqua, 2024. Retrieved from. https://www.sincereaqua.com.

Minnowtech, 2024. Retrieved from. https://minnowtech.com.