Design and Implementation of Multi-Line Juvenile Fish Counting System using YOLO Detection Model with Kalman Filter-based Tracking

Raditya Bagus Wardhana‡, Infall Syafalni‡§, and Trio Adiono‡§
‡School of Electrical Engineering and Informatics, Bandung Institute of Technology, Indonesia
§University Center of Excellence on Microelectronics, Bandung Institute of Technology, Indonesia
e-mail: 13220036@std.stei.itb.ac.id

***Abstract*—In aquaculture industries, fish hatcheries are essential in the production cycle of aquaculture. They are responsible to provide the industry with juvenile fish required for cultivation. In order to achieve high counting accuracy that allows for continuous counting of juvenile fish, a system that captures video of fish flowing through a channel is selected. During the development of the counting system, we limit the fish type to catfish with size range of 5-12 cm and tilapia with size range of 7-12 cm. The design of the counting system consists of three subsystems: the detection module, the tracking module, and the counting module. The detection module used YOLOv6-N as it performs the best in our use case, 0.9 recall and 0.885 average precision and performs the fastest compared to another YOLO models in our hardware. The tracking module used Improved SORT algorithm, which is a kalman filter-based multi-object tracking algorithm. The counting module used two separate algorithm for catfish and tilapia. When tested on real time counting with different number of fish and three time iterations, the average accuracy of both catfish tilapia, still lacks around 1% from reaching the desired accuracy, only reaching average accuracy of 94.4%.**

***Index Terms*—juvenile fish, counting, object detection, object tracking**

I. Introduction

In aquaculture industries, fish hatcheries are essential in the production cycle of aquaculture. They are responsible to provide the industry with juvenile fish required for cultivation. Large-scale hatchery industries can produce tens of thousands juvenile fish in a single harvest [1]. Precise counting of juvenile fish is very important, as it influences decision-making in aquaculture, such as fish feed management and even market transactions like the selling of juvenile fish.

Currently, there are two mainstream methods in juvenile fish counting. The first is to use tools such as sieves, cups, or bowls to count the fish per group or per individual. This method can be accurate, but they are inefficient for handling large numbers of fish as they are very time consuming. Another method used by aquaculture farmer is to measure juvenile fish by weight (kilograms) or volume (liters). Although this approach is faster, it is inaccurate [2].

The trade-off between accuracy and efficiency highlights the need for a system that can automatically count juvenile fish with high levels of both accuracy and efficiency. In this paper, we designed and implemented a counting system which can count juvenile fish accurately. Beside that, we also take into consideration the fact that a hatchery may cultivate more than one type of fish species and size. The development of the system is limited to two species of fish, catfish with sizes of 5-7 cm, 7-9 cm, and 9-12 cm, and tilapia with sizes of 7-9 cm and 9-12 cm.

This paper presents only the design and implementation of the counting system. For the implementation of multi-processing and multi-threading on the embedded computer, please refer to the paper by Raihan, et al [11]. For the design and implementation of the counting channel and power supply of the device, please refer to the paper by Fathin, et al [10].

This paper is divided into another four sections: Section 2 provides a short review of the related literature and works. Section 3 describes the methods and design of the system. Section 4 describes the implementation and the results. Section 5 is the conclusion of the works.

II. Literature Review

Several researches have been conducted on automatic fish fry or juvenile fish counting. A research proposed the use of infrared-based sensor to perform the counting. Tunnels with sensors attached to them are used to flow the fish through it. As the fish flows through the tunnel and passed through the sensor, the fish is counted. This method gives inaccuracy when multiple fish overlap or are close to each other [2]. Another drawback is that this method requires the fish movement to be restricted which may damage the fish [3].

In recent years, the improvement of image processing and computer vision technology have led to an increase in the use of vision-based methods. Some of that utilize thresholding and blob detection or edge detection to segment the fish and then to count it [4] [5]. These methods are static which means a camera will capture an image and then the processing is done. However, static counting still faces the problem of fish overlapping and can count only a fixed number of fish within a container, where the size of the container, the size of the fish, and the number of fish can affect the degree of accuracy.

Another research uses video-based data of the fish flowing through a channel to perform the counting. This method can achieve high count of fish as it allows for continuous counting. This method utilizes deep learning-based object detection and tracking algorithm [6]. The researchers proposed the use of lightweight, single-stage detection model like YOLO for real-time implementation, along with an improved version of Simple Online Realtime Tracking (SORT) [7] that utilizes a Kalman Filter for state estimations of the fish and prediction of fish location in the next moment to count tilapia with size 7-8 cm. This research paper is going to be the main inspiration for the design and implementation of our system. However, a hatchery may also cultivate more than one species of fish and may sale more than one category of fish size, we further apply this method of counting to two fish species, catfish with size categories of 5-7 cm, 7-9 cm, and 9-12 cm and tilapia with size categories of 7-9 cm and 9-12 cm.

A. *Object Detection*

To achieve a good result in dynamic counting method, high-precision and high-recall detection algorithm is required. Deep learning-based object detection can provide this capability and can be categorized into two-stage and one-stage detection algorithms. R-CNN is an example of two-stage detection algorithm. It consists of region proposal, feature extraction by convolutional network, bounding box regression, and classification. This can provide good accuracy, but the computing cost is too high for real-time implementation. An improvement of R-CNN in term of computing speed is called Faster RCNN. However, the computational cost of faster R-CNN is still too high for real time implementation [3].

An alternative to two-stage detection is a single-stage detection algorithm. This can achieve faster detection, even real-time. An example of single-stage detection that is commonly used in object detection task is called YOLO (You Only Look Once) [8]. In YOLO, rather than using sliding windows for object localisation like RCNN, YOLO uses grid system which will divide an image into some grids where each grid is responsible for prediciting the objects. This method of YOLO greatly increases the speed, but the accuracy of the model is lower compared to RCNN or Faster RCNN. However, in recent years, many researches have been conducted in improving YOLO, like YOLOv4, YOLOv5, YOLOv6, YOLOv7, etc. This improvement greatly increases the accuracy of the YOLO model to be comparable to R-CNN or Fastern R-CNN while maintaining real-time detection capability [9]. In the case of juvenile fish counting, YOLOv5-Nano, the most lightweight version of YOLOv5, has been used previously to detect 7-8 cm tilapia which achieves 97.9% recall and 85.4% average precision [6].

B. *Multi-Object Tracking*

In dynamic counting method, object detection alone is not enough. It needs to be combined with Multi Object Tracking (MOT) algorithm. MOT algorithm has been proven to be effective in such scenarios. These methods use a combination of a detection model and a data association algorithm to track the movement of objects. The object detection model detects the bounding boxes of objects and the data association algorithm assigns unique identities, called ID, to each object. By matching objects with the same identity over time, counting of each object can be performed. This method of detection plus tracking has been used in many applications, such as human counting, vehicle counting, etc. For the use of juvenile fish counting, this method has also been proved to be performing well, achieving counting accuracy of 96% [6].

III. Proposed Design

The specifications that we try to achieve for our system is the ability to count juvenile fish with the accuracy of $96 \pm 1\%$. Beside that, we also take into consideration the fact that a hatchery may cultivate more than one type of fish species and size. To achieve this, the system must also be capable to count more than one species and more than one size category for each fish type. During development, We limit the scope of development to two fish type, catfish and tilapia, with size categories of 5-7 cm, 7-9 cm, and 9-12 cm.

In order to achieve high counting accuracy that allows for continuous counting of juvenile fish, a system that captures video of fish flowing through a channel is selected. This method utilizes deep learning-based object detection and multi-object tracking algorithm. Figure 1 shows the top-level block diagram of the designed system. The counting system takes the captured frame of the camera as the input and outputs the number of fish counted. It consists of three subsystems: the detection module, the tracking module, and the counting module.

A. *Detection Module*

The detection module is responsible for identifying the presence of fish within the captured frames. In designing and developing our system, we used a variation of YOLO model as it has been proven to be capable of being used in juvenile fish counting process by having a great detection performance and computing speed [6] as this counting system is going to be deployed on resource-constrained embedded computing system. Selected models are variations of the YOLO model, which include YOLOv5, YOLOv8, YOLOv7, YOLOv6, and YOLOX, given in Table I. These models are going to be trained using catfish and tilapia dataset. Then, the best performing models, in term of detection ability, like recall and average precision, and in term of computing speed is going to be used as the final detection model. Details are given in Section IV. This module's input is frame captured by camera. The output of this module is the bounding box of the fishes. For the implementation, we didn't modify the architecture of the YOLO model and instead focused on data acquisition to perform the training of the model.

B. *Tracking Module*

The tracking module takes the bounding box output of detection module as the input. The output of this module is the ID assigned to each bounding box. This would ensure that each detected fish is different from each other. Ideally, each fish would maintain the same ID for the entire time it flows in the counting chamber until it exits through the outlet. Considering

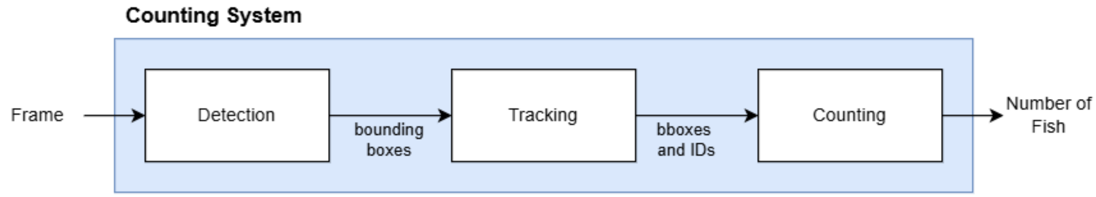

Fig. 1. Counting system

TABLE I
YOLO Model Variants

No	Model Name
1	YOLOv5-n
2	YOLOv8-n
3	YOLOv6-n
4	YOLOv7-tiny
5	YOLOX-n
6	YOLOX-tiny

that detection and tracking are performed within the same processing unit, the tracking algorithm must be kept as simple as possible, but still accurate. For this, we selected an algorithm called Improved SORT proposed by Zhang, et al. [6], which has been used in fish tracking and achieved the best counting result compared to another lightweight tracking algorithm. This algorithm is an improvement from an algorithm called SORT (Simple Online And Realtime Tracking) [7]. SORT uses Kalman Filter for predicting the linear movement of the fish in the next frame. Then, in the next frame, Hungarian Algorithm is used to matches the detection module's bounding box with Kalman Filter's predicted bounding box. For the Kalman Filter, the state vectors of the bounding box is represented in Equation 1 similar to SORT [7], where u and v are position of the center point of the bounding box, s is the scale of the bounding box, r is the aspect ratio of the bounding box, $\dot{u}$ is the velocity of the bounding box in the x-direction, $\dot{v}$ is the velocity of the bounding box in the y-direction, and $\dot{s}$ is the rate of change of the bounding box scale.

$$\mathbf{x} = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T \tag{1}$$

In Improved SORT, there are three matching stages. The first stage uses euclidean distance to match between detected bounding box and the predicted bounding box.

$$D_{ij} = 1 - \frac{(x_i - x_j)^2 + (y_i - y_j)^2}{L^2} \tag{2}$$

The second and third stage use combination of euclidean distance and IOU (Intersection over Union), called DIOU, to perform the matching.

$$DIOU = \frac{IOU + D_{ij}}{2} = \frac{1 - IOU + \frac{(x_i - x_j)^2 + (y_i - y_j)^2}{L^2}}{2} \tag{3}$$

where,

$$IoU = \frac{|D \cap T|}{|D \cup T|} \tag{4}$$

This three-staged matching algorithm of Improved SORT is capable of handling occlusion of fish while keeping the speed of the tracking algorithm fast [6]. The block diagram of the tracking module (Improved SORT) as proposed by Zhang, et al. [6] is shown in Figure 2

C. Counting Module

The counting module is responsible for counting each fish that flows through a channel. The input to this module is the bounding box and the ID. The output is the number of fish counted. Counting can be done using two methods: the first method involves using the highest ID value of detected and tracked fish. The second method uses line counting. In the second method, each fish that crosses a fixed line (a boundary) within a frame is counted. Although the tracking algorithm used has been improved using the Improved SORT algorithm, it remains highly susceptible to ID switching which can inflate the number of fish counted [6]. Therefore, we chose line counting as the preferred method for counting juvenile fish as it minimizes the effect of ID switching during counting. When a juvenile fish intersects with the designated line counter, number of fish counted is incremented by 1.

Furthermore, we explored two different algorithms for line counting. The first one is when bounding box intersects with line, it is counted. Another is when the connected current centroid and past centroid intersects with line counter, it is counted. The flow diagram of the algorithms used in this module are given in Figure 4. These two algorithms are further expanded into five algorithms with adjustments made. Figure 5 shows the difference between each algorithm. Then, experiments are performed to determine which one is the best. Experiments are conducted using one video for each fish type and size. In the video, around 30 fish for each type and size are continuously poured into the device five times to simulate continuous counting of the system. Result of experiments are shown in Table VI. Figure 3 shows the accuracy and precision of each iteration for each fish type with different size. From the experiments, two best performing line counting algorithms are chosen, line counting 1 + position adjustment for catfish with average accuracy of 96.86% and line counting 2 + 3 lines for tilapia with average accuracy of 92%. The result of counting for tilapia is low because Tilapia of size 9-12 cm is easily stuck on the channel as shown in Figure 6. This is a result of misdesign on the counting channel [10]. In the case

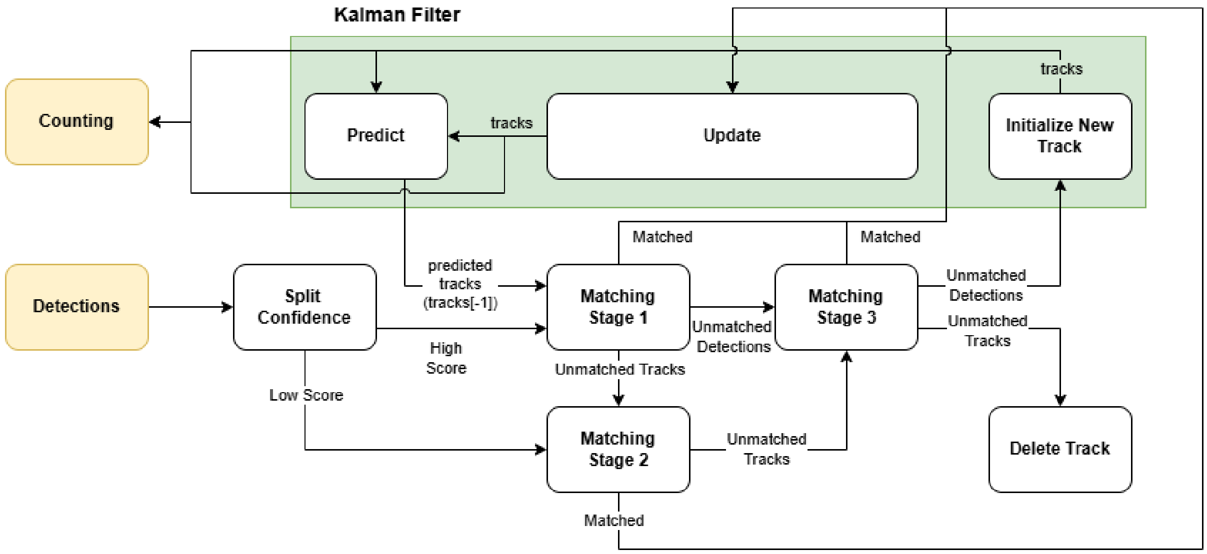

Fig. 2. Block Diagram of Improved SORT

when the fish not stuck, the accuracy of the counting result is still 95% or higher.

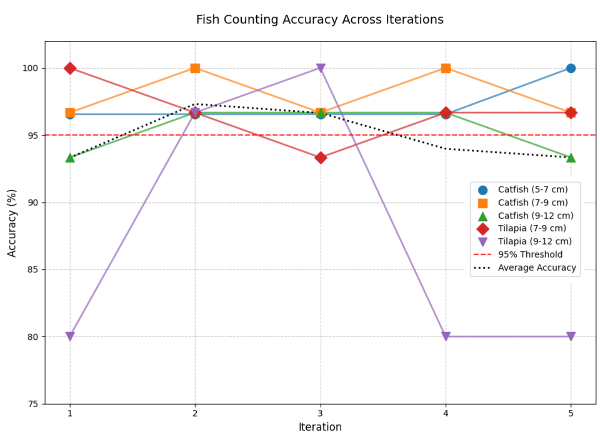

Fig. 3. Result of Fish Count Across Iterations Using Selected Line Counting Algorithms

IV. Implementation & Results

A. Data Preparation

For detection model training, we prepared video data of fish that flow through the channel. The data prepared are for both catfish and tilapia. Video data is acquired from a camera attached to our designed device, this is to ensure that the detection model will work perfectly on our device. Video data is then converted into frames. Details are given in Table II.

For training of the model, we acquired 711 frames of both catfish and tilapia. To further increase the training data, data augmentation is performed. Data augmentation techniques performed are image flip, rotation, brightness, and blur. This increased the training data to 2133 images. For validation during training, 500 frames are used. For testing of the detection model, 3 videos of each fish types and sizes are used. This gave 815 frames for testing the model. In terms of annotation counts, details are shown in Table III

For testing of the tracking and counting module, one video for each fish type and size is used. In the video, around 30 fish for each type and size are continuously poured into the device five times.

TABLE II
Detection Model Dataset Preparation by Frame Counts

No	Dataset	Catfish	Tilapia	Total (Frames)
1	Train	408	303	711
2	Augmentation Train	852	570	1422
3	Validation	296	204	500
4	Test	420	395	815
	Total			3448

TABLE III
Detection Model Dataset Preparation by Annotation Counts

No	Dataset	Catfish	Tilapia	Total Annotations
1	Train + Augmentation	3090	5069	8159
3	Validation	454	911	1365
4	Test	2388	1776	4164
	Total			13688

B. Detection Model Training

Models that have been previously proven to be capable of being used in the use case of juvenile fish counting are selected [6]. These models are variations of YOLO model as shown in Table I. These models are trained for 200 epochs with 640x640 input image size and training configurations used are similar to the model's original configurations.

The performance evaluation of each model on test dataset are shown in Table IV. Each model performs nearly the

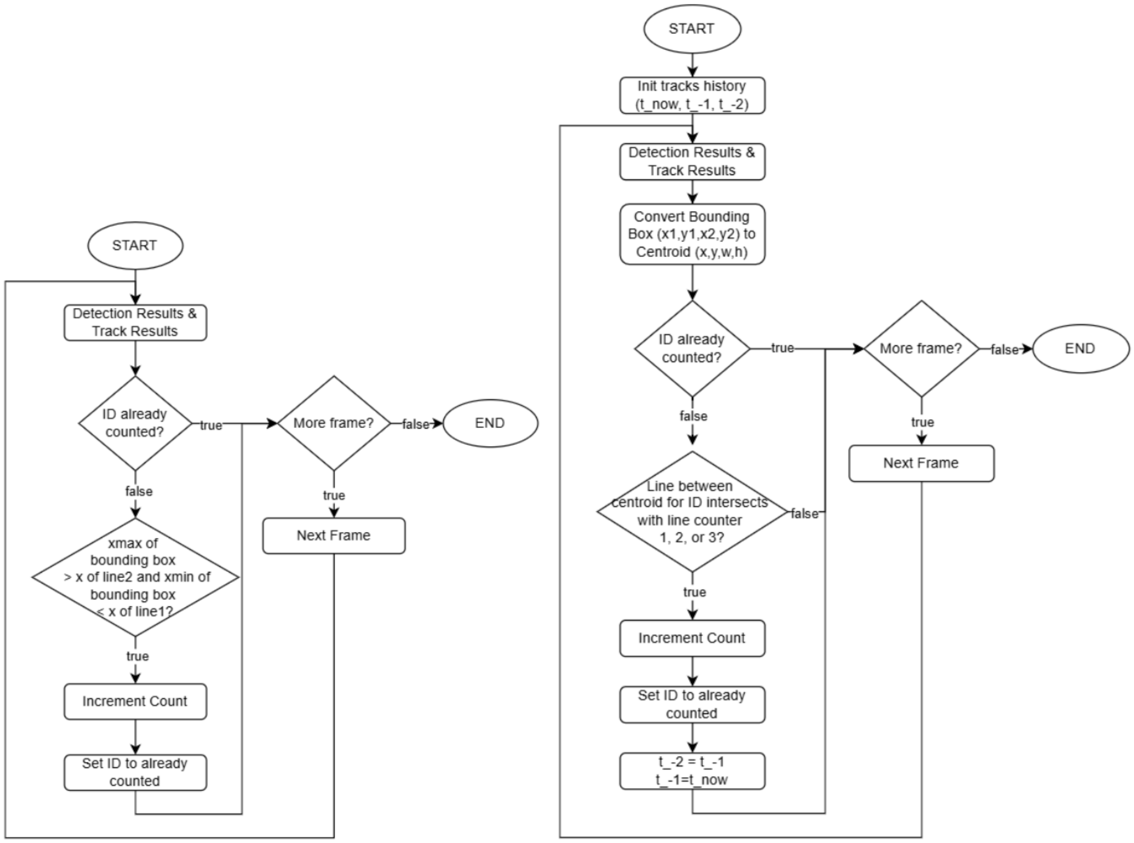

Fig. 4. Flowchart of the two selected line counting algorithm. Bounding box intersects with line counter algorithm (left). Connected current centroid and past centroid intersects with line counter algorithm (right)

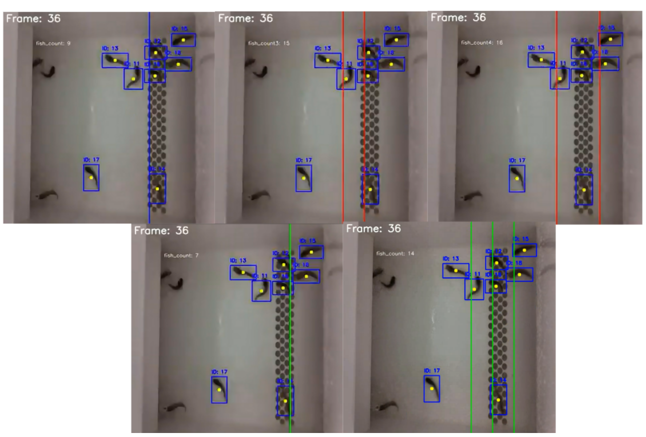

Fig. 5. Variations of Line Counting Algorithm. In order from top-left to bottom-right according to Table VI

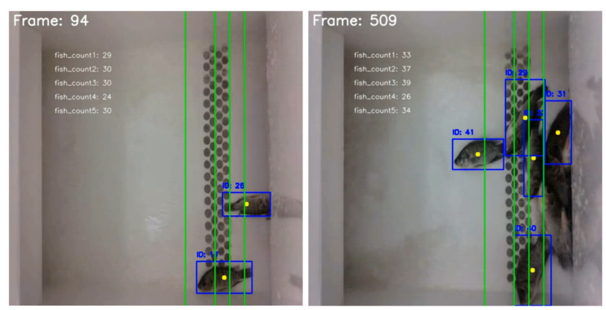

Fig. 6. Fish Stuck on the Channel

same as each other. For selection of the the model, we decided to evaluate all model performences on our counting device selected computer, which is an Orange Pi 5 Pro with RK3588s chip that contains 3 cores NPU. This board and this chip is selected because of it capability to infer various YOLO model up to 60 FPS. This performance will allow the implementation of the proposed design. Further discussion on computer selection and the implementation of the models given in Table I can be seen in the paper by Raihan, et al [11]. From there, we selected YOLOv6-Nano because it has the fastest inference speed on 2 cores RK3588s NPU setup. The example of the detection results of YOLOv6-Nano are shown in Figure 7.

C. Tracking Module

For the implementation of Improved SORT [6], the code from SORT [7] is used as the base. From there, the code is modified with diagram from Figure 2 as the basis for modification. The implementation of Improved SORT has

TABLE IV
PERFORMANCE COMPARISON OF YOLO MODELS

Model	Input Size	Recall	mAP (0.5)	Params (M)	FLOPs (G)
YOLOv5-n	640x640	0.86	0.846	1.9	4.5
YOLOv8-n	640x640	0.87	0.861	3.2	8.7
YOLOv7-tiny	640x640	0.89	0.880	6.22	13.74
YOLOv6-n	640x640	**0.90**	0.885	4.7	11.4
YOLOX-n	640x640	0.69	0.650	0.9	2.55
YOLOX-tiny	640x640	**0.90**	**0.896**	5.03	15.23

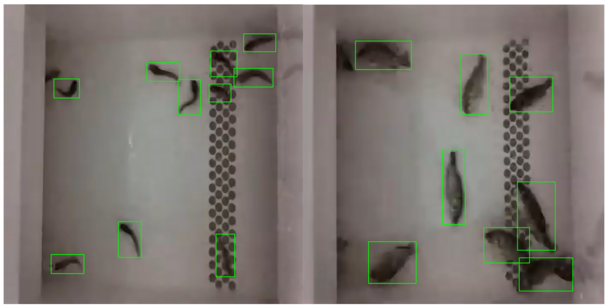

Fig. 7. Detection Results, Catfish (Left) and Tilapia (Right)

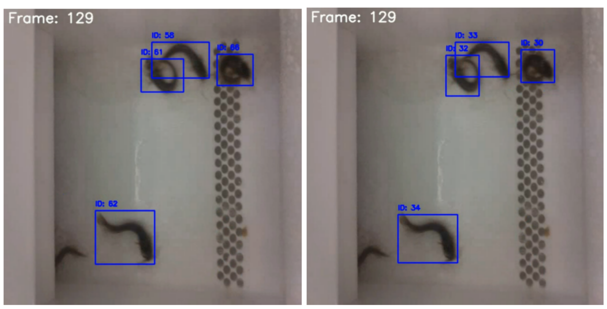

Fig. 8. Result of SORT vs Improved SORT

outputted a unique ID for each bounding box and has managed to reduce the ID switching from SORT which can affect the counting process. Figure 9 shows the result of the tracking module on 3 continuous frames. It can be seen that each fish maintain its unique ID and even though an occlusion occur, as long as the detection module is capable of detecting the fish, the tracking module will also work. Figure 8 shows the difference between SORT and Improved SORT in the same frame.

D. Counting Module

For the implementation of the counting module, the proposed line counting algorithm for catfish and tilapia is both used. For catfish, line counting 1 + position adjustment algorithm is used. For tilapia, line counting 2 + 3 lines algorithm is used. As shown in Table VI, the counting accuracy for catfish has passed the required specification, while counting accuracy for tilapia still lacks around 3-4% to reach the desired 96$\pm$1% accuracy. This is a result of a problem where tilapia of all sizes are easily stuck on the channel's hole for pump water recirculation because of a mistake during the design of the channel [10]. Because of the fish getting stuck, the kalman filter of the tracking module is unable to predict the motion of fish with good result anymore as it loses its assumed motion linearity, resulting in a change of ID, where it will increase the fish count result. For each ID changed, the count result will keep on increasing. In some cases, when only a few fish get stuck, the counting accuracy can still reach the desired 96%. Comparison is shown in Figure 6.

This system is further verified by testing it on real-time video data using camera attached to the device. Tests are done with different number of juvenile fish, 10, 50, 100, and 500. For each test batch, the test is conducted three times to ensure the precision and the robustness of the counting system. The result is shown in Figure 10 and the average accuracy is compiled in Table V. From the result presented, with the smallest sample size (count=10), the system shows some variability in accuracy. However, as the count size increases from 50 to 500, the accuracy stabilizes and shows less fluctuation. The system achieves its highest average accuracy at Count=500 with 96.1%. This is because one or two miscount in small sample size, i.e. count=10, will result in the accuracy being much worse than larger sample size.

Fish size appears to play a crucial role in counting accuracy. Smaller fish (5-7 cm) resulted in lower accuracy, averaging 91.1%. This can be caused by the physical characteristics of catfish in this size range, being very thin and small, which presents additional challenges for the detection model. Beside that, catfish of this range is easily occluded and when that happens, only one fish can be detected and counted. This low accuracy is also contributed by the previously mentioned misdesign in the counting channel, which also affects the accuracy of tilapia.

Overall, the system achieves an average accuracy of 94.4%. The system consistently maintains high accuracy above 90% in all experiment scenarios. The highest sample value, count=500, resulted in an average accuracy that is consistently mantained above the threshold of 96$\pm$1% for all three iterations.

E. Future Improvements

Further investigation and improvement to the device could be done in the future to improve the accuracy much more. This includes the following things: Improving the counting channel so that no fish will get stuck anymore. This can be done by redesigning the water recirculation system. The hole or inlet of the water recirculation can be placed much further into the

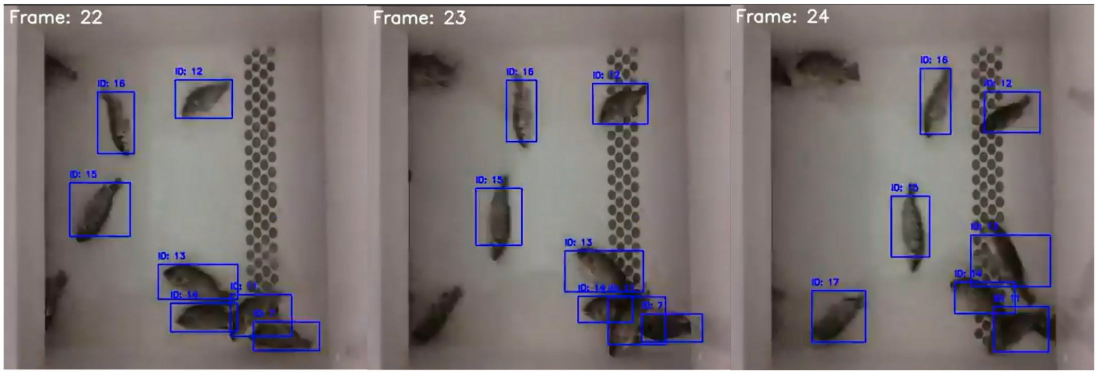

Fig. 9. Result of Tracking Module on 3 continuous frames

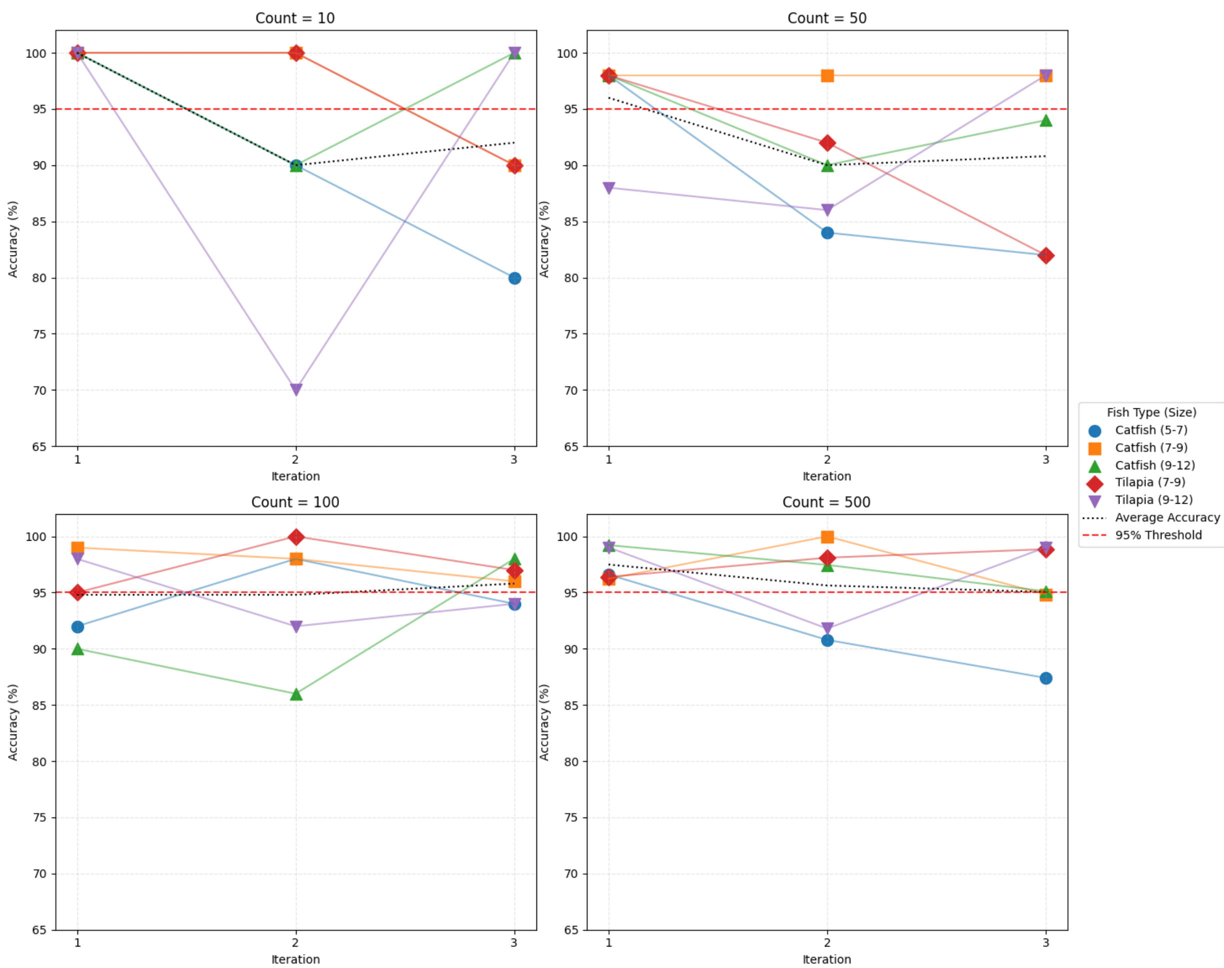

Fig. 10. Result of Real Time Counting

outlet of the device. This will stop the fish from getting stuck in the counting channel which could be captured by the camera which could decrease the accuracy of the counting system as stated previously. Another test and verification is needed to verify this proposed design improvement.

V. Conclusion

In aquaculture hatchery industries, there is a need for an automatic counting system that can count juvenile fish with high accuracy and is capable to be used on a variety of juvenile fish. In this paper, we developed a juvenile fish counting system that that can count two fish species, catfish with size range of 5-12 cm and tilapia with size range of 7-12 cm. We

TABLE V
AVERAGE ACCURACY RESULT OF REAL TIME COUNTING

Fish Types	Fish Sizes	Count Result (Percent of Accuracy)				Average Accuracy	Average Accuracy (Fish Type)
		10	50	100	500		
Catfish	5-7 cm	90.0%	88.0%	94.7%	91.6%	91.1%	**94.5%**
	7-9 cm	96.7%	98.0%	97.7%	97.0%	97.3%	
	9-12 cm	96.7%	94.0%	92.4%	97.3%	95.1%	
Tilapia	7-9 cm	96.7%	90.7%	97.3%	97.8%	95.6%	**94.3%**
	9-12 cm	90.0%	90.7%	94.7%	96.6%	93.0%	
Average Accuracy		94.0%	92.3%	95.4%	96.1%	**94.4%**	

TABLE VI
LINE COUNTING EXPERIMENT RESULTS

Fish Types	Fish Sizes	Real Count	Method	Counting Iteration										Average Accuracy
				1		2		3		4		5		
				Count Result	Accuracy	Count Result	Accuracy	Count Result	Accuracy	Count Result	Accuracy	Count Result	Accuracy	
Catfish	5-7 cm	29	Line Counting 1	21	72.41%	20	68.97%	17	58.62%	23	79.31%	11	37.93%	63.45%
			Line Counting 1 + 2 Lines	28	96.55%	29	100.00%	27	93.10%	27	93.10%	28	96.55%	95.86%
			Line Counting 1 + 2 Lines + Position Adjustment	30	96.55%	30	96.55%	28	96.55%	28	96.55%	29	100.00%	97.24%
			Line Counting 2	20	68.97%	28	96.55%	24	82.76%	24	82.76%	25	86.21%	83.45%
			Line Counting 2 + 3 lines	27	93.10%	29	100.00%	27	93.10%	26	89.66%	29	100.00%	95.17%
	7-9 cm	30	Line Counting 1	25	83.33%	21	70.00%	20	66.67%	23	76.67%	22	73.33%	74.00%
			Line Counting 1 + 2 Lines	29	96.67%	29	96.67%	29	96.67%	27	90.00%	30	100.00%	96.00%
			Line Counting 1 + 2 Lines + Position Adjustment	29	96.67%	30	100.00%	29	96.67%	30	100.00%	31	96.67%	98.00%
			Line Counting 2	25	83.33%	29	96.67%	27	90.00%	26	86.67%	29	96.67%	90.67%
			Line Counting 2 + 3 lines	29	96.67%	29	96.67%	29	96.67%	26	86.67%	30	100.00%	95.33%
	9-12 cm	30	Line Counting 1	24	80.00%	28	93.33%	27	90.00%	28	93.33%	27	90.00%	89.33%
			Line Counting 1 + 2 Lines	27	90.00%	29	96.67%	31	96.67%	29	96.67%	31	96.67%	95.33%
			Line Counting 1 + 2 Lines + Position Adjustment	28	93.33%	29	96.67%	31	96.67%	29	96.67%	32	93.33%	95.33%
			Line Counting 2	24	80.00%	28	93.33%	30	100.00%	28	93.33%	25	83.33%	90.00%
			Line Counting 2 + 3 lines	25	83.33%	29	96.67%	30	100.00%	29	96.67%	31	96.67%	94.67%
Tilapia	7-9 cm	30	Line Counting 1	29	96.67%	26	86.67%	32	93.33%	28	93.33%	29	96.67%	93.33%
			Line Counting 1 + 2 Lines	30	100.00%	30	100.00%	32	93.33%	31	96.67%	31	96.67%	97.33%
			Line Counting 1 + 2 Lines + Position Adjustment	30	100.00%	33	90.00%	33	90.00%	33	90.00%	31	96.67%	93.33%
			Line Counting 2	24	80.00%	28	93.33%	23	76.67%	24	80.00%	27	90.00%	84.00%
			Line Counting 2 + 3 lines	30	100.00%	29	96.67%	32	93.33%	31	96.67%	31	96.67%	96.67%
	9-12 cm	30	Line Counting 1	35	83.33%	30	100.00%	30	100.00%	35	83.33%	38	73.33%	88.00%
			Line Counting 1 + 2 Lines	38	73.33%	32	93.33%	30	100.00%	36	80.00%	40	66.67%	82.67%
			Line Counting 1 + 2 Lines + Position Adjustment	41	63.33%	35	83.33%	31	96.67%	39	70.00%	40	66.67%	76.00%
			Line Counting 2	26	86.67%	30	100.00%	28	93.33%	28	93.33%	30	100.00%	94.67%
			Line Counting 2 + 3 lines	36	80.00%	31	96.67%	30	100.00%	36	80.00%	36	80.00%	87.33%

aim to achive counting accuracy of 96±1% for all fish species and sizes. From the result shown in Table VI, the system managed to reach 96.63% average accuracy for catfish and 94.44% average accuracy for tilapia. However, when tested on real time counting with different number of fish, 10, 50, 100, and 500, and three test iterations, the average accuracy of both catfish tilapia, still lacks around 1% from reaching the desired accuracy, only reaching average accuracy of 94.4%. This is a result of a misdesign of the counting channel which causes the tilapia fish to be easily stucked which can interfere with the tracking algorithm which resulted in the counting being wrong. The system consistently maintains high accuracy above 90% in all experiment scenarios. When tested on the highest sample value, count=500, the average accuracy can be consistently mantained above 95%.

REFERENCES

[1] A. Oktoriyana, U. Sumarwan, and H. Hartoyo, "Perumusan Strategi Pemasaran Fry Counter Dengan Pendekatan Model AIDA," Jurnal Ilmu Keluarga Dan Konsumen IPB, vol. 7, no. 2, pp. 103–112, 2014. doi: 10.24156/jikk.2014.7.2.103.

[2] W. Purbowaskito and R. Handoyo, "Perancangan Alat Penghitung Benih Ikan Berbasis Sensor Optik," Jurnal Rekayasa Mesin, vol. 8, Universitas Atma Jaya, pp. 141–148, 2017.

[3] D. Li, Z. Miao, F. Peng, L. Wang, Y. Hao, Z. Wang, T. Chen, H. Li, and Y. Zheng, "Automatic counting methods in aquaculture: A

review," J. World Aquacult. Soc., vol. 52, pp. 269–283, 2021. doi: 10.1111/jwas.12745.

[4] R. T. Labuguen, E. Volante, A. Causo, R. Bayot, G. Peren, R. M. Macaraig, N. Libatique, and G. L. Tangonan, "Automated fish fry counting and schooling behavior analysis using computer vision," in 2012 IEEE 8th International Colloquium on Signal Processing and its Applications (CSPA), Barcelona, 2012, pp. 255–260. doi: 10.1109/cspa.2012.6194750.

[5] D. P. Andini, I. C. J. Riadi, and F. A. Ariiq, "Fish Fry Counter Based on Digital Image Processing Method," in E3S Web of Conferences, vol. 424, 2024. doi: 10.1051/e3sconf/202447907027.

[6] H. Zhang, W. Li, Y. Qi, H. Liu, and Z. Li, "Dynamic fry counting based on multi-object tracking and one-stage detection," Computers and Electronics in Agriculture, vol. 209, 2023, Art. no. 107871. doi: 10.1016/j.compag.2023.107871.

[7] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, 2016, pp. 3464–3468. Available: https://github.com/abewley/sort.

[8] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," CoRR, vol. abs/1506.02640, 2015. Available: http://arxiv.org/abs/1506.02640.

[9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, Apr. 2020. Available: http://arxiv.org/abs/2004.10934.

[10] M. F. M. Rochaka, I. Syafalni, and T. Adiono, "Design and Implementation of Water Flow and Power Supply Systems for Catfish and Tilapia Seed Counters," unpublished, School of Electrical Engineering and Informatics, Bandung Institute of Technology.

[11] R. F. Yanuarsyah, I. Syafalni, and T. Adiono, "Implementation of Realtime Inference Using Cost-Efficient CPU and NPU Based Computing Device for Automatic Juvenile Fish Counter," unpublished, School of Electrical Engineering and Informatics, Bandung Institute of Technology.