A multitask model for realtime fish detection and segmentation based on YOLOv5

QinLi Liu [Equal first author, 1], Xinyao Gong [Equal first author, 1], Jiao Li [1], Hongjie Wang [1], Ran Liu [1], Dan Liu [1], Ruoran Zhou [1], Tianyu Xie [1], Ruijie Fu [1], Xuliang Duan [Corresp. 1]

[1] College of Information Engineering, Sichuan Agricultural University, Ya'an, Sichuan, China

Corresponding Author: Xuliang Duan
Email address: duanxuliang@sicau.edu.cn

The accuracy of fish farming and real-time monitoring are essential to the development of "intelligent" fish farming. Although the existing instance segmentation networks (such as Maskrcnn) can detect and segment the fish, most of them are not effective in real-time monitoring. In order to improve the accuracy of fish image segmentation and promote the accurate and intelligent development of fish farming industry, this paper uses YOLOv5 as the backbone network and object detection branch, combined with semantic segmentation head for real-time fish detection and segmentation. The experiments show that the object detection precision can reach 95.4% and the semantic segmentation accuracy can reach 98.5% with the algorithm structure proposed in this paper, based on the golden crucian carp dataset, and 116.6 FPS can be achieved on RTX3060. On the publicly available dataset PASCAL VOC 2007, the object detection precision is 73.8%, the semantic segmentation accuracy is 84.3%, and the speed is up to 120 FPS on RTX3060.

1 # A multitask model for realtime fish detection and seg
2 # mentation based on YOLOv5

4 Qinli Liu[1,*], Xinyao Gong[1,*], Jiao Li[1], Hongjie Wang[1], Ran Liu[1], Dan Liu[1], Ruoran Zhou[1],
5 Tianyu Xie[1], Ruijie Fu[1], Xuliang Duan[1]

6 [1] College of Information Engineering, Sichuan Agricultural University, Ya'an, Sichuan, China;
7 * These authors contributed equally to this work.

9 Corresponding Author:
10 Xuliang Duan[1]
11 Xinkang Road, Ya'an, Sichuan, 625014, China
12 Email address: duanxuliang@sicau.edu.cn

Abstract

15 The accuracy of fish farming and real-time monitoring are essential to the development of
16 "intelligent" fish farming. Although the existing instance segmentation networks (such as
17 Maskrcnn) can detect and segment the fish, most of them are not effective in real-time
18 monitoring. In order to improve the accuracy of fish image segmentation and promote the
19 accurate and intelligent development of fish farming industry, this paper uses YOLOv5 as the
20 backbone network and object detection branch, combined with semantic segmentation head for
21 real-time fish detection and segmentation. The experiments show that the object detection
22 precision can reach 95.4% and the semantic segmentation accuracy can reach 98.5% with the
23 algorithm structure proposed in this paper, based on the golden crucian carp dataset, and 116.6
24 FPS can be achieved on RTX3060. On the publicly available dataset PASCAL VOC 2007, the
25 object detection precision is 73.8%, the semantic segmentation accuracy is 84.3%, and the speed
26 is up to 120 FPS on RTX3060.
27 **Keywords:** realtime monitoring, golden crucian carp dataset, multi-task,object detection,
28 semantic segmentation, YOLOv5

Introduction

31 Humans have known how to catch fish since ancient times, and now the demand for fish is
32 increasing year by year. The nutritional value of fish is extremely high and fish is easy to absorb,
33 which is popular with people. In the process of the aquaculture, managers need to gain
34 information about the lives of fish, shrimp, shellfish and other aquatic organisms, specifically,
35 species, behavior identification and biomass estimation(total weight of fish and shrimp in
36 specific waters). Among them, the biomass estimation is the total weight of fish and shrimps in a
37 specific water. The information on the biomass of fish, shrimp and other aquatic organisms at

38 various growing stages is crucial, because managers need to optimize feeding requirements and
39 make effective decisions based on this information.
40   The conventional estimation of fish biomass mainly adopts manual fishing and weighing,
41 which not only consumes human resources and decreases efficiency, but also may have adverse
42 effects on the growth of fish. Since there is a relationship between the weight of organisms, and
43 their body length and image area, the weight can be estimated indirectly through a deep learning-
44 based image segmentation method to predict daily feed intake of aquatic organisms. It can also
45 monitor the growth rate of aquatic organisms, control breeding density, determine optimal
46 harvesting time, and ensure optimal utilization of facility investment. Due to the reliance on
47 visual pattern matching in recognition process, it is convenient and least invasive compared to
48 traditional manual monitoring. Non-invasive techniques offer significant advantages in terms of
49 cost, safety and convenience. Therefore, using computer vision methods for intelligent detection
50 of fisheries is an inevitable trend in the development of fish farming industry in modern society
51 [1].
52   There are mainly four tasks in computer vision regarding image recognition: Classification,
53 Location, Detection, and Segmentation. Classification focuses on "What broad category of object
54 is in this photograph" [2], and is concerned with the overall content of the image. It is mainly
55 divided into two types of tasks: dichotomous and multiclassification tasks, and is widely used in
56 agriculture, medicine, soil, etc. [3~5]. The main task of localization is to find where the object is,
57 generally in the form of a bounding box to circle the location of the object [6]. Unlike
58 classification, detection is to find out which objects are in the photograph and obtain their
59 category information and location information, which is a combination of classification and
60 location [2]. Segmentation mainly focuses on pixels, solving the problem of "What pixels belong
61 to the object in the image" [2]. Segmentation includes semantic segmentation and instance
62 segmentation. The semantic segmentation is to determine whether pixels go well with the object
63 without distinguishing different instances of the same category, while the instance segmentation
64 needs to determine different instances of the same category on the basis of semantic
65 segmentation, which is a combination of object detection and semantic segmentation.
66   Object detection is a collection of classification and regression, mainly aiming to find out
67 all the targets of interest in an image and determine their categories and locations. Early object
68 detection algorithms were based on handcrafted features. The representative ones are Viola Jones
69 Detectors [7], HOG Detectors [8] and Deformable Part based Model (DPM) [9]. These
70 algorithms have laid an important foundation for later object detection algorithms. The
71 development of object detection algorithms was limited due to the saturation of handcrafted
72 features. However, with the advent of Convolutional Neural Networks, object detection begins
73 developing unprecedentedly, and object detection algorithms based on deep learning has become
74 mainstream. Deep learning-based object detection algorithms are mainly divided into two
75 categories, One-Stage and Two-Stage. Common algorithms for One-Stage include OverFeat [6],
76 YOLOv2 [10], YOLOv3 [11], YOLOv4 [12], SSD [13], and RetinaNet [14], while for Two-
77 Stage there are R-CNN [15], SPP-Net [16], Fast R-CNN [17], and Faster R-CNN [18], etc.

78   Currently, these algorithms are widely used in many fields such as security, military,
79   transportation, and daily life, etc. [19~22]. Among the algorithms for object detection, One-
80   Stage's object detection algorithm enables a completely single training of shared features and
81   rapid acceleration, while guaranteeing accuracy. For example, YOLOv1 enables real-time object
82   detection with the basic network running at 45 frames per second and processes streaming videos
83   in real time with a latency of less than 25 ms [23].
84   As a quintessential computer vision problem, semantic segmentation involves taking some
85   raw data (e.g., planar images) as input and converting them into masks with highlighted regions
86   of interest. Semantic segmentation is mainly divided into standard semantic segmentation and
87   instance-aware semantic segmentation, commonly using FCN [24], OCRNet [25], Deeplabv3+
88   [26], and UPerNet [27]. These algorithms are mainly used in self-driving cars, geological
89   detection, facial segmentation, precision agriculture, clothing classification, and other fields [28–
90   32].
91   Instance segmentation is currently a comparatively mature technique in the field of real-
92   time scene understanding and image information processing [33]. It is the localization of
93   instances in an image using a object detection algorithm, and then on the basis of semantic
94   segmentation, the target objects in different localization frames are further segmented into
95   specific objects of classified categories. In contrast to the bounding box of object detection,
96   instance segmentation can be accurate to the edges of objects, while compared with semantic
97   segmentation, it needs to annotate different individuals of the same object on the graph. The
98   existing instance segmentation networks, such as Mask R-CNN [34] and Faster R-CNN [18],
99   have the high rate of accuracy, however, they fail to process details such as segmented edges
100   meticulously and have a low speed.
101   All stages of fish growth and development are susceptible to various external factors,
102   possibly resulting in poor growth or even death of fish, which brings serious losses to fish
103   farmers. The focus of current intelligent fishery research is the way to analyze and understand
104   the state of fish growth process in a timely and effective manner Therefore, our research will
105   focus on the real-time nature of the model, so that we can later deploy it on cell phones or
106   monitoring devices to be applied to fish farms for real-time monitoring of fish growth process. In
107   recent years, researches on fish image detection segmentation based on deep learning have
108   attracted great attention. Chen G et al [35] proposed an automatic fish classification system
109   based on deep learning, Akgul T et al [36] proposed fish detection in turbid underwater, Xiu L et
110   al [37] proposed a deep and lightweight network for detecting fish, Wang J H et al [38] proposed
111   a detection of abnormal behaviors of underwater fish using artificial intelligence techniques,
112   Funkuralshdaifat N F et al [39] proposed a deep learning framework for segmentation of fish
113   with underwater videos, KM Knausgrd et al [40] proposed a method for detecting and
114   classificating temperate fish, Piyadarsan Parida and Nilamani Bhoi (2018) [41] proposed a
115   hybrid transition region color image segmentation method based on dual transition region
116   extraction for fish image segmentation applications. Xiangxiao Lei (2018) et al [42] proposed an
117   image segmentation method based on equivalent 3D entropy and artificial fish population

118   optimization algorithm, which is more efficient than the traditional 3D entropy method and the
119   equivalent 3D entropy method. Kazim Raza and Song Hong (2020) [43] proposed a quick and
120   accurate method for fish detection based on the improved YOLO-v3 model and migration
121   learning. Its average mean Accuracy Precision (mAP) of 87.56%, which increased from 87.17%
122   to 91.30%, compared with the experimental analysis of the YOLOv3 model and the improved
123   model. Kewei Cai et al [44] (2020) proposed an improved YOLOv3 fish detection model based
124   on MobileNetv1 as the backbone, which combined YOLOv3 with MobileNetv1 for real
125   detection of farmed fish.C. S. Arvind et al [45] used Mask R-CNN model along with GOTURN
126   tracking algorithm for fish detection and tracking, showing that image multi-region parallel
127   processing and tracking are accurate with F1 score of 0.91 at 16 frames per second inland.
128   Nevertheless, most methods mentioned above suffer from slow segmentation and low speed,and
129   also cannot be better monitored in real time. Thus, we use a multi-task network YOLOv5 as the
130   backbone network and object detection branch, combined with semantic segmentation head
131   optimized by GhostC3 module, improving the speed and accuracy of the model in general.
132   In short, our main contributions are:
133   1.   Construct a golden crucian carp segmentation dataset: Because of little public dataset of fish
134   segmentation, we have built a new golden crucian carp dataset,containing 640 semantic
135   segmentation images of 10 golden crucian carps.
136   2.   Give comparison to the existing mainstream instance segmentation networks, object
137   detection networks and semantic segmentation networks.
138   3.   Detect and segment golden crucian carp based on YOLOv5. We use a multitasking model to
139   solve the problem of monitoring fish and acquiring fish segmentation images in real time,
140   which perform real-time detection and segmentation while ensuring high accuracy. At the
141   same time, we use Ghost module to further optimize the model and reduce the number of
142   parameters of the model. Compared with most of the networks framed by instance
143   partitioning, the FPS of our model is higher and can reach the standard of real-time
144   monitoring.

145

Materials & Methods

147   Multi-task learning model is learning multiple tasks together, and the efficiency and quality
148   of studying each task can be improved by learning the connections and differences between
149   different tasks. Different from traditional single-task learning model, which seeks to use a
150   specific model [46] to accomplish the task, multi-task learning model is more effective to
151   improve the generalization of models. In addition, computational repetition can be minimized,
152   inference speed can be increased, and memory utilization can be reduced by sharing layers
153   across multiple tasks. [47]
154   In this paper, our dataset is divided into two types of annotations, and our model also
155   includes 2 tasks: object detection and semantic segmentation of golden crucian carp. These two
156   share the backbone based on convolutional neural network, and the shared layer improves the
157   feature extraction ability of golden crucian carp and obtains better performance and

158 generalization. The architecture of our model is shown in Figure 1, and the specific experimental
159 procedure and methodology will be pointed out below.
160
161 **2.1. Acquisition of Materials**
162   Golden crucian carp is a kind of fish with strong physique, strong resistance and wide eating
163 habits. It is easy to raise and doesn't need good care. At the same time, the shape of golden
164 crucian carp is similar to crucian carp and grass carp, so the method we proposed can also be
165 well promoted to the cultivation of crucian carp and grass carp in the future. To sum up, we took
166 the golden crucian carp as the experimental object. During the experiment, we collected pictures
167 of fish in a non-intrusive way, without any impact on the fish itself.
168   We purchased 10 easily distinguishable golden crucian carps in the ornamental fish market
169 and used a transparent tank, 80 cm long, 30 cm wide and 60 cm high, as the breeding
170 environment. The size of the purchased tank and the number of fish is reasonable, so as to ensure
171 that the data collected will not be too concentrated and easy to distinguish. The 10 golden
172 crucian carps were raised in a transparent square tank, provided with sufficient oxygen, and fed
173 with common ornamental fish feed. At the same time, in order to simulate a real environment in
174 the water, we added different doses of water conditioner DEBAO and nitrifying bacteria
175 YUECAI to the initial water environment. Finally, it formed a stable and balanced water
176 environment and achieved the optimal effect.
177   We used the DJI pocket2 camera to capture golden crucian carps from different angles and
178 distances. In order to ensure the reliability of the experimental data, the images were taken
179 randomly, taking into account the effects of sunlight as well as indoor lighting and other factors.
180 When collecting the dataset, we placed the camera about 15-20 cm away from the front of the
181 fish tank and adjusted the camera angle to ensure that we could get a complete figure of the
182 inside of the fish tank without too much background outside the fish tank, which not only ensure
183 the diversity of the collected images, but also enhance the adaptability of subsequent models to
184 various environments [1]. In the deep learning model, the quality of the image has a crucial
185 impact on the training results of the model. The higher the quality of the image annotation is, the
186 better the performance of the model may be. Images without golden crucian carp are useless
187 images for data labeling work, and images with golden crucian carp heavily obscured by aquatic
188 plants and images with turbid water will make labeling work difficult. Therefore, for the
189 captured images, we manually screened and removed some images of poor quality (such as
190 images of golden crucian carp not captured, images of golden crucian carp heavily occluded by
191 aquatic plants and images with turbid water).
192   Finally, we obtained a dataset consisting of 1858 images from 10 golden crucian carps.
193 There are 2 categories of datasets, which contain 1858 general frame object detection datasets
194 and 640 segmentation datasets.
195   We raised golden crucian carps and photographed them, and our experiments met the
196 ethical requirements for animal welfare, and we obtained the affidavit of approval of animal
197 ethical and welfare approved by the Sichuan Agricultural University IACUC, numbered

198   20200054. After the data collection was completed, we released the ten golden crucian carps and
199   placed them in an ornamental fish pond on campus, where they could be fed and cared for by
200   relevant professionals.
201   The above dataset was annotated by 10 annotators under the guidance of professionals. The
202   annotation work is divided into two main parts. In the first part, the object detection dataset was
203   produced. LabelImg is a graphical image annotation tool written in Python language, often used
204   for the production of object detection datasets. The annotations are stored as xml files in
205   PASCAL VOC format, and YOLO format is also supported.. Therefore, we use labelImg to
206   frame the target, as shown in Figure 2, and the annotation is done by framing golden crucian
207   carps' bodies with a horizontal box to ensure that each fish is completely framed into the
208   corresponding box.
209   For the second part, the segmented dataset is created. Labelme, an image annotation tool
210   developed by the Computer Science and Artificial Intelligence Laboratory (CSAIL) of MIT, can
211   be used to annotate images in polygons, polylines, points and other forms, which is a common
212   tool for image annotation in segmentation tasks. Therefore, we use labelme to label golden
213   crucian carp and generate json format for the corresponding image. As shown in Figure 3, the
214   body outline of each golden crucian carp in the image is framed with dense points, and the
215   number of golden crucian carp in each image varies. Finally we obtained the annotation results
216   of about 6000 golden crucian carps, and these datasets were used for the subsequent experiments
217   of semantic segmentation and instance segmentation.
218   In the subsequent experiments, we divided both the object detection and segmentation
219   datasets into a training dataset, a validation dataset, and a test dataset, all in the ratio of 8:1:1.
220
221   **2.2. Data enhancement for object detection**
222   YOLOV5 passes each batch of training data through the data loader and enhances the
223   training data at the same time. The data loader performs three types of data enhancement: scaling,
224   color space adjustment and mosaic enhancement. For the training of the object detection
225   YOLOv5 model, we used various tricks such as Fliplrud, MixUp, Mosaic, and HSV_Aug for
226   data enhancement on the golden crucian carp dataset, analyzing and comparing the effect of
227   different combinations of tricks on the golden crucian carp dataset, and selecting the most
228   suitable tricks.
229   **2.2.1. Fliplrud**
230   Fish are moving freely in the water and their positions and poses are diverse. The dataset
231   should be contained more information of fish to improve the recognition ability of the model.
232   Therefore, we set up two methods respectively:random horizontal flip and random vertical flip,
233   and the execution probability is 0.5. As shown in Figure 4, fliplr is flipping the image 180
234   degrees from left to right or right to left, and flipud is flipping 180 degrees from top down or
235   bottom up.
236   **2.2.2. MixUp**

237   MixUp is an algorithm for mixed-class enhancement of images used in computer vision. It
238   can mix images between different classes, specifically by fusing the features and labels of two
239   samples to expand the training dataset.
240   We took the fusion ratio λ to obey the beta (α, β) distribution (where the α value range is
241   generally [0.1, 0.4], and the λ value range is between [0, 1]). As shown in Equation 1, we mix
242   two random samples, and for each input batch image, together with the randomly selected image,
243   it is fused with the randomly selected image, separately for the image itself and the
244   corresponding label according to the fusion ratio λ, to obtain the mixed tensor Mixed_Batch.
245   Among them, the principle of MixUp actually uses linear interpolation.

$$\text{Mixed\_batch} = \lambda * \text{batch}_1 + (1 - \lambda) * \text{batch}_2 \qquad (1)$$

246   **2.2.3. Mosaic**
247   Mosaic randomly selects four images from the dataset at a time. After randomly cropping,
248   randomly flipping, and color gamut changes for each image, the four images are placed in the
249   order of top left, top right, bottom left, and bottom right and superimposed to form a new image.
250   As shown in Figure 5, this is the image obtained after Mosaic operation during the training
251   process. Mosaic not only enriches the background of the dataset images, but also expands the
252   dataset and improves the robustness of the model.
253   **2.2.4. HSV_Aug**
254   RGB is the most common color space that we often come into contact with. The golden
255   crucian carp dataset image we've collected is an RGB image. Images acquired in natural
256   environments are often disturbed by factors such as lighting, occlusion, shadows, etc. For such
257   changes in brightness, the three components in the RGB color space are highly correlated with it.
258   In the HSV color space, the change of light can be directly expressed through the change of
259   brightness, which is closer to people's perception of color and more suitable for image
260   processing. Therefore, we converted the values of the R, G, and B components of the golden
261   crucian carp data into HSV components according to the following Equations (2–4), thereby
262   obtaining an HSV image. In this way, the characteristics of the image can be expressed more
263   intuitively, and the effect can be significantly enhanced.

$$V = \max(R,G,B) / 255 \qquad (2)$$

$$S = (\max(R,G,B) - \min(R,G,B)) / (float)\max(R,G,B) / 255 \qquad (3)$$

$$H = \begin{cases} 0°, & \text{if } \max(R,G,B) == \min(R,G,B); \\ 60 * \dfrac{G - B}{\max(R,G,B) - \min(R,G,B)} + 0, & \text{if } \max(R,G,B) == R \text{ and } G \geq B; \\ 60 * \dfrac{B-R}{\max(R,G,B) - \min(R,G,B)} + 120, & \text{if } \max(R,G,B) == G; \\ 60 * \dfrac{R-G}{\max(R,G,B) - \min(R,G,B)} + 240, & \text{if } \max(R,G,B) == B; \\ 60 * \dfrac{G - B}{\max(R,G,B) - \min(R,G,B)} + 360, & \text{if } \max(R,G,B) == R \text{ and } G < B \end{cases} \qquad (4)$$

264   **2.2.5. FocalLoss**
265   FocalLoss is a kind of dynamic scaling cross-entropy loss based on dichotomous cross-
266   entropy, which focuses on hard-to-distinguish samples by a dynamic scaling factor to reduce the
267   weight of easy-to-distinguish samples in the training process. FocalLoss mainly solves the
268   problem of one-stage object detection cases with extreme imbalance between foreground and
269   background classes during training. The formula is as follows:

$$FL(p_t) = -(1 - p_t)^\gamma \log(p_t), \tag{5}$$

270   where γ is a constant in the range of values [0, 5], and the formula is as follows:

$$P_t = \begin{cases} p & if\ y = 1 \\ 1-p & otherwise, \end{cases} \tag{6}$$

271   where the values of y are 1 and -1, representing the foreground and background, respectively,
272   and the values of p are 0~1, representing the probability that the model prediction belongs to the
273   foreground.
274   By using the above data enhancement techniques, we are able to diversify our dataset,
275   reduce overfitting of the data, and effectively improve the generalization ability of the model.
276   **2.3. Methods of detection and estimation**
277   We used the golden crucian carp dataset to test some of the current mainstream instance
278   segmentation networks, and the experimental results are shown in Table 1.
279   mAP (mean Average Precision) is an important measure of the object detection algorithm,
280   which is calculated by the comprehensive weighted average of the average accuracy rate (AP) of
281   all class detection. We calculate the mAP for IoU at 0.5, and the process requires a definite
282   integral to the Precison-Recall curve to find the area. The mAP@0.5 can be calculated as

$$mAP@0.5 = \frac{1}{n}\sum_{i=0}^{n-1}\int_0^1 p(r_i)dr, \tag{7}$$

283   where n is the number of categories and $p(r_i)$ is a Precison-Recall curve function formed by
284   using the model's Recall as the abscissa and Precision as the ordinate.
285   FPS refers to the number of images that can be processed by the network model in 1s.
286   As is seen from Table 1, although these instance segmentation networks have high precision,
287   FPS is low and real-time monitoring is poor. According to the principle of instance segmentation,
288   instance segmentation can be regarded as a combination of object detection and semantic
289   segmentation. Moreover, the current object detection networks (such as Yolov4s, CenterNet, etc.)
290   are fast and have a good real-time detection effect. Therefore, we try to use the combination of
291   object detection and semantic segmentation instead of instance segmentation to achieve real-time
292   detection and segmentation.
293   In a word, our main methods in this paper are as follows: using the Yolov5 network as the
294   backbone network and object detection head, adding semantic segmentation head in Yolov5,
295   using Ghost module to further optimize the semantic segmentation head, replacing the activation
296   function.
297   **2.3.1. Object detection**

298   In this paper, we primarily use the Yolov5 network, version 5.0, as the backbone network
299   and object detection branch, and the full name of YOLO is you only look once, which means that
300   you only need to browse once to identify the category and location of objects in the figure.
301   The YOLOv5 model was publicly released by Ultralytics on June 9, 2020, and their code
302   and experimental results are open-sourced at https://github.com/ultralytics/yolov5/tree/v5.0. The
303   YOLOv5 model is based on the improved YOLOv3 model, with YOLOv5s, YOLOv5m,
304   YOLOv5l, and YOLOv5x models. In terms of model volume, the number of parameters for
305   YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x are 7.3M, 21.4M, 47.0M, and 87.7M,
306   respectively. The detection speed of YOLOv5s is faster than all the other three models,
307   especially 2 times faster than YOLOv5x. In general, the smaller the model is, the faster the
308   detection speed is, and it's also easier to deploy on embedded devices, so we choose YOLOv5s
309   for our subsequent experiments.
310   YOLOv5 is a One-Stage network, which has improved accuracy and speed compared with
311   other networks before YOLOv5 in the YOLO series, and its network flow is shown in Figure 6.
312   The network structure of YOLOv5 mainly is composed of Backbone, Neck, and Head,
313   where YOLOv5 uses CSPDarknet as Backbone to extract abundant information features from the
314   input image. Neck utilizes PANet structure, and Neck is mainly used to generate feature
315   pyramids. YOLOv5 uses the same Head as YOLOv3, which is a 1*1 convolutional structure
316   with three sets of outputs.
317   Either Aadm or SGD optimizer can be chosen in YOLOv5, and we use SGD optimizer.
318   For the CSP structure in YOLOv5, YOLOv5 uses two kinds of CSP structures, the first one
319   mainly uses CSP1_X in Backbone, where the Bottleneck is using the Resunit structure, and the
320   second one is using CSP2_X in Neck, where the Bottleneck does not use the Resunit structure.
321   YOLOv5 has significant advantages shown as follows:
322   ● Compared with the Darknet framework used in YOLOv4, using the Pytorch framework is
323   very user-friendly and easy to train your own dataset, ,and implement into production
324   ● It's easy to read code, integration of a large number of computer vision techniques, very
325   conducive to learning and learning
326   ● Not only is it easy to configure the environment, but also the model training is rapid, and
327   batch inference can produce real-time results
328   ● It's able to effectively inference directly on individual images, batch images, video and even
329   webcam port inputs
330   ● The Pytorch weights can be easily converted to ONXX format for Android, which can then
331   be converted to OPENCV format, or to IOS format via CoreML for direct deployment to
332   mobile applications.
333   ● The speed of YOLOv5s object recognition is impressively up to 140FPS, and the using
334   experience is very excellent
335   **2.3.2. Semantic segmentation**
336   In this paper, we add a segmentation head to YOLOv5 network for the segmentation task.
337   The segmentation head takes the output of the 16th layer of YOLOv5 as input, and the channel

338   configuration is 512. It is mainly composed of C3, SPP and Up Sampling modules. Our
339   segmentation head network refers to the design idea of Deeplabv3+ [26], PSPNet [48] and other
340   networks. The specific network structure is shown in Figure 7.
341   SPP [16] is the abbreviation of spatial pyramid pooling, which can effectively avoid image
342   distortion caused by image region clipping and scaling operations.
343   The Up Sampling here uses bilinear upsampling, which is one of the interpolation
344   algorithms and is an extension of linear interpolation. It uses four real existing pixel values
345   around the target point in the original image to jointly determine a pixel value in the target map,
346   and the core idea is to perform linear interpolation once in each of the two directions.
347   As shown in Figure 8, we need to ask for the pixel values of point P. We know the
348   coordinates of Q11, Q21, Q12, Q22, and P. The pixel values of Q11, Q21, Q12, and Q22 are also
349   known. So first use the single linear interpolation about X to calculate the pixel values of R1 and
350   R2 respectively, as shown in Equations 8 and 9.

$$f(x,y_1) \approx \frac{x_2 - x}{x_2 - x_1}\, f(Q_{11}) + \frac{x - x_1}{x_2 - x_1}\, f(Q_{21}) \tag{8}$$

$$f(x,y_2) \approx \frac{x_2 - x}{x_2 - x_1}\, f(Q_{12}) + \frac{x - x_1}{x_2 - x_1}\, f(Q_{22}) \tag{9}$$

351   The letters f(Q11), f(Q12), f(Q21), f(Q22), x1, x2, and x in the equation on the right are
352   already known, and the derived f(x,y1) and f(x,y2) are the pixel values of R1 and R2.
353   Then the pixel values of point P are calculated with single linear interpolation about the y
354   direction to obtain Equation 10.

$$f(x,y) \approx \frac{y_2 - y}{y_2 - y_1}\, f(x,y_1) + \frac{y - y_1}{y_2 - y_1}\, f(x,y_2) \tag{10}$$

355   The letters y1, y2, and y are known in the equation on the right. f(x,y1) and f(x,y2) are the
356   values of R1 and R2 pixels found in the previous equation.
357   The C3 module in the segmentation head is a feature extraction module designed by
358   YOLOv5 based on CSPNet (Cross Stage Partial Network) (Wang et al., 2020), in which the
359   stacking of multiple Bottleneck layers causes significant overhead.
360   In addition, affected by the GhostNet [49] function of the lightweight network architecture,
361   we replaced the C3 module with the GhostC3 module. The GhostC3 module structure is shown
362   in Figure 9. The main difference between GhostC3 and C3 is that it replaces the Bottleneck layer
363   in C3 with the GhostBottleneck layer, which makes GhostC3 use fewer parameters, but has
364   higher accuracy than the original. In our test, we found that after replacing C3 module with
365   GhostC3 module, the overall parameters used by the model changed from 7731361B to
366   7577985B, a decrease of 2%.
367   GhostC3 mainly consists of Conv, concat, and GhostBottleneck modules, and the main
368   components of GhostBottleneck module are GhostConv. Among them, the structure of
369   GhostConv is shown in Figure 10, C1 and C2 are the number of input and output channels, half
370   of the output feature map comes from one regular convolution, and the other half is generated by

371 5×5 Depthwise convolution on the result of the first one. Compared with the original
372 convolutional layer, GhostConv is able to achieve the same or even more efficient feature
373 extraction with less complexity. In CNN networks, it is common to have a lot of convolutional
374 computations with redundant intermediate feature maps, while GhostConvolution forces the
375 network to learn useful features from half of the convolutional kernels.
376   The structure of the GhostBottleneck layer is similar to the residual part of the Resnet.
377 Figure 11 shows the structure for a step size of 1, which consists of two Ghost convolutional
378 stacks with 1×1 kernels and residual connections. The GhostBottleneck layer with a step size of
379 1 can be used to increase the number of channels of the input feature map and to expand the
380 processing for later operations.
381   Figure 12 shows the structure of GhostBottleneck with a step size of 2. Depthwise
382 convolution is applied to downsample the feature map between GhostConv. To ensure that the
383 features in the residual branch have the same dimension as the original features, the Depthwise
384 convolution is used for downsampling and the original 1×1 convolution is used to change the
385 number of channels. The GhostBottleneck layer with a step size of 2 enables feature extraction
386 by stacked GhostConv when downsampling the feature map and lifting the channel. At the same
387 time, it mitigates the gradient disappearance problem by residual concatenation.
388   The GhostBottleneck structure approach can not only effectively reduce the model
389 parameters and computation, but also improves the detection efficiency of the model by
390 optimizing the feature map with the Ghost module. Therefore, we replaced the original C3
391 module with the GhostC3 module.
392 **2.3.3. Activation function improvement**
393   The YOLOv5 model provides a variety of activation functions, such as ReLU, Swish, etc.
394 The activation functions can increase the nonlinear factors. The linear model is not sufficiently
395 expressive, and a nonlinear function is introduced as the excitation function so that the deep
396 neural network is more expressive.
397   Therefore, to further improve the accuracy of our model, for the activation function, we
398 replaced the Swish activation function used Conv of the YOLOv5 model with the HardSwish
399 activation function.
400   The Swish activation function (as shown in Figure 13) replaces the ReLU activation
401 function, which can significantly improve the accuracy of the neural network, and the specific
402 definition of the Swish activation function is shown in Equation 11.

$$f(x) = x * sigmoid(x) \qquad\qquad (11)$$

403   Although this non-linearity improves the precision, the sigmoid function is composed of
404 exponentials, which are much more computationally expensive on mobile devices. The sigmoid
405 activation function can be fitted with the segmented linear function HardSigmoid, as shown in
406 Equation 12.

$$Hardsigmoid(x) = \begin{cases} 0, & x \leq -3 \\ 1, & x \geq 3 \\ \dfrac{x}{6} + \dfrac{1}{2}, & \text{otherwise} \end{cases} \tag{12}$$

407   As a consequence, replacing sigmoid with Hardsigmoid can greatly reduce the cost of
408 operations, which led to the birth of HardSwish (As shown in Figure 14), as specified in the
409 Equation 13.

$$HardSwish(x)$$
$$= x * Hardsigmoid(x) = x * \frac{ReLU6(x + 3)}{6} = x * \begin{cases} 1, & x \geq 3 \\ \dfrac{x}{6} + \dfrac{1}{2}, & -3 < x < 3 \\ 0, & x \leq -3 \end{cases} \tag{13}$$

410   The derivative of this function with respect to x is:

$$HardSwish'(x) = \begin{cases} 1, & x \geq 3 \\ \dfrac{x}{3} + \dfrac{1}{2}, & -3 < x < 3 \\ 0, & x \leq -3 \end{cases} \tag{14}$$

411

Results

413   Our experiments are divided into three main steps.
414   First, we validate multiple object detection models, and then select the network with better
415 performance as the baseline to be used as the model in the first step.
416   Second, we add a segmentation head to the baseline model and perform ablation experiment
417 on it to select the most suitable tricks.
418   Third, we further optimize the proposed model and compare the results with those produced
419 by other models to demonstrate the competitive advantage of our approach.
420   Through stepwise experiments, we can ensure that each step of our identification of golden
421 crucian carp is a local optimal solution, thus an effective model is built for the detection and
422 segmentation of golden crucian carp.

3.1. Model selection

3.1.1. Object detection model selection

425   Currently, the mainstream target detection models are CenterNet, YOLOv4, YOLOv5,
426 EfficientDet, and RatinaNet, and we trained these five target detection models on the existing
427 golden crucian carp dataset, and the results of these model tests are listed in Table 2. To ensure
428 the accuracy of comparison between the results of each model above, we uniformly conducted
429 the experiments on RTX 5000 with uniform settings of parameters: image size of 640x640,
430 optimizer of Adamw, initial learning rate set to 0.001, and epoch of 300.
431   In the above experiments, we choose Precison, Recall, F1-score, mAP@0.5,
432 mAP@0.5:0.95 and Inference@batch_size 1 as the performance evaluation indexes for model
433 effectiveness, and we consider these indexes together to determine the selection of our final

434   model. Through Table 2, we find that CenterNet has the highest precision of 95.21% on the
435   golden crucian carp dataset, but its Recall value is lower compared to other networks, only
436   reaching 92.48%, while YOLOv5s Recall reaches 95.38%. Therefore, we reconcile Precision and
437   Recall for the mean. And we summed Precision and Recall to obtain the F1-score, which is
438   exactly the same. But in terms of mAP, YOLOv5s is better. At the same time, from the overall
439   point of view, YOLOv5s reaches the top of each index compared with the rest of the network
440   models, so we finally choose YOLOv5s.
441   In the experimental prediction process, the following four cases occur for the golden crucian
442   carp dataset: (a)correctly identify the golden crucian carp, indicated by TP (b) incorrectly
443   identified other things as golden crucian carp, denoted by FP (c) did not identify golden crucian
444   carp, denoted by FN (d) non- golden crucian carp things were not identified as golden crucian
445   carp, denoted by TN. To obtain the specific values of these three cases, the calculation of IoU is
446   required, which is the degree of overlap between the predicted boxes and the boxes marked in
447   the original figure, as shown in Figure 15, which is calculated as in Equation 15 (where G is the
448   real box and D is the predicted box), and in general, we write down the number of detected boxes
449   with IoU greater than 0.5 as TP, while the data of detected boxes whose values are less than or
450   equal to 0.5 are written down as FP.

$$IoU = \frac{G \cap D}{G \cup D} \tag{15}$$

451   Using TP,FP and FN, we can calculate the above four metrics in the following way.
452   As shown in Equation 16, Precison is the ratio of the number of correct predictions in the
453   forecast results.

$$Precison = \frac{TP}{TP + FP} \tag{16}$$

454   Recall denotes the probability of being detected in all positive samples during the prediction
455   of experimental results, which in this experiment is the probability of being detected in all
456   pictures containing golden crucian carp, and is calculated as in Equation 17.

$$Recall = \frac{TP}{TP + FN} \tag{17}$$

457   The F1-score can well distinguish the strengths and weaknesses of the algorithm and is the
458   summed average of Precision and Recall, as shown in Equation 18.

$$F1\text{-}score = 2 \times \frac{Recall \times Precison}{Recall + Precison} \tag{18}$$

459   mAP@0.5:0.95 represents the average mAP at different IoU thresholds (from 0.5 to 0.95 in
460   steps 0.05).
461   Inference@batch_size 1 indicates the inference time required for a picture.
462   The experimental results show that YOLOv5s and CenterNet show relatively better
463   performance, with F1-scores reaching 0.94 and mAP@0.5 around 0.95. However, the
464   Inference@batch_size 1 of YOLOv5 is smaller than CenterNet ,and takes less inference time to

465 process the same photo, so we chose YOLOv5s as the backbone network for subsequent
466 experiments.

3.1.2. Selection of data enhancement tricks

468 We added a semantic segmentation head in YOLOv5 for subsequent experiments. To
469 further improve the effectiveness of our model and enhance its generalization ability, we used
470 different tricks to process the model and further ablation experiments were conducted to extract
471 the best model training configuration. Table 3 and Table 4 show the effects of using tricks such
472 as HSV_Aug, FocalLoss, Mosaic, MixUp and the evaluated metrics afterwards. This symbol,"√"
473 indicates that the technique was used and "×" indicates that it was not used.

474 Considering the connection and impact in different data enhancement strategies (e.g.,
475 MixUp [50] is implemented on top of Mosaic, thus MixUp is only used as the Mosaic strategy is
476 enabled), we finally selected only 12 combinations as shown in Table 3 for our experiments.

477 The results of the experiments are shown in Table 4, and the results show that some tricks
478 decrease the accuracy of our network, while some tricks increase the accuracy of the network.

479 For the semantic segmentation model, mIoU is its standard metric. The formula of mIoU is
480 shown in Equation 19, which is labeled as Seg mIoU in Table 4. In our experiment, the golden
481 crucian carp dataset is mainly divided into two categories, fish and background, so k is 1.

$$\text{mIoU} = \frac{1}{k+1}\sum_{i=0}^{k}\frac{TP}{FN + FP + TP} \tag{19}$$

482 seg ACC refers to the pixel accuracy, and its corresponding formula is shown in Equation
483 20.

$$\text{seg ACC} = (TP + TN)/(TP + TN + FP + FN) \tag{20}$$

484 FocalLoss is often used to suppress background classes in unbalanced data and target
485 detection[51]. As can be seen in the results of Table 4 for both group 2 and 3, and group 7 and 10,
486 the accuracy of the model after FocalLoss processing decreases with the golden crucian carp
487 dataset and the corresponding settings in this paper. We speculate that there is no significant
488 imbalance in the space allocation between background and object classes in the data, causing
489 FocalLoss not to work.

490 All of our detectors were experimented using pre-trained weights. The experimental results
491 of groups 3 and 4 can clearly see that Mosaic has a certain effect on improving the precision of
492 the model, and the experimental results of groups 6 and 7 can also show that Mosaic can improve
493 the performance of the model, so we decided to use Mosaic. Looking at the experiments in
494 groups 8 and 9, we can see that MixUp has little effect on model precision, recall and other
495 parameters. Group 5 added MixUp to group 4, and the precision of the model decreased, and
496 group 11 added MixUp to group 10, and the performance of the model also decreased, so we
497 decided not to use MixUp. Similarly, the experiments in groups 7 and 8 led us to reject Fliplrud.
498 Therefore, we chose the tricks of group 8, that is, we only used HSV_Aug and Mosaic for
499 subsequent experiments.

500

3.2. Parameter setting

502 In the Yolov5s model with segmentation head, we finally used the 8th set of tricks as in
503 Table 4, and we set the hyperparameters of HSV color model as H (hue): 0.015, S (saturation):
504 0.7, V (brightness): 0.4, and mosaic as 1. We set the image input size as 640*640, epochs We set
505 the input size to 640*640, epochs to 300, SGD optimizer and initial learning rate to 0.0015, and
506 put the custom golden crucian carp dataset with batch size of 2 into the network for training, and
507 our experimental environment for training is ubuntu 20.04 with RTX3060 graphics card.
508
509 **3.3. Experimental results**
510 Our final model uses YOLOv5s version 5.0 as the backbone network, and adds a
511 segmentation head with Ghost module optimization, and replaces the original Swish activation
512 function with the HardSwish activation function, which is relatively less computationally
513 expensive. The final object detection accuracy is 0.954, the recall is 0.930, and the speed is 116.6
514 FPS on RTX3060.
515 The experimental results after improving the semantic segmentation head and replacing the
516 activation function are shown in Table 5.
517 According to Equation 16, we can see that precision refers to the probability of True
518 positive in all samples that are predicted to be correct, mainly focusing on "finding precision";
519 According to Equation 17, we can see that recall refers to the probability that True positive
520 accounts for the sample that is actually True, and focuses mainly on "finding the full". If you just
521 look at recall or precision, you may go to extremes without knowing it. We expect the results to
522 be both accurate and complete, so we choose F1-score, which combines precision and recall, as
523 the evaluation metric. We tend to choose the model with a higher F1-score.
524 After replacing the C3 module in segmentation head with the GhostC3 module, the F1-score
525 and FPS values of the model were improved compared with the results of the initial version of
526 the segmentation head model. After replacing the Swish activation function with the HardSwish
527 activation function on the segmentation head model improved with the Ghost module, the F1-
528 score of the model increase to more than 0.94 and the FPS increase by 0.6, which is relatively
529 obvious. Therefore, we will use the model of the Ghost module that has optimized the
530 segmentation head and the activation function is HardSwish as our final model.
531 As shown in Table 6, our multi-task model shows advantages over other models in both
532 detection and segmentation tasks. In particular, in terms of speed, the FPS of our model is much
533 higher than the other models. To further demonstrate the effectiveness of our model, we conduct
534 experiments on the publicly available dataset PASCAL VOC 2007. We divide the dataset into
535 training set, validation set and test set in the ratio of 8:1:1, and the size of the training images is
536 set to 512*512. Finally, our object detection accuracy is 0.738, semantic segmentation accuracy
537 is 0.843, and FPS is 120. We use our final model to make predictions for the golden crucian carp
538 videos and take four images from it, as shown in Figure 16.
539 From this prediction result, we can see the video prediction effect of our model: high
540 accuracy of object detection and high coincidence of semantic segmentation with the object to be
541 detected.

542

Discussion

544   The current development trend shows that instance segmentation technology has a large
545 potential for application in smart fisheries. However, the problems of instance segmentation,
546 such as low speed and inferior real-time detection, have contributed to the poor results of its
547 application. In this paper, we study a multi-task model based on YOLOv5, add a segmentation
548 head for semantic segmentation based on the original YOLOv5 framework, improve the
549 segmentation head by Ghost module ,and replace the activation function of the model to further
550 improve the overall accuracy of the model. Based on this, our discussions are as follows:

4.1. Contribution to smart fisheries

552   Fishery is an indispensable part of China's national economy. With the continuous
553 improvement of people's living standards, the process of fish farming has received wide social
554 attention. At all the stages of fish growth and development are susceptible to a variety of external
555 factors, possibly resulting in poor fish growth or even death, thus causing serious losses to
556 farmers. Therefore, the major focus of current research on smart fisheries is how to analyze and
557 understand the state of fish growth process in a timely and effective manner. To address this
558 problem, researchers have proposed many effective methods. Among them, real-time monitoring
559 of fish using deep learning techniques is by far the most effective method. In this paper, we add
560 the optimized segmentation head of Ghost module to the YOLOv5s object detection model to
561 achieve the function of real-time detection and segmentation, and experimentally prove that our
562 method greatly improves the detection speed in real-time monitoring, and the object detection
563 accuracy and semantic segmentation accuracy are above 0.95. Therefore, our method can be
564 applied to the smart fishery aspect, which is important for the monitoring activities of fish
565 growth and development process.

566

4.2. Limitations and future work

568   At present, our model mainly utilizes a object detection algorithm with a semantic
569 segmentation algorithm, and the segmentation head technique is used to effectively combine the
570 two algorithms to form a new network model that guarantees the detection of the object while
571 being able to segment it. Our experiments test on the existing golden crucian carp dataset, and
572 the final results have achieved an accuracy of more than 95% and an FPS of 116.6, showing
573 excellent results. However, our experiments still have some limitations. Our dataset is only for
574 the golden crucian carp, and we have not yet experimented the effect of the model on the dataset
575 of common fish such as grass carp and crucian carp. In the future, we will work on the expansion
576 of the original golden crucian carp dataset to diversify the fish species to ensure that the dataset
577 can contain more fish with generalizability, and we will also keep on expanding the quantities of
578 each type of fish to form a large-scale fish dataset. On this foundation, we will conduct further
579 experiments and optimization on the basis of this model to ensure the universality of the model.
580 Our ultimately purpose is to implement the model in practice, and we intend to develop it into a

581   system that can estimate the biomass of fish and continuously optimize and expand its functions
582   to effectively promote the intelligent development of fishery farming.
583

Conclusions

585   In this paper, we first detected and segmented golden crucian carps by the general instance
586   segmentation algorithm. Although the experimental results were comparatively good in terms of
587   accuracy, each FPS was low and it was difficult to reach the standard of real-time monitoring.
588   Subsequently, we added the Ghost module optimized semantic segmentation head to the
589   YOLOv5 model and replaced the activation function, and finally propose our model. It is
590   experimentally confirmed that our model can effectively identify grass carp and perform real-
591   time segmentation, with object detection accuracy up to 95.4% , semantic segmentation accuracy
592   up to 98.5%, and a speed of 116.6 FPS on RTX3060 , which can effectively meet the
593   requirements related to real-time detection.
594   In the future, our research will be further developed in the following aspects.
595   1. In this paper, the survival environment of our fish is a fish tank, but in the actual farm, it may
596   be more complex. For this reason, we will consider further optimization of the experimental
597   environment to ensure a closer fit to the farm's culture environment and increase the robustness
598   of the model.
599   2. Grass carp is a unique fish species and an important freshwater economic fish in China.
600   Considering its mature farming technology, low-cost input, little difficulty in management , high
601   survival rate and high economic value, we choose grass carp as the research target. However, the
602   characteristics of grass carp are not obvious enough compared with golden crucian carp. For this
603   reason, we intend to add some hardware such as one-way pipes to assist us in collecting more
604   detailed and comprehensive data sampling of grass carp to avoid the problem of poor model
605   training results caused by insufficient features.
606   3. We will further extend our features to perform fish pose estimation, behavior recognition,
607   target detection of fish diseases and semantic segmentation of fish diseases based on semantic
608   segmentation, in order to form a completely intelligent fish farming system and improve the
609   foundation for future application in actual farms.
610

Acknowledgements

612   We are grateful to Prof.Duan and other teachers for their support and assistance. It's their
613   help that we can carry out research so smoothly during our college life.

614

Data Availability

616   The following information was supplied regarding data availability :
617   The model is available at https://doi.org/10.5281/zenodo.7413306
618   The golden crucian carp detection dataset is available at https://doi.org/10.5281/zenodo.7413354
619   The golden crucian carp segmentation dataset is available at
620   https://doi.org/10.5281/zenodo.7410176

621

622

623

References

1. Lin, Bin, Kailin Jiang, Zhiqi Xu, Feiyi Li, Jiao Li, Chaoli Mou, Xinyao Gong, and Xuliang Duan. "Feasibility Research on Fish Pose Estimation Based on Rotating Box Object Detection." Fishes 6, no. 4 (2021): 65.

2. Brownlee J. Deep learning for computer vision: image classification, object detection, and face recognition in python[M]. Machine Learning Mastery, 2019.

3. Ashraf T, Khan Y N. Weed density classification in rice crop using computer vision[J]. Computers and Electronics in Agriculture, 2020, 175: 105590.

4. Liaqat, Amna, Muhammad A. Khan, Muhammad Sharif, Mamta Mittal, Tanzila Saba, K. Suresh Manic, and Feras NH Al Attar. "Gastric tract infections detection and classification from wireless capsule endoscopy using computer vision techniques: A review." Current Medical Imaging 16, no. 10 (2020): 1229-1242.

5. Srivastava P, Shukla A, Bansal A. A comprehensive review on soil classification using deep learning and computer vision techniques[J]. Multimedia Tools and Applications, 2021, 80(10): 14887-14914.

6. Sermanet, Pierre, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. "Overfeat: Integrated recognition, localization and detection using convolutional networks." arXiv preprint arXiv:1312.6229 (2013).

7. Viola P, Jones M. Rapid object detection using a boosted cascade of simple features[C]//Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001. Ieee, 2001, 1: I-I.

8. Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]//2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). Ieee, 2005, 1: 886-893.

9. Felzenszwalb P, McAllester D, Ramanan D. A discriminatively trained, multiscale, deformable part model[C]//2008 IEEE conference on computer vision and pattern recognition. Ieee, 2008: 1-8.

10. Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263-7271.

11. Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.

12. Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv:2004.10934, 2020.

13. Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "Ssd: Single shot multibox detector." In European conference on computer vision, pp. 21-37. Springer, Cham, 2016.

660   14. Lin, T. Y., P. Goyal, R. Girshick, K. He, and P. Dollár. "Focal loss for dense object
661   detection. Proceedings of the IEEE International Conference on Computer Vision." (2017):
662   2980-2988.
663   15. Girshick, Ross, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich feature hierarchies
664   for accurate object detection and semantic segmentation." In Proceedings of the IEEE
665   conference on computer vision and pattern recognition, pp. 580-587. 2014.
666   16. He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Spatial pyramid pooling in
667   deep convolutional networks for visual recognition." IEEE transactions on pattern analysis
668   and machine intelligence 37, no. 9 (2015): 1904-1916.
669   17. Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer
670   vision. 2015: 1440-1448.
671   18. Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time
672   object detection with region proposal networks." Advances in neural information processing
673   systems 28 (2015).
674   19. Saikia, Surajit, Eduardo Fidalgo, Enrique Alegre, and Laura Fernández-Robles. "Object
675   detection for crime scene evidence analysis using deep learning." In International
676   Conference on Image Analysis and Processing, pp. 14-24. Springer, Cham, 2017.
677   20. Janakiramaiah, B., G. Kalyani, A. Karuna, L. V. Prasad, and M. Krishna. "Military object
678   detection in defense using multi-level capsule networks." Soft Computing (2021): 1-15.
679   21. Li, Yanfen, Hanxiang Wang, L. Minh Dang, Tan N. Nguyen, Dongil Han, Ahyun Lee,
680   Insung Jang, and Hyeonjoon Moon. "A deep learning-based hybrid framework for object
681   detection and recognition in autonomous driving." IEEE Access 8 (2020): 194228-194239.
682   22. Jiang, Qingsheng, Dapeng Tan, Yanbiao Li, Shiming Ji, Chaopeng Cai, and Qiming Zheng.
683   "Object detection and classification of metal polishing shaft surface defects based on
684   convolutional neural network deep learning." Applied Sciences 10, no. 1 (2019): 87.
685   23. Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once:
686   Unified, real-time object detection." In Proceedings of the IEEE conference on computer
687   vision and pattern recognition, pp. 779-788. 2016.
688   24. Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic
689   segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern
690   recognition. 2015: 3431-3440.
691   25. Yuan, Yuhui, Xiaokang Chen, Xilin Chen, and Jingdong Wang. "Segmentation transformer:
692   Object-contextual representations for semantic segmentation." arXiv preprint
693   arXiv:1909.11065 (2019).
694   26. Chen, Liang-Chieh, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam.
695   "Encoder-decoder with atrous separable convolution for semantic image segmentation."
696   In Proceedings of the European conference on computer vision (ECCV), pp. 801-818. 2018.
697   27. Xiao, Tete, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. "Unified perceptual
698   parsing for scene understanding." In Proceedings of the European conference on computer
699   vision (ECCV), pp. 418-434. 2018.

700 28. Masood, Sharjeel, Fawad Ahmed, Suliman A. Alsuhibany, Yazeed Yasin Ghadi, M. Y. Siyal,
701   Harish Kumar, Khyber Khan, and Jawad Ahmad. "A deep learning-based semantic
702   segmentation architecture for autonomous driving applications." Wireless Communications
703   and Mobile Computing 2022 (2022).

704 29. Hu, Guang, Zhengwang Hu, Jiangping Liu, Fei Cheng, and Daicheng Peng. "Seismic Fault
705   Interpretation Using Deep Learning-Based Semantic Segmentation Method." IEEE
706   Geoscience and Remote Sensing Letters (2020).

707 30. Khan K, Mauro M, Leonardi R. Multi-class semantic segmentation of faces[C]//2015 IEEE
708   International Conference on Image Processing (ICIP). IEEE, 2015: 827-831.

709 31. Du, Zhenrong, Jianyu Yang, Cong Ou, and Tingting Zhang. "Smallholder crop area mapped
710   with a semantic segmentation deep learning method." Remote Sensing 11, no. 7 (2019): 888.

711 32. de Souza Inácio A, Brilhador A, Lopes H S. Semantic segmentation of clothes in the context
712   of soft biometrics using deep learning methods[C]//Anais do 14o Congresso Brasileiro de
713   Inteligência Computacional. Curitiba, PR: ABRICOM. 2019: 1-7.

714 33. LIANG, Xin-yu, Xi-kun LIN, Ji-chuan QUAN, and Kai-hong XIAO. "Research on the
715   Progress of Image Instance Segmentation Based on Deep Learning." *ACTA ELECTONICA*
716   *SINICA* 48, no. 12 (2020): 2476.

717 34. He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. "Mask r-cnn."
718   In Proceedings of the IEEE international conference on computer vision, pp. 2961-2969.
719   2017.

720 35. Chen G, Sun P, Shang Y. Automatic fish classification system using deep learning[C]//2017
721   IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI). IEEE,
722   2017: 24-29.

723 36. Akgül T, Çalik N, Töreyın B U. Deep Learning-Based Fish Detection in Turbid Underwater
724   Images[C]//2020 28th Signal Processing and Communications Applications Conference
725   (SIU). IEEE, 2020: 1-4.

726 37. Li X, Tang Y, Gao T. Deep but lightweight neural networks for fish detection[C]//OCEANS
727   2017-Aberdeen. IEEE, 2017: 1-5.

728 38. Wang, Jung-Hua, Shih-Kai Lee, Yi-Chung Lai, Cheng-Chun Lin, Ting-Yuan Wang, Ying-
729   Ren Lin, Te-Hua Hsu, Chang-Wen Huang, and Chung-Ping Chiang. "Anomalous behaviors
730   detection for underwater fish using ai techniques." IEEE Access 8 (2020): 224372-224382.

731 39. Alshdaifat N F F, Talib A Z, Osman M A. Improved deep learning framework for fish
732   segmentation in underwater videos[J]. Ecological Informatics, 2020, 59: 101121.

733 40. Knausgård, Kristian Muri, Arne Wiklund, Tonje Knutsen Sørdalen, Kim Tallaksen
734   Halvorsen, Alf Ring Kleiven, Lei Jiao, and Morten Goodwin. "Temperate fish detection and
735   classification: a deep learning based approach." Applied Intelligence 52, no. 6 (2022): 6988-
736   7001.

737 41. Parida P, Bhoi N. Dual transition region extraction based colour image segmentation:
738   Application to fish image segmentation[J]. Global Journal of Computer Science and
739   Technology, 2018.

740   42. Lei X, Ouyang H, Xu L. Image segmentation based on equivalent three-dimensional entropy
741   method and artificial fish swarm optimization algorithm[J]. Optical Engineering, 2018,
742   57(10): 103106.
743   43. Raza K, Song H. Fast and accurate fish detection design with improved YOLO-v3 model
744   and transfer learning[J]. International Journal of Advanced Computer Science and
745   Applications, 2020, 11(2).
746   44. Cai, Kewei, Xinying Miao, Wei Wang, Hongshuai Pang, Ying Liu, and Jinyan Song. "A
747   modified YOLOv3 model for fish detection based on MobileNetv1 as
748   backbone." Aquacultural Engineering 91 (2020): 102117.
749   45. Arvind, C. S., R. Prajwal, Prithvi Narayana Bhat, A. Sreedevi, and K. N. Prabhudeva. "Fish
750   detection and tracking in pisciculture environment using deep instance segmentation."
751   In TENCON 2019-2019 IEEE Region 10 Conference (TENCON), pp. 778-783. IEEE, 2019.
752   46. Ma, Jiaqi, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. "Modeling task
753   relationships in multi-task learning with multi-gate mixture-of-experts." In Proceedings of
754   the 24th ACM SIGKDD international conference on knowledge discovery & data mining,
755   pp. 1930-1939. 2018.
756   47. Lu, Kangjie, Jingwen Huang, Junhui Li, Jiashun Zhou, Xianliang Chen, and Yunfei Liu.
757   "MTL-FFDET: A Multi-Task Learning-Based Model for Forest Fire Detection." Forests 13,
758   no. 9 (2022): 1448.
759   48. Zhao, Hengshuang, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. "Pyramid
760   scene parsing network." In Proceedings of the IEEE conference on computer vision and
761   pattern recognition, pp. 2881-2890. 2017.
762   49. Han, Kai, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. "Ghostnet:
763   More features from cheap operations." In Proceedings of the IEEE/CVF conference on
764   computer vision and pattern recognition, pp. 1580-1589. 2020.
765   50. Zhang, Hongyi, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. "mixup:
766   Beyond empirical risk minimization." arXiv preprint arXiv:1710.09412 (2017).
767   51. Li, Danyang, Houcheng Su, Kailin Jiang, Dan Liu, and Xuliang Duan. "Fish Face
768   Identification Based on Rotated Object Detection: Dataset and Exploration." Fishes 7, no. 5
769   (2022): 219.

Results of the instance segmentation networks.

1   **Table 1 :**

2   **Results of the instance segmentation networks.**

Model	mAP@0.5	FPS
Mask R-CNN	0.887	10.6
YOLACT	0.887	23.0
Cascade Mask R-CNN	0.916	8.8

Comparison of Object Detection Models.

1 **Table 2 :**

2 **Comparison of Object Detection Models.**

Model	Precision	Recall	F1-score	mAP@0.5	mAP@0.5:0.95	Inference @batch_size 1 (ms)
CenterNet	95.21%	92.48%	0.94	94.96%	56.38%	32
Yolov4s	84.24%	94.42%	0.89	95.28%	52.75%	10
Yolov5s	92.39%	95.38%	0.94	95.38%	58.31%	8
EfficientDet	88.14%	91.91%	0.90	95.19%	53.43%	128
RatinaNet	88.16%	93.21%	0.91	96.16%	57.29%	48

4 [1]

[1] **Refers from [1]**

YOLOv5s + segmentation head with different data enhancement tricks.

1 **Table 3 :**

2 **YOLOv5s + segmentation head with different data enhancement tricks.**

HSV_Aug	FocalLoss	Mosaic	MixUp	Fliplrud	Group
×	×	×	×	×	1
√	×	×	×	×	2
√	√	×	×	×	3
√	√	√	×	×	4
√	√	√	√	×	5
√	√	×	×	√	6
√	√	√	×	√	7
√	×	√	×	×	8
√	×	√	√	×	9
√	×	√	×	√	10
√	×	√	√	√	11
√	√	√	√	√	12

The experiment results of different data enhancement tricks.

1 **Table 4 :**

2 **The experiment results of different data enhancement tricks.**

Group	Precision	Recall	F1-score	mAP@0.5	Seg Acc	Seg mIoU	FPS
1	0.904	0.931	0.917	0.955	0.981	0.978	117.1
2	0.905	0.935	0.920	0.963	0.980	0.977	115.0
3	0.888	0.924	0.906	0.947	0.975	0.972	114.4
4	0.922	0.939	0.930	0.966	0.985	0.983	114.8
5	0.904	0.924	0.914	0.957	0.984	0.982	113.8
6	0.897	0.918	0.907	0.942	0.985	0.983	118.7
7	0.899	0.937	0.918	0.955	0.982	0.980	113.4
8	0.954	0.917	0.935	0.976	0.985	0.984	115.1
9	0.952	0.924	0.938	0.974	0.986	0.984	116.2
10	0.935	0.924	0.929	0.964	0.984	0.982	115.3
11	0.893	0.910	0.901	0.943	0.986	0.984	114.3
12	0.900	0.936	0.918	0.954	0.986	0.984	113.3

Experimental results after improving the semantic segmentation head and replacing the activation function.

YOLOv5s+segmentation head(C3) is Group 8 of Table 4, YOLOv5s+segmentation head(GhostC3) means replace C3 in the segmentation head with GhostC3, YOLOv5s +segmentation head(GhostC3)+HardSwish means replacing C3 in the segmentation head with GhostC3 and replacing the original Swish activation function with the HardSwish activation function.

1 **Table 5:**
2 **Experimental results after improving the semantic segmentation head and replacing the**
3 **activation function. YOLOv5s+segmentation head(C3) is Group 8 of Table 4,**
4 **YOLOv5s+segmentation head(GhostC3) means replace C3 in the segmentation head with**
5 **GhostC3, YOLOv5s +segmentation head(GhostC3)+HardSwish means replacing C3 in the**
6 **segmentation head with GhostC3 and replacing the original Swish activation function with**
7 **the HardSwish activation function.**

Group	Precision	Recall	F1-score	mAP@0.5	Seg Acc	Seg mIoU	FPS
YOLOv5s+segmentation head(C3)	0.954	0.917	0.935	0.976	0.985	0.984	115.1
YOLOv5s+segmentation head(GhostC3)	0.952	0.927	0.939	0.975	0.985	0.983	116.0
YOLOv5s +segmentation head(GhostC3)+ HardSwish	0.954	0.930	0.942	0.976	0.985	0.983	116.6

Comparison results of different models in segmentation task and detection task.

1   **Table 6 :**

2   **Comparison results of different models in segmentation task and detection task.**

Model	mAP@0.5	Seg Acc	Seg mIoU	FPS
FCN	\	0.982	0.965	8.6
OCRNet	\	0.985	0.967	10.1
Deeplabv3+	\	0.981	0.966	8.0
UPerNet	\	0.982	0.966	8.6
ANN	\	0.982	0.965	7.0
CcNet	\	0.982	0.965	6.9
DANet	\	0.982	0.966	8.0
PSPNet	\	0.981	0.965	9.1
Mask R-CNN	0.887	\	\	10.6
YOLACT	0.887	\	\	23.0
Cascade Mask R-CNN	0.916	\	\	8.8
YOLOv5s+segmentation head(GhostC3)+HardSwish **(Our method)**	0.976	0.985	0.983	116.6

Figure 1

The network structure of our model. It contains two tasks, which are the detection task and the segmentation task.
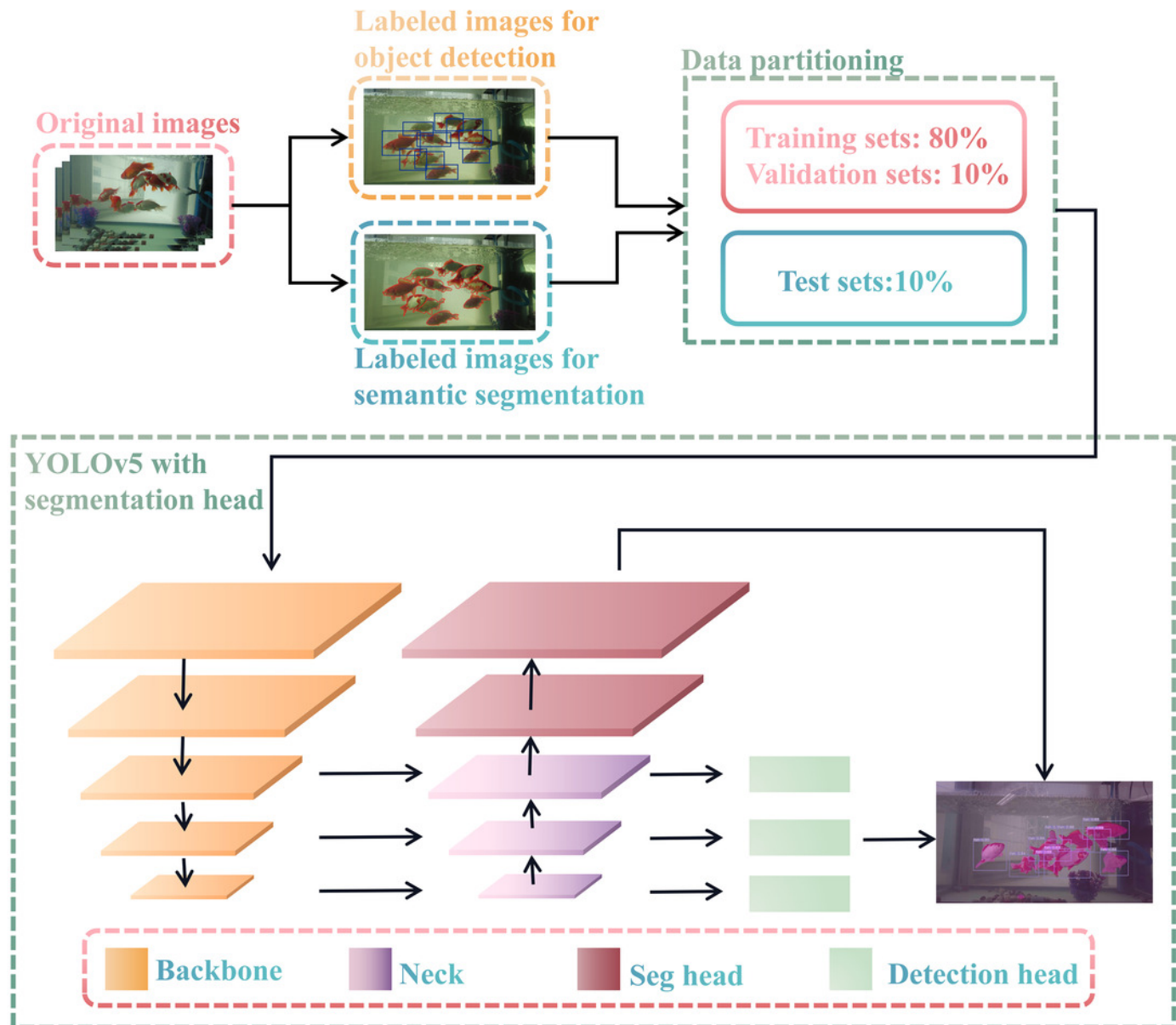
Figure 2

Annotation of object detection dataset.
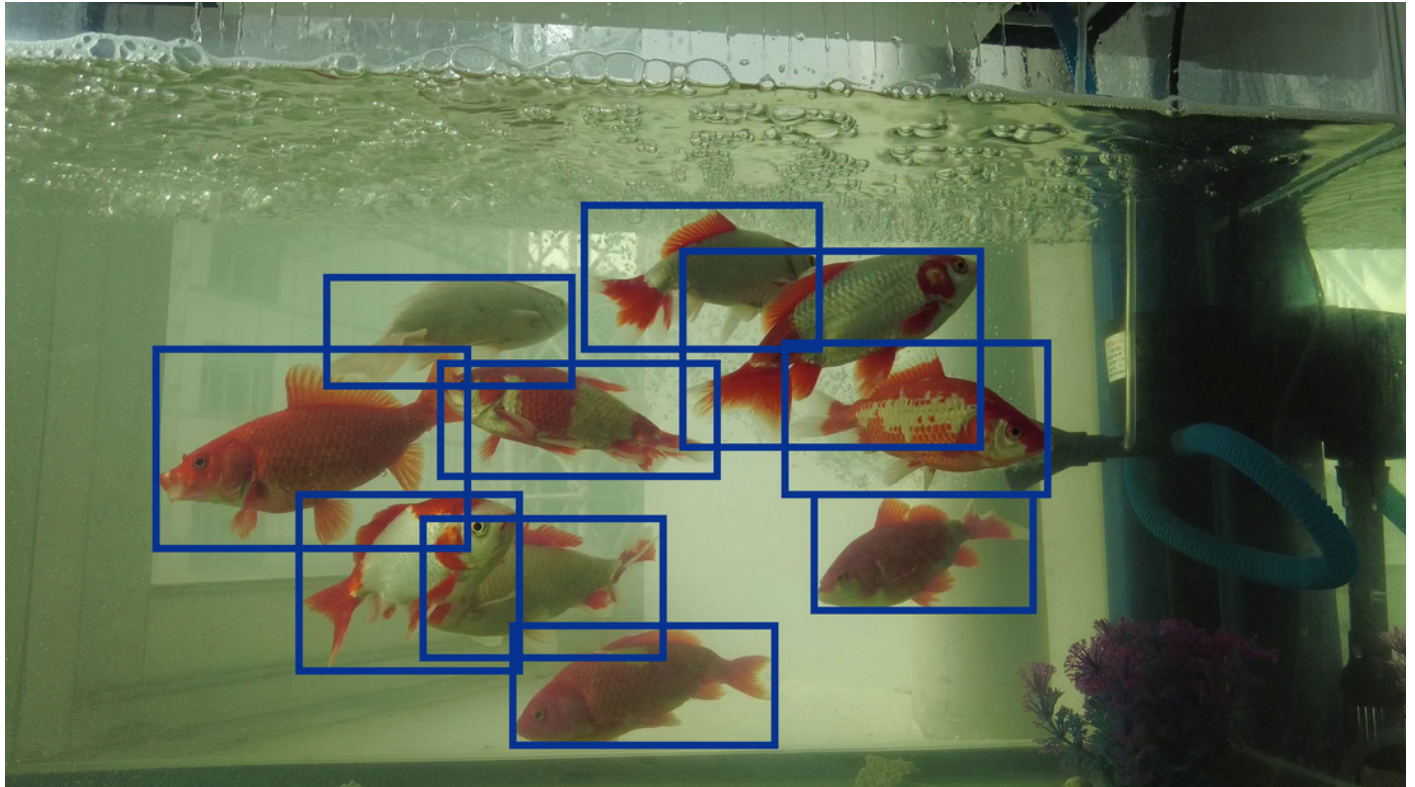
Figure 3

Annotation of segmentation dataset.

Figure 4

Figure (a) is a picture in our dataset, Figure (b) is a picture of Figure (a) flipped left and right, Figure (c) is a picture of Figure (a) flipped up and down, and Figure (d) is a picture of Figure (a) flipped up, down, left and right.

Figure 5

Training images after mosaic operation.
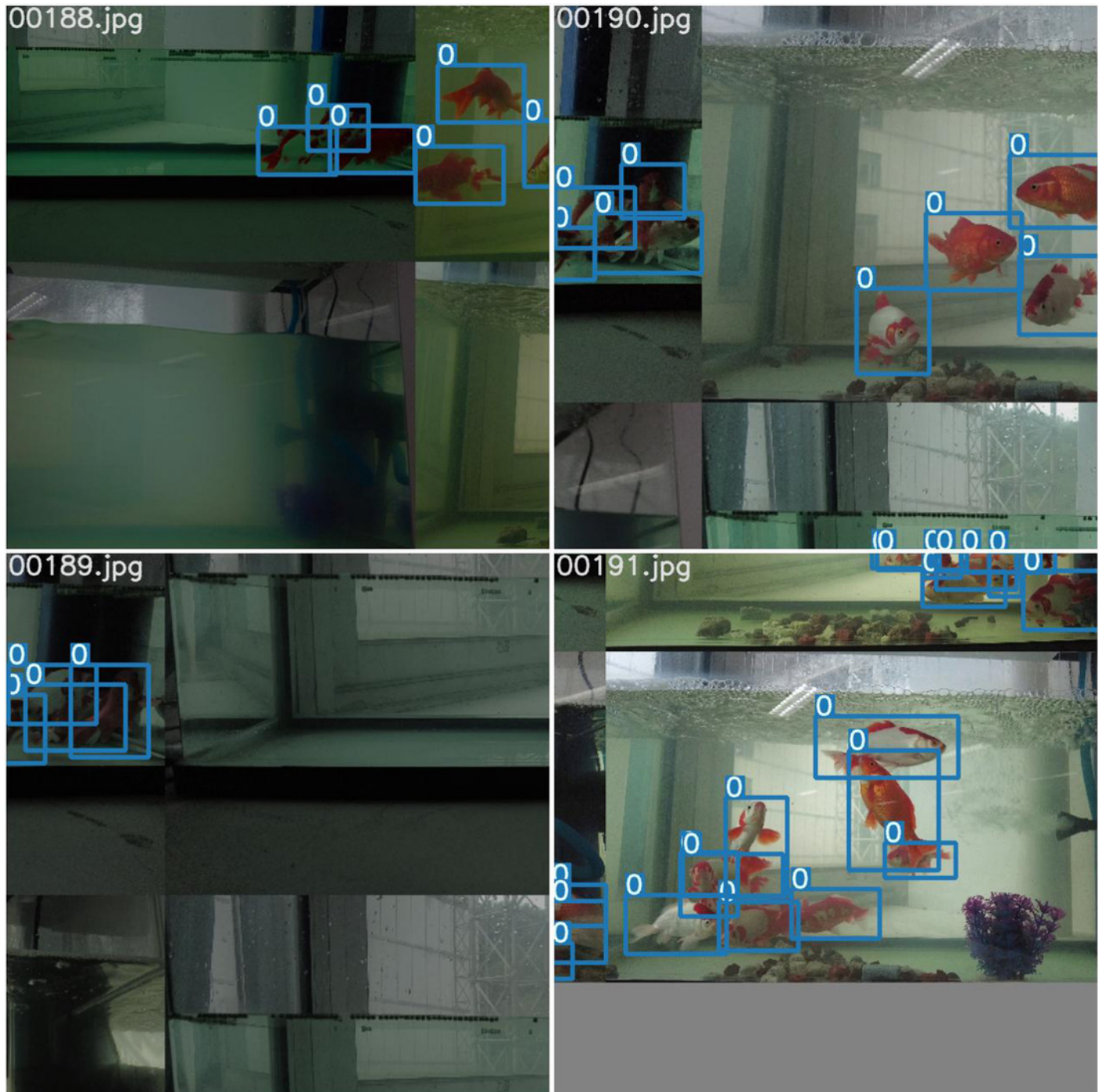
Figure 6

The network structure and application of YOLOv5s-5.0.

The CBS component in the figure is composed of the Convolutional layer + BatchNormalization + SiLU activation function. The Resunit component draws on the residual structure in the Resnet network and can play a role in building a deeper network. The CSP1_X and CSP2_X component draws on the CSPNet network structure. The FOCUS component is to slice the data, which can play a role in the down-sampling operation without information loss. The SPP component adopts the maximum pooling method of 1 × 1, 5 × 5, 9 × 9, and 13 × 13 for multi-scale fusion.
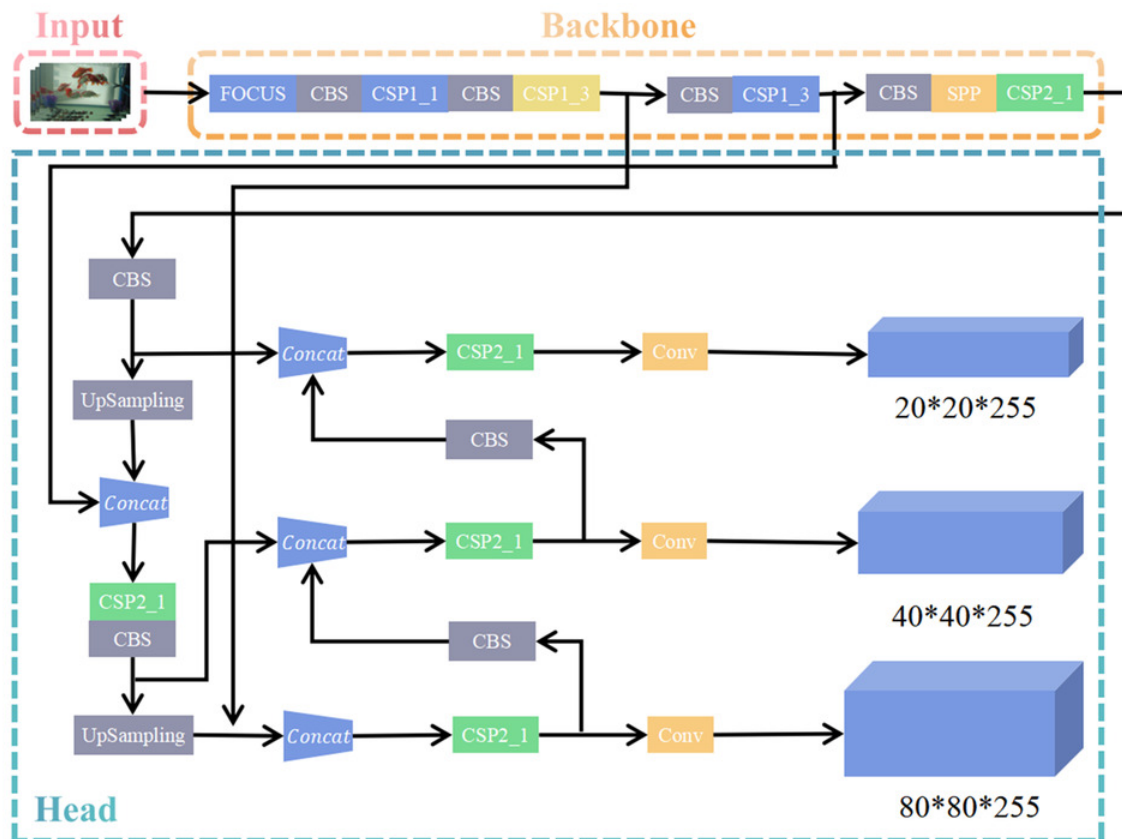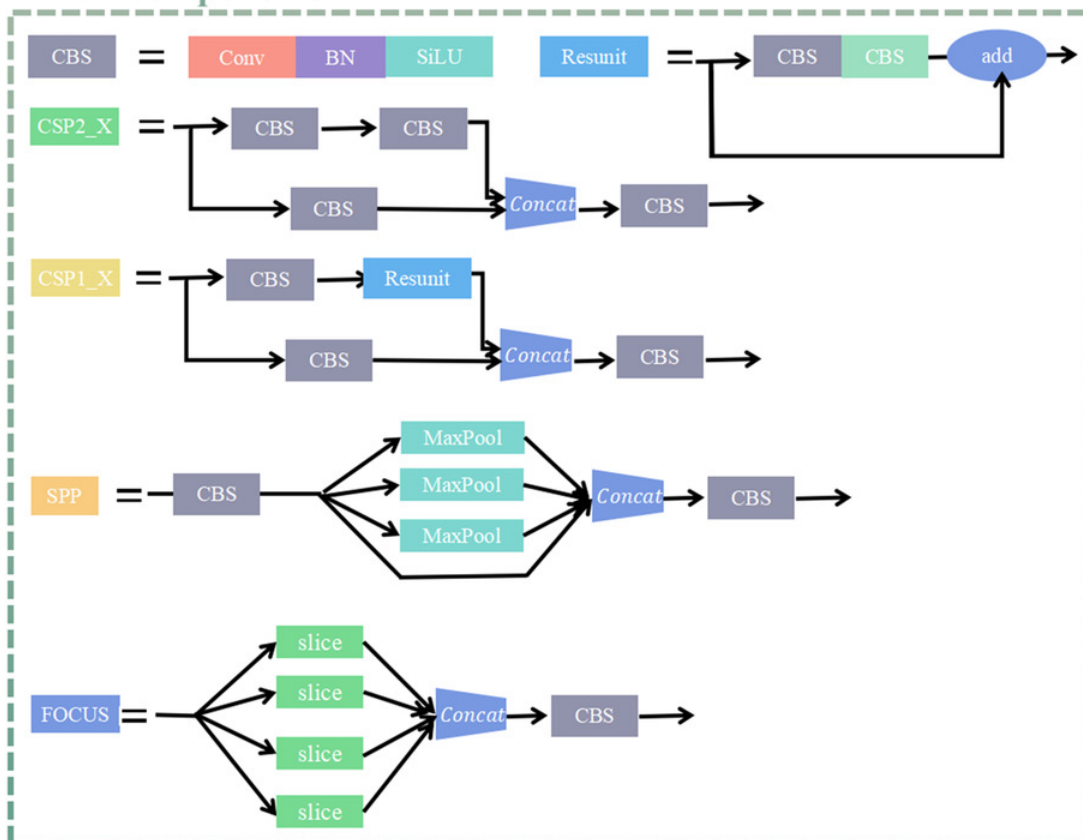
Figure 7

Structure of segmentation head.
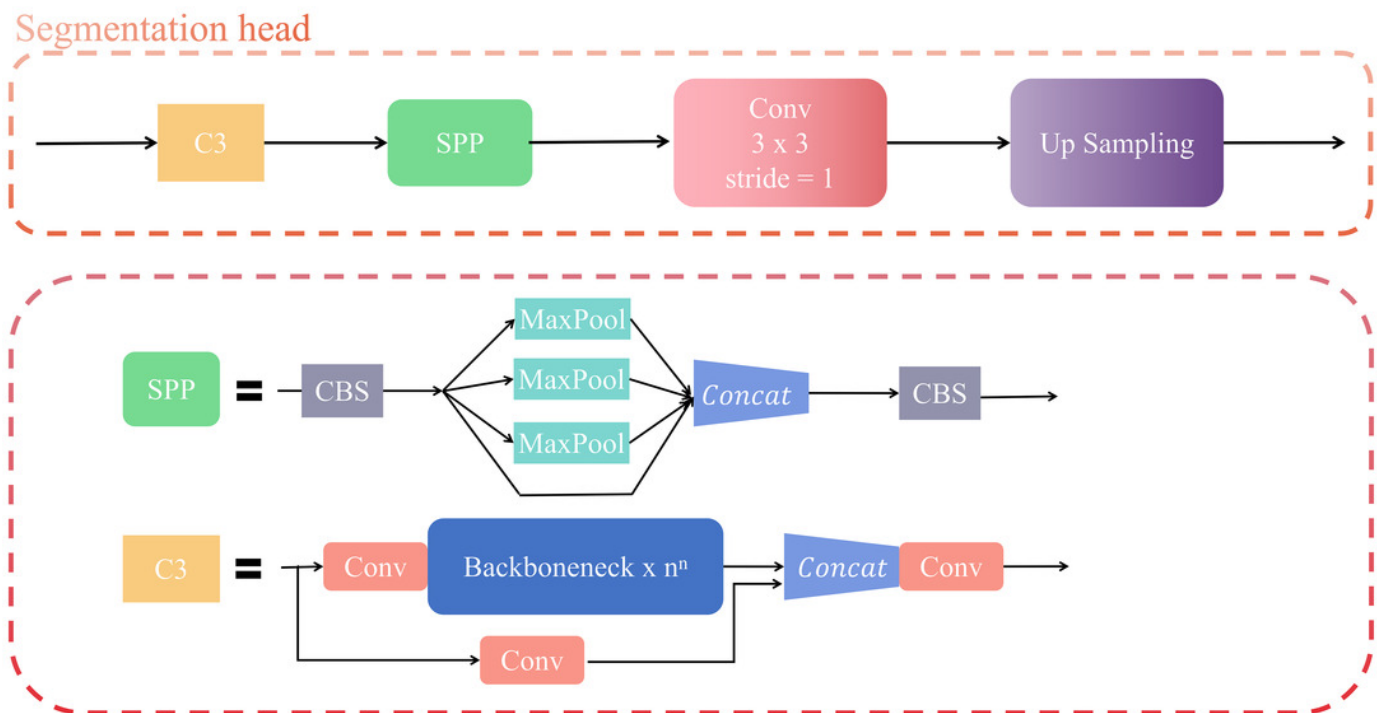
Figure 8

Example diagram of bilinear interpolation algorithm.
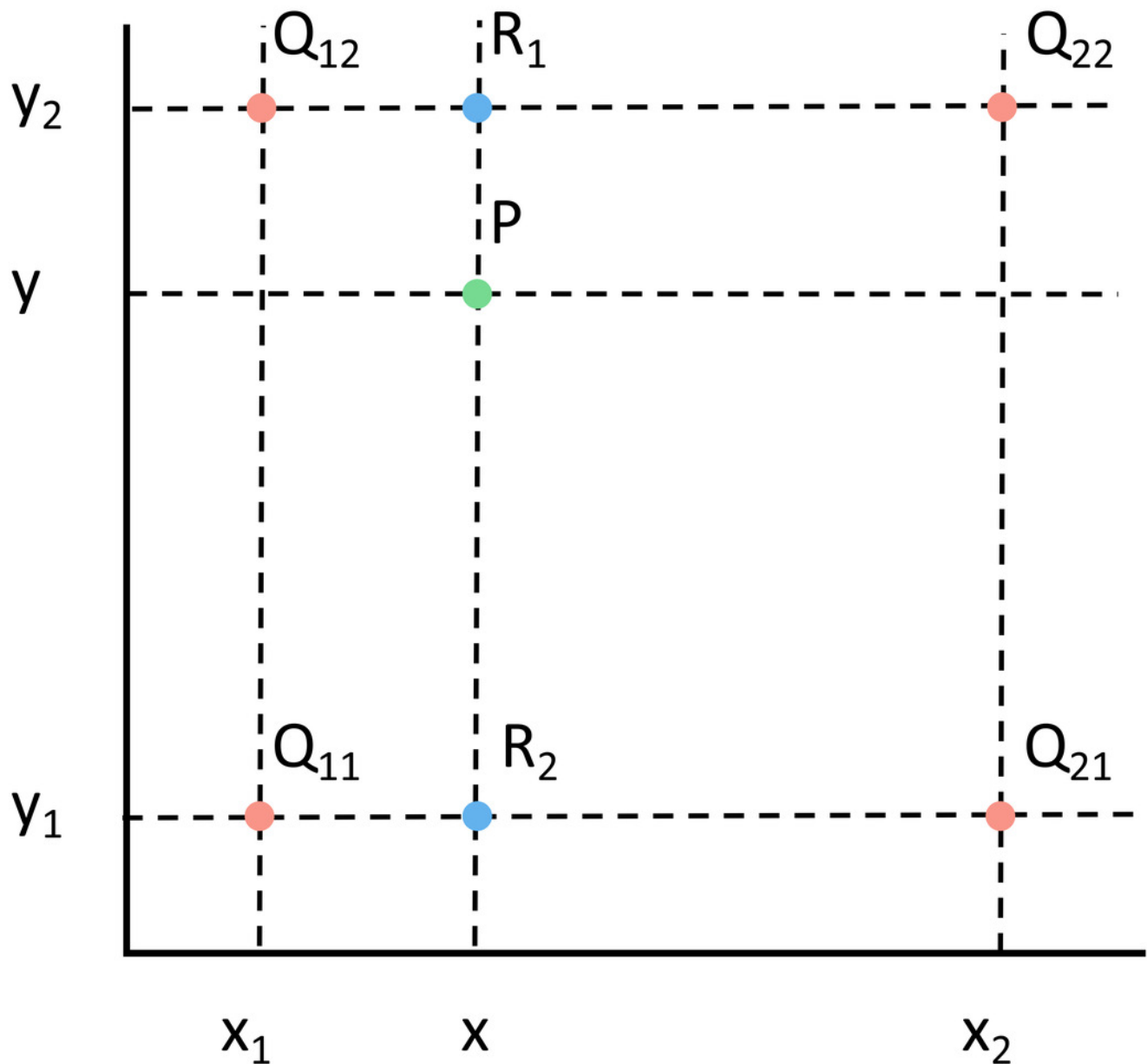
Figure 9

The structure of GhostC3 module.
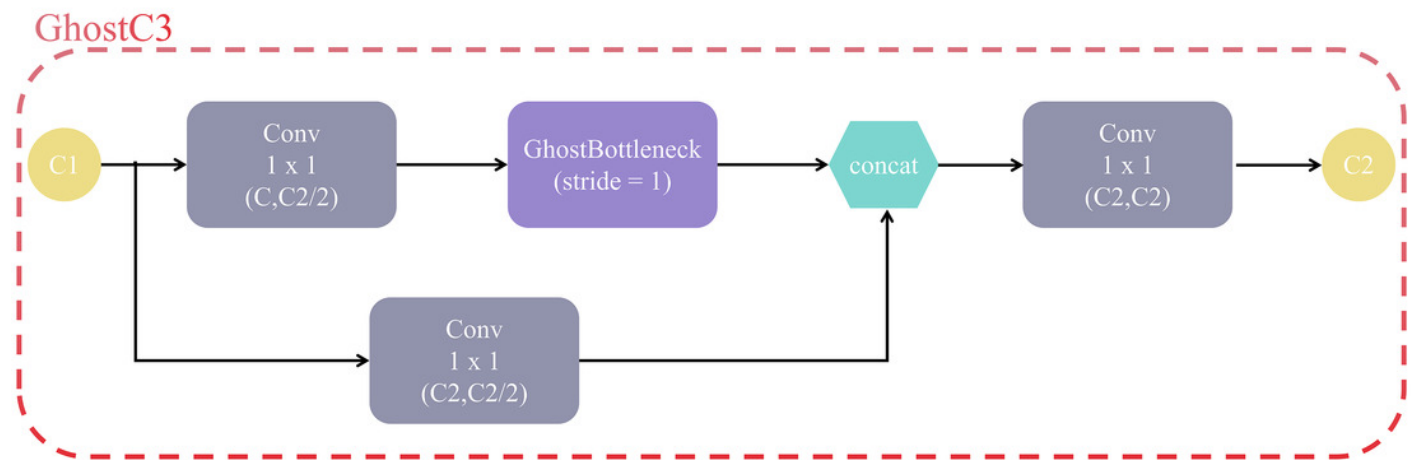
Figure 10

The structure of GhostConv module.
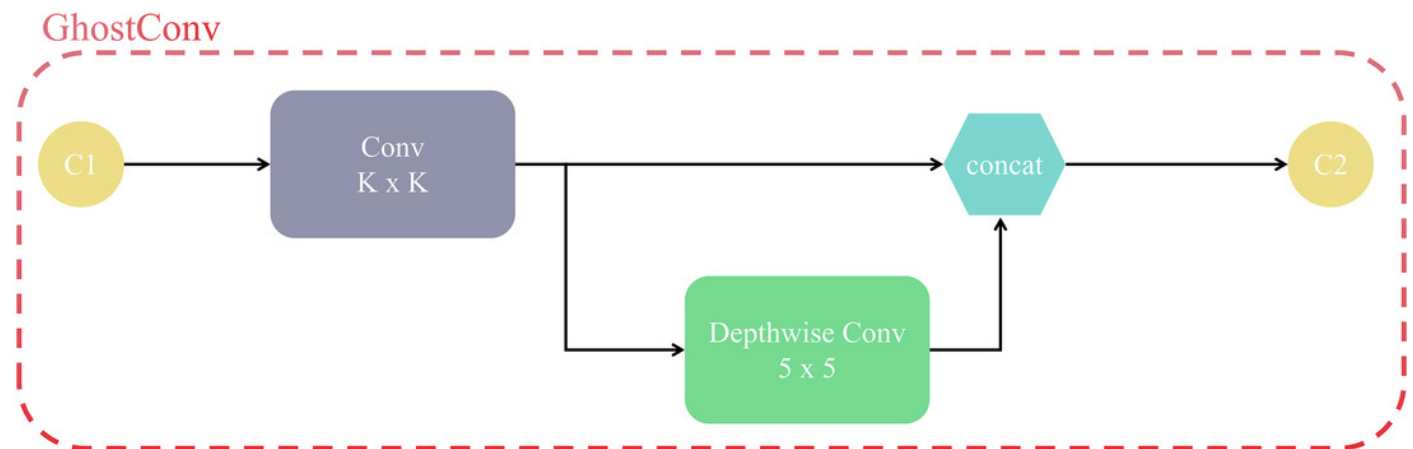
Figure 11

The structure of GhostBottleneck(stride=1).
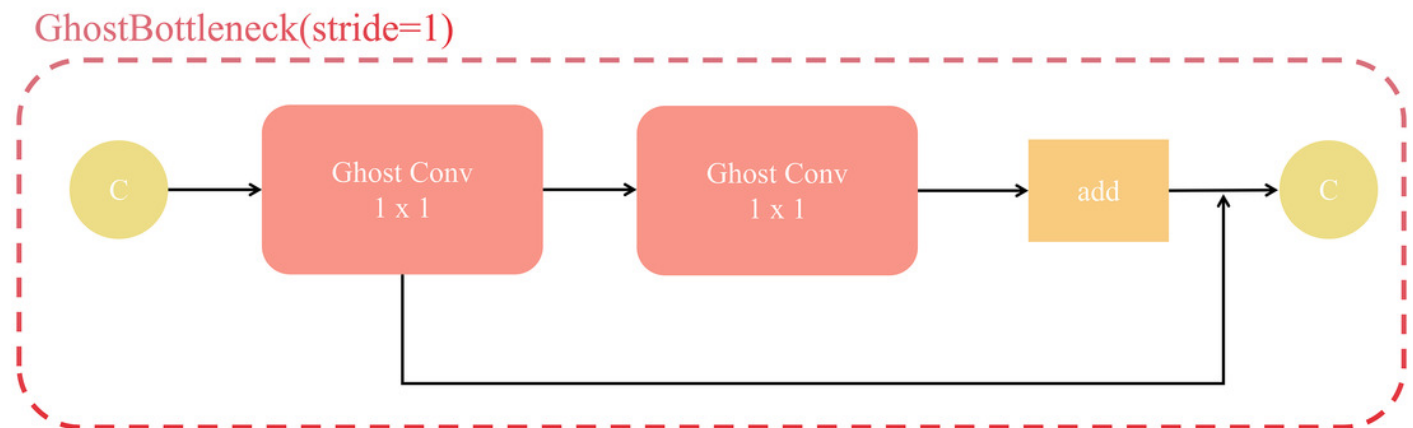
Figure 12

The structure of GhostBottleneck(stride=2).
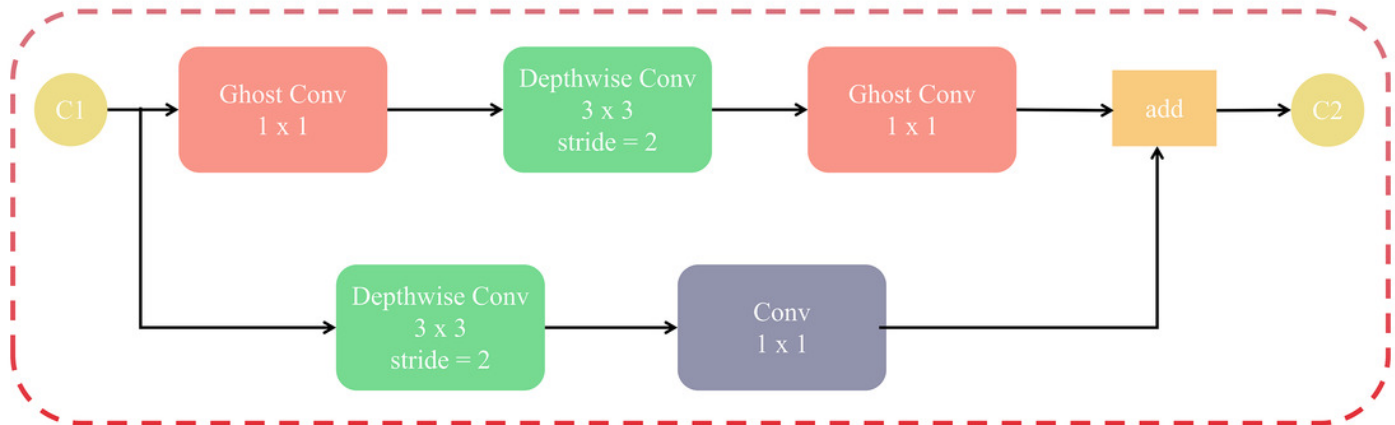
Figure 13

Swish function diagram.
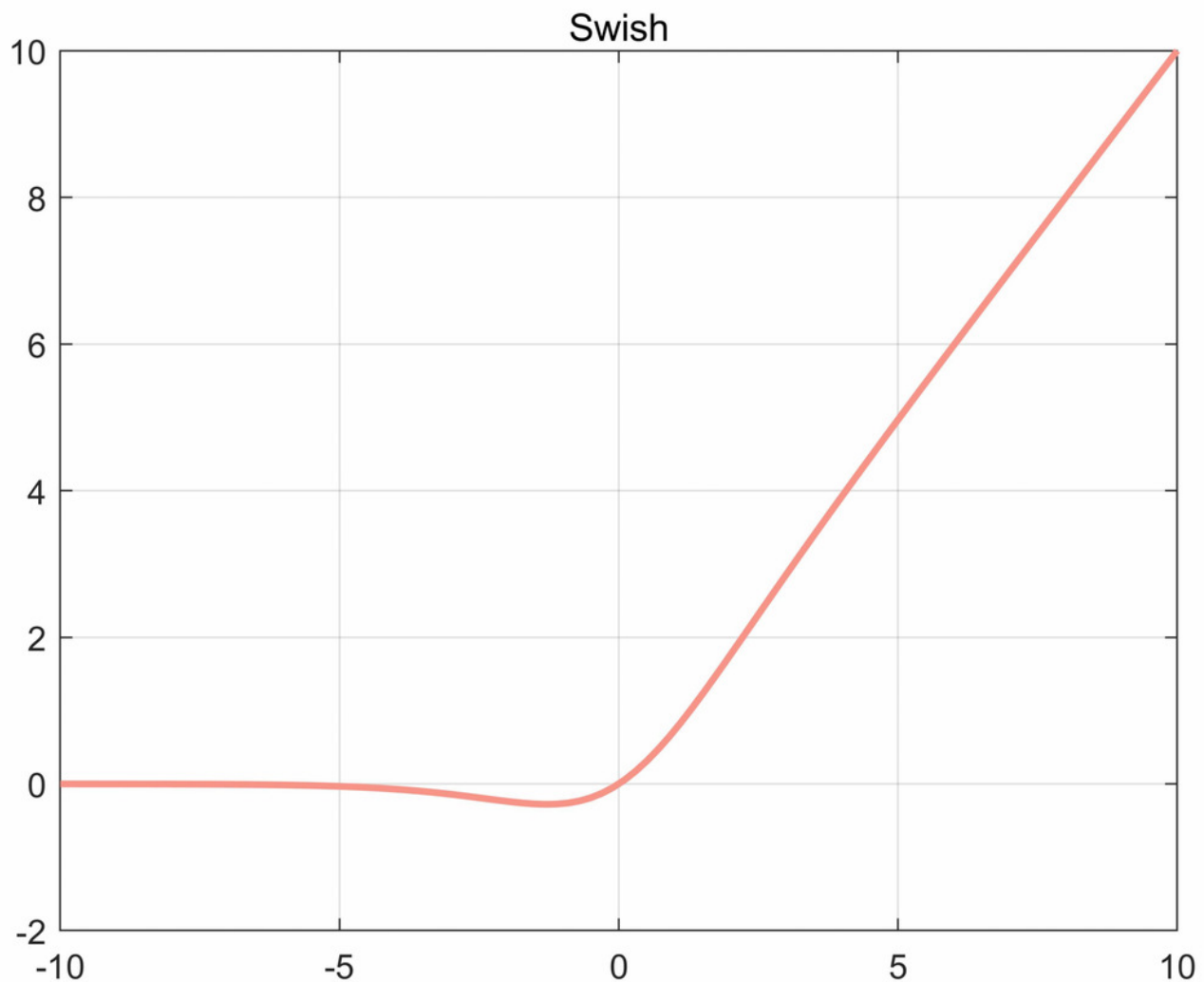
Figure 14

HardSwish function diagram.
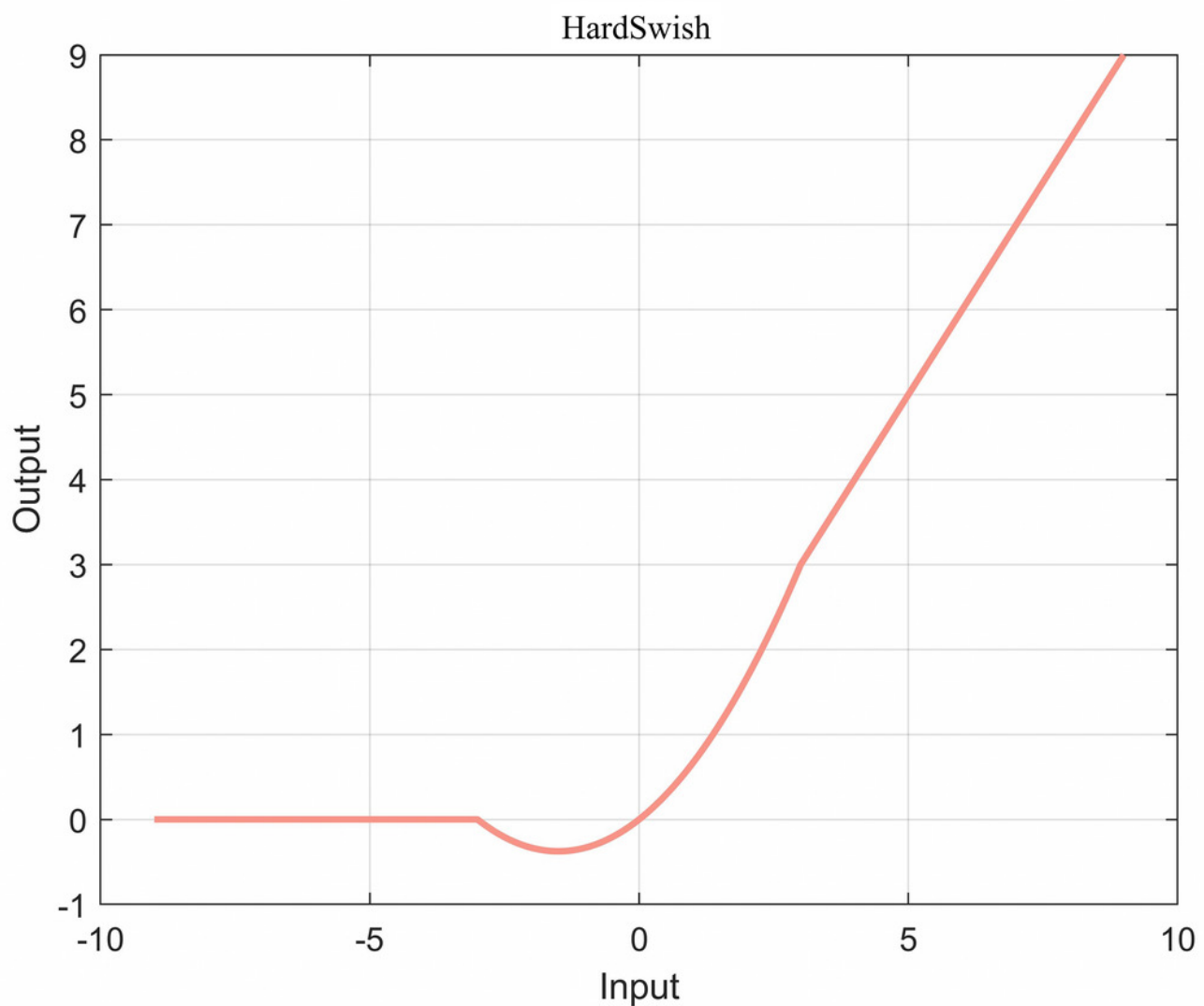
Figure 15

Visual representation of IoU.
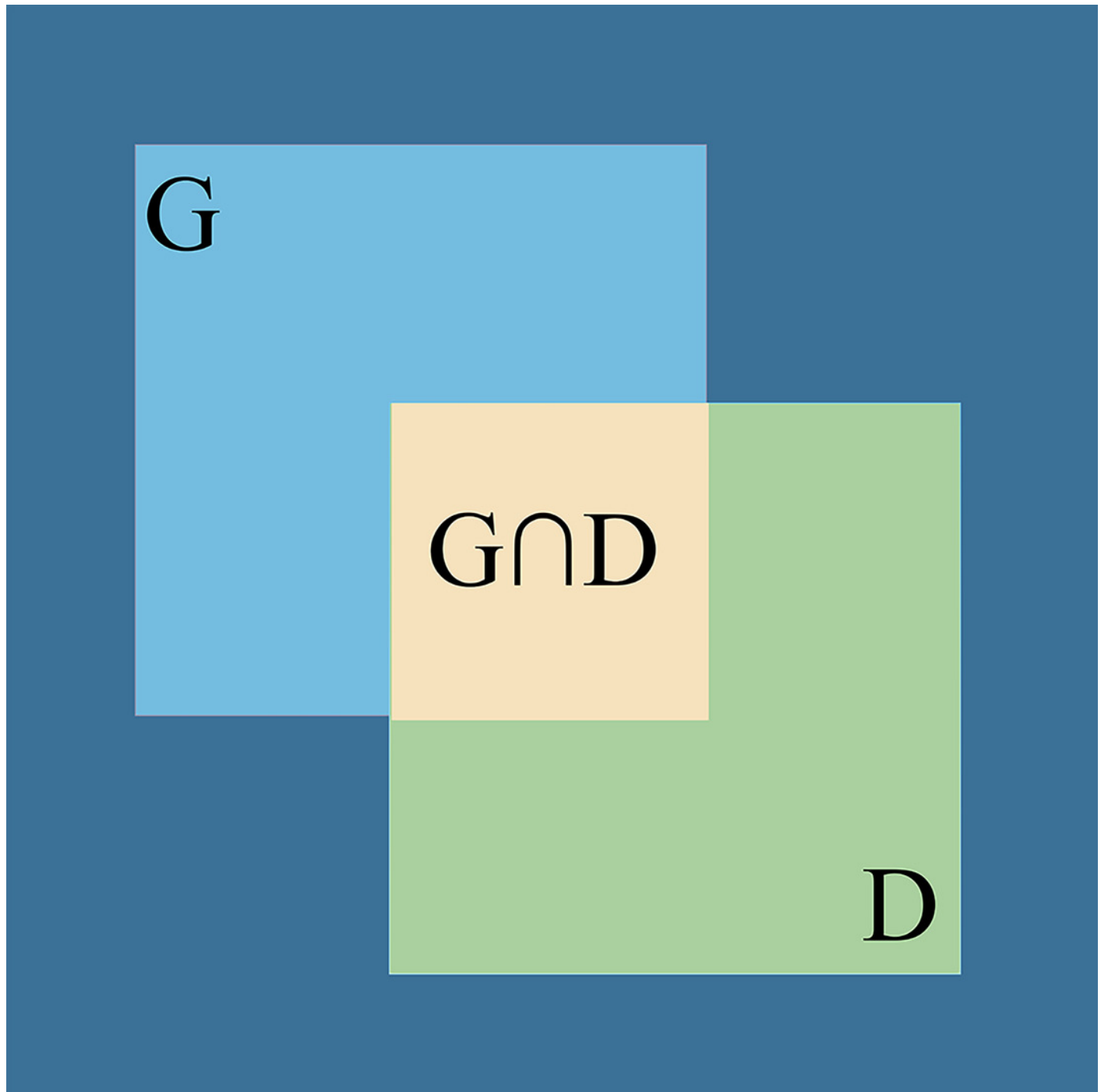
Figure 16

The results of our model for detecting and segmenting the golden crucian carp.