A smart chicken farming platform for chicken behavior identification and feed residual estimation

A smart chicken farming platform for chicken behavior identification and feed residual estimation

Jiezhi Yang†
Mashang Consumer Finance Co., Ltd
Chongqing 401121, China
jiezzzyoung@163.com

Chaochao Qu†
Mashang Consumer Finance Co., Ltd
College of Computer Science, Sichuan University
Chengdu 610065, China
qu_chaochao@stu.scu.edu.cn

Zirong Liu
Mashang Consumer Finance Co., Ltd
William Marsh Rice University, Houston, TX 77005, United States
zirong718@gmail.com

Kangzhe Ma
Mashang Consumer Finance Co., Ltd
Chongqing 401121, China
kangzhe.ma@msxf.com

Jun Gao
College of Computer Science, Sichuan University
Chengdu 610065, China
gaojun@stu.scu.edu.cn

Kuo Zhao
Mashang Consumer Finance Co., Ltd
Chongqing 401121, China
kuo.zhao@msxf.com

Ning Jiang
Mashang Consumer Finance Co., Ltd
Chongqing 401121, China
ning.jiang@msxf.com

Quan Lu
Mashang Consumer Finance Co., Ltd
Chongqing 401121, China
quan.lu@msxf.com

Yanyan Li
School of Mechanical Engineering, Sichuan University
Coleraine, United Kingdom
liyanyan@scu.edu.cn

Linjing Wei
School of Computer science and technology, Gansu Agriculture University
Lanzhou730070, China
wlj@gsau.edu.cn

Wenjing Tao
Key Laboratory of Freshwater Fish Reproduction and Development (Ministry of Education), Key Laboratory of Aquatic Science of Chongqing, School of Life Sciences, Southwest University, Chongqing, 400715, China
enderwin@163.com

Huiru Zheng
School of Computing, Ulster University
BelfastUlsterColeraine, United Kingdom
h.zheng@ulster.ac.uk

Antong Zhou
Mashang Consumer Finance Co., Ltd
Chongqing 401121, China
antong.zhou@msxf.com

Le Zhang*
College of Computer Science, Sichuan University
Chengdu 610065, China
zhangle06@scu.edu.cn

† These authors have contributed equally to this work and share first authorship.
* Correspondence: zhangle06@scu.edu.cn

***Abstract*—It is very potential to develop digital villages for promoting smart agriculture. As one of the important research fields of smart agriculture, smart chicken farms encounter management problems such as difficulties in quickly and accurately warning of sick and dead chickens and estimating feed residuals. Therefore, this study not only respectively proposed CKTrack and FRCM to detect sick and dead chickens and estimate feed residuals, but also developed a smart chicken farming platform for automagical management. Our main results include (1) the proposed CKTrack method can effectively identify sick and dead chickens under the condition of limited data volume and computing capacity; (2) the proposed FRCM method can accurately estimate the feed residuals; and (3) the smart chicken farming platform developed can provide farmers with functions such as early warning of sick and dead chickens, visualization of the chicken quantity inventory, and feed residual estimation.**

Keywords—computer vision; computational biology; object tracking; instance segmentation; smart chicken farming.

I. INTRODUCTION

Because of the difficulties of management in the large-scale chicken farming industry due to the crowded chickens and the complexity of the farm environment, it is neither possible to prevent large-scale disease and death, nor to estimate the feed residuals accurately. In addition, although traditional environmental monitoring systems can provide early warning and monitoring of the farm environment, they are unable to trace back to the health status of individual chickens and to estimate the feeding consumption of individual cages. To improve the chicken management capacity for the large-scale chicken farming industry, it is therefore important to develop such a smart farming system that can identify the health status of

979-8-3503-3748-8/23/$31.00 ©2023 IEEE

individual chickens and estimate the feeding consumption of individual cages.

Currently, there are two mainstream methods for real-time monitoring of agricultural product behavior. One is behavioral recognition using video data. For example, Han et al. [1] used deep learning technology[2-4]for model training and inference to predict pig behavior, they obtain over 96% accuracy of pig behavior. However, this pig image dataset required labeling a large amount of different state data, and the model is subjected to overfitting for the pig scenario. Moreover, it is not suitable for the cross-domain research except pig behavior identification. The other method carries out behavioral discrimination tasks by building up the deep learning models for object detection and multi-object tracking. For example, for the object detection task, they are usually processed by Faster-RCNN [5] and YOLO [3, 6, 7] series with high accuracy, but their detection speed is greatly dependent on the computing capacity. For multi-object tracking tasks, Wojke et al. [3] integrate Kalman filter [8] into CNN (convolutional neural network) and ReID model [9] to track targets, which ca decrease the error rate of identification (ID) in the presence of occlusion, while it requests a large amount of annotated data for ReID model training

For the first mainstream method, because most of chicken behavior are normal walking condition and abnormal behavior (sickness and death) of the chicken are not easy to collect, the data is imbalance. It results in a long-tailed distribution of the collected chicken behavior data, which is not good for model training.

For the second mainstream method, chickens usually live in crowded scenes, and factors such as shading, movement and inconspicuous texture features can make us hard to capture the individual identity information of chickens. Moreover, the limited computing capacity of chicken farms cannot support high performance neural network computing. For this reason, we propose **our first research question**: how to develop a lightweight object detection algorithm that can identify the health status of chickens with limited training data and computing capacity?

For single cage feed residual estimation, as it is essentially an image segmentation problem, previous studies usually employ instance segmentation to do it. Hafiz et al.[10] indicated that there are four types of instance segmentation. (1) The first is the classification of mask proposals. For example, Zhou et al. [11] proposed a U-shaped encoder and decoder framework to segment instances of input size by layer recovery manner, but it is often not easy to recognize the continuous objects of the overlapping instances. (2) The second is the detection followed by segmentation. For example, He et al. [12] proposed the MASK-RCNN architecture, which added an object detection branch to the instance segmentation task to circumvent the instance overlap problem with multi-task learning, but it has great deviations in video scenes, especially for the moving and continuous instances from the same class. (3) The third is labelling pixels followed by clustering. For example, Chen et al. [13] guided the precise human instance matting by segmentation network and predicting matting alpha task in high-resolution images, but it is not effective for greyscale images with low resolution. (4) The fourth approach is employing the dense sliding window method. For example, DeepMask [14] and InstanceFCN [15] use DenseNet (Densely Connected Convolution Networks) as feature extractor, which generate region proposals by sliding window, and then classifies the pixel points for each region. Although it can guarantee effective segmentation for discontinuous sparse instances, it will increase the inference cost of the model and decrease real-time performance.

Since the feed residuals information is discrete due to the non-uniform feeding habits of the chicken and the feeding line information is continuous, it is hard for us to employ the first and second approaches to segment the discrete and continuous instances simultaneously. Secondly, because the collected images are greyscale and low-resolution, we are hard to employ third approach to fit the model against the image data. Thirdly, limited to the cost of farming, it is impossible to employ the fourth type of approach, since the servers of chicken farms are usually unable to support large-scale and long-time high-performance computing. For these reasons, we propose **our second research question**: how to simultaneously carry out low-latency refinement segmentation for continuous and discrete instances for the low-resolution greyscale moving images to accurately obtain the single cage feed residuals.

Meanwhile, although smart farming systems have been implemented in many scenarios, they mostly focused on smart environmental control systems [16], smart body monitoring systems, and smart monitoring systems in large-scale agricultural product scenarios such as pig, cattle, and sheep, rather than chicken farming. Therefore, **our third research scientific question is** how to establish a visualized smart breeding system for chicken farm.

Here, we propose three innovations to address the above research questions. Firstly, we propose a multi-object tracking algorithm (CKTrack) to recognize chicken health status accurately and quickly without chicken identity and behavioral status data [3] by integrating Kalman filtering [8], Hungarian matching [17] and Intersection over Union (IOU) filtering into YOLO [6] object detection. Secondly, based on multi-task instance segmentation, dynamic threshold for binarization and moving normal vectors, we develop an analysis process for feed residuals computing (FRCM) to accurately estimate the feed residuals. Thirdly, a visual smart chicken farming web platform has been established, which provides farmers with a one-stop chicken farming service system with various features, including cost monitoring, asset inventory, risk warning and tracing back to the source of disease.

The research results demonstrate that (1) our developed CKTrack method can effectively identify sick and dead chickens under the condition of limited data volume and computing capacity; (2) our developed FRCM method can accurately estimate the feed residuals; (3) the smart chicken farming platform can provide farmers with functions such as early warning of sick and dead chickens, visualization of the chicken quantity inventory, and feed residual estimation.

II. MATERIALS AND METHODS

A. Data source

We collected surveillance videos from the chicken farm in 2023. After that, we generate the health status data (CKHS-2023) and feed residual data (CKFR-2023).

1) CKHS-2023 datasets

CKHS-2023 datasets consist of 4 videos. The duration of each video is over 20 minutes, includes about 30-100 chickens and has at least 1 sick or dead chicken. We take out the frame of the image every 5 seconds and label the chicken coordinates, ID, and health status on the image.

2) CKFR-2023 datasets

Referring to the lane line detection data production paradigm in the field of autonomous driving [17], CKFR-2023 datasets divide the annotation part of the feed information image into feed line and feed area (Supplementary Figure 1A).

The annotation of the feed area employs mask to visually showcase the feed residuals. Here, the white area of Figure 1B is the mask information of the feed area, whereas the green pixel area of Supplementary Figure 1C is its visualization result. To prevent the model from identifying areas outside the feed area as feed, we introduce a feed line to label the segmentation area out of the feed area. The labelling of the feed line can not only limit the visual information to the range of the feed area, but also display the moving trajectory of the feeder. Here, the white area of Supplementary Figure 1D is the mask information of the feed line, and the red pixel region of Supplementary Figure 1E are its visualization results.

B. CKTrack method

The CKTrack method is mainly composed of two modules: Detection and Health analysis, the workflow of which is shown by Figure 1.

1) Detection Module: This module is described by P1 of Figure 1. Firstly, the CKHS-2023 datasets are generated by Eq. 1.

$$Dataset = \{x \mid c_x, c_y, w, h, y_{cls}\} \tag{1}$$

where x represents the original image with chicken information, w and h represent the width and height of the rectangular box, respectively. And y_{cls} represents whether it is a chicken or not. Since the label of chicken is marked by a rectangular box, the center point (c_x, c_y) of the rectangular box can be calculated. Then, we used the YOLOv7 [7] to train the model by loss function (Eq. 2).

$$Loss_{cls} = -\frac{1}{n}\sum_{i=1}^{n}[y_{cls} \cdot log(\sigma(x)) + (1 - y_{cls})\, log(1 - \sigma(x))] \tag{2.1}$$

$$Loss_{iou} = 1 - \frac{b^{gt} \cap b^{pt}}{b^{gt} \cup b^{pt}} + \frac{\left|\left|P_{cx,cy}, P_{\widehat{cx},\widehat{cy}}\right|\right|_2^2}{c^2} \tag{2.2}$$

$$Loss_{all} = \lambda_1 Loss_{cls} + \lambda_2 Loss_{iou} \tag{2.3}$$

$$\lambda_1 + \lambda_2 = 1 \tag{2.4}$$

We defined the final loss function $Loss_{all}$ as the weighting function of $Loss_{cls}$ and $Loss_{iou}$. λ_1 and λ_2 represent their respective weighting factors. In addition, in $Loss_{cls}$, n is the batch size of the sample during the training process, and σ is the classification branch of the YOLOv7 model.

In addition, in $Loss_{iou}$, b^{gt} and b^{pt} represent the bounding box of ground truth and the predictive results for chicken, respectively. $P_{cx,cy}$, $P_{\widehat{cx},\widehat{cy}}$ are the centroids of the true and predictive box, and c is denoted as the distance from the upper left corner of the true box to the lower right corner of the predictive box.

Finally, we obtain the central coordinates ($\widehat{c_x}$, $\widehat{c_y}$), its width and height ($\hat{w}$, $\hat{h}$), and the predictive value of the category ($\hat{y}$)for the corresponding object by Eq.3.

$$\{\widehat{c_x}, \widehat{c_y}, \hat{w}, \hat{h}, \hat{y}\} = \text{model(x)} \tag{3}$$

It is noted that this study employs stochastic gradient descent (SGD) [18] method for model training.

2) Health analysis module: Health analysis module is described as in P2 of Figure 1 and Supplementary TABLE I, which consists of data conversion (B1 of P2 of Figure 1), Hungarian matching algorithm [17] based on Intersection over Union (IOU) strategy [19] (B2 of P2 of Figure 1) and Kalman filtering method [8] (B3-5 of P2 of Figure 1).

a) Data conversion – converting the $\widehat{c_x}, \widehat{c_y}, \hat{w}, \hat{h}, \hat{y}$ into x

$$\mathrm{x} = \{\mathrm{c_x}, \mathrm{c_y}, \mathrm{r}, \mathrm{h}, \mathrm{v_x}, \mathrm{v_y}, \mathrm{v_r}, \mathrm{v_h}\} \tag{4.1}$$

$$det = \{c_x, c_y, r, h, \} \tag{4.2}$$

$$OpticalTracks = \{x, P\} \tag{4.3}$$

where, (c_x, c_y) represent the center coordinates of chicken object, r is the aspect ratio, h is the height. v_x, v_y, v_r, v_h are the changes of velocity corresponding to the center point, aspect ratio and height, which are initialized to zero. x is the position of the object, represented by an 8-dimensional column vector; P is the covariance matrix, represented by an 8*8 diagonal matrix.

The optical tracks are randomly initialized using two variables (x and P). (Eq. 4.3)

b) IOU strategy based on Hungarian matching algorithm

$$\hat{X} = \{\hat{x}_i | \hat{x}_1, \hat{x}_2 \ldots \hat{x}_n\} X = \{x_i | x_1, x_2 \ldots x_n\} \tag{5.1}$$

$$IOU = \frac{\hat{x} \cap x}{\hat{x} \cup x} \tag{5.2}$$

$$matches = Hungarian\left(IOU(\hat{X}, X)\right) \tag{5.3}$$

X and $\hat{X}$ represent the sequences of target monitoring results and predictive results by Kalman filtering, respectively. We employ IOU strategy based Hungarian matching algorithm [17] to find the optimal matching object from the candidate objects that meet the matching criteria.

c) Kalman filtering method

Kalman filter is used to predict the possible location of the target at the next moment. Eq. 6 describes the Kalman filtering method, where F is the transition matrix and Q is the noise matrix.

$$tracks = \begin{cases} \hat{x} = F \cdot x \\ \hat{p} = FPF^T + Q \end{cases} \tag{6}$$

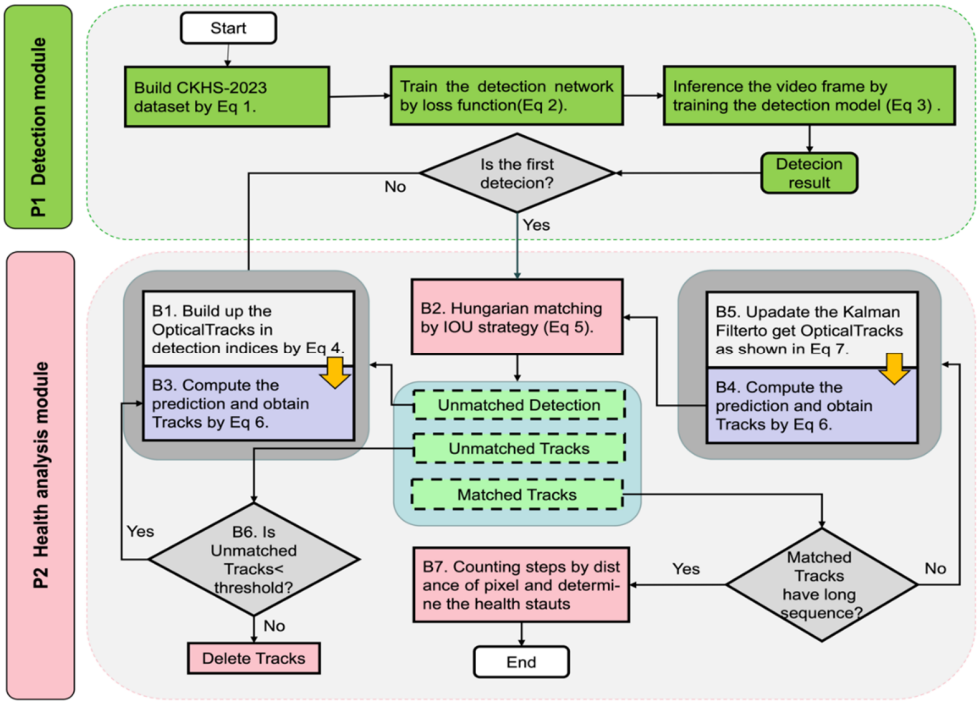

Figure 1. Workflow of CKTrack method. P1, P2, and B1-B7 are the index of figure.

$$y = z - H\hat{x} \tag{7.1}$$
$$S = H\hat{P}H^T + R \tag{7.2}$$
$$K = \hat{P}H^T S^{-1} \tag{7.3}$$
$$OpticalTracks = \begin{cases} x = \hat{x} \cdot Ky \\ P = \hat{P} - KH\hat{P} \end{cases} \tag{7.4}$$

The update phase of Kalman filter is listed by Eq. 7, where H matrix maps the predictive space to the observation space. Here, the difference between the observed value z and the predicted value $\hat{x}$ can be directly obtained.

Similarly, H matrix is introduced to map the covariance matrix $\hat{P}$ into the observation space. R is the noise matrix, and K represents the significance of the degree of estimation errors.

3) Data Validation Method: To investigate the distribution of the training data, Eq. 8 is used to compute the ratio ρ between the number of examples for majority and minority classes.

Indicated by Mateusz B et al. [20], if ρ<=10, the dataset is evenly distributed, otherwise it is long-tail distribution. C_i is the set of examples of classes.

$$\rho = \frac{max\{|C_i|\}}{min\{|C_i|\}} \tag{8}$$

C. Feed residuals computing method

Feed residuals computing method (FRCM) consists of segmentation module (P1 of Figure 2) and residual analysis module (P2 of Figure 2)

Segmentation module: Here, the weighted average of the feed line segmentation loss (Eq. 9.1 & 2) and the feed area segmentation loss (Eq. 9.3) are employed as the objective function for segmentation model training (Eq. 9.4). After training, we freeze the weights of segment module in inference status.

$$\begin{cases} L_{IOU} = 1 - \dfrac{TP}{TP + FN + FP} \\ F_{Fl}^{CE} = -\dfrac{1}{n}\sum_{i=1}^{n}\left[y_{line}^{i} \cdot log(\sigma(x^i)) + (1 - y_{line}^{i})\, log(1 - \sigma(x^i))\right] \end{cases} \tag{9.1}$$

In Eq. 9.1, TP, FN and FP represent the number of predicting positive samples as positive class (True Positive), the number of predicting positive samples t as negative class(False Negative), and the number of predicting negative samples as positive class (False positive) in the pixel space, respectively. y_{line}^{i} is the label of the feed line, x^i is the input image, σ is the segmentation model using YOLO layers [7] as backbone, and n is the number of samples.

$$L_{Fl} = \alpha L_{FL}^{IOU} + L_{FL}^{CE} \tag{9.2}$$

Eq. 9.2 represents the loss function of the feed line segmentation. In Eq. 9.3, y_{area}^{i} is represented as the label of feed area.The objective function of the final model is shown by Eq. 9.4, and Eq. 9.5 is used as the initial condition for λ.

$$L_{FA} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_{area}^{i} \cdot \log(\sigma(x^i)) + (1 - y_{area}^{i}) \log(1 - \sigma(x^i))\right] \tag{9.3}$$

$$L_{all} = \lambda_1 L_{FL} + \lambda_2 L_{FA} \tag{9.4}$$

$$\lambda_1 + \lambda_2 = 1 \tag{9.5}$$

1) Residual Analysis module:

We implemented the feed residual estimation by dynamic threshold for binarization (B3 of P2 of Figure 2) and the moving normal vectors (B4 of P2 of Figure 2).

a) Dynamic threshold for binarization method

$$T(x,y) = M_{r*r}(x,y) \cdot \left[1 + k \cdot \left(\frac{S_{r*r}(x,y)}{R} - 1\right)\right] \tag{10}$$

In Eq. 10, $T(x,y)$is the threshold of the current coordinate. $M(x,y)$ and $S(x,y)$ are the mean and standard deviation of the grey scale in the r*r neighborhood of the current coordinate, respectively. R and k are hyperparameters.

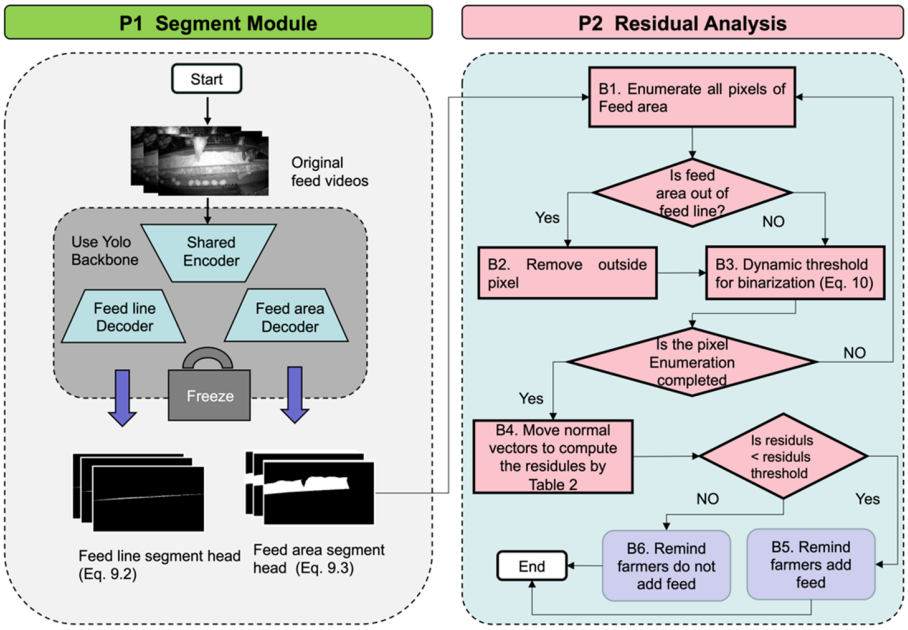

Figure 2. Feed residuals computing method. B1 and B2 are the index of figure.

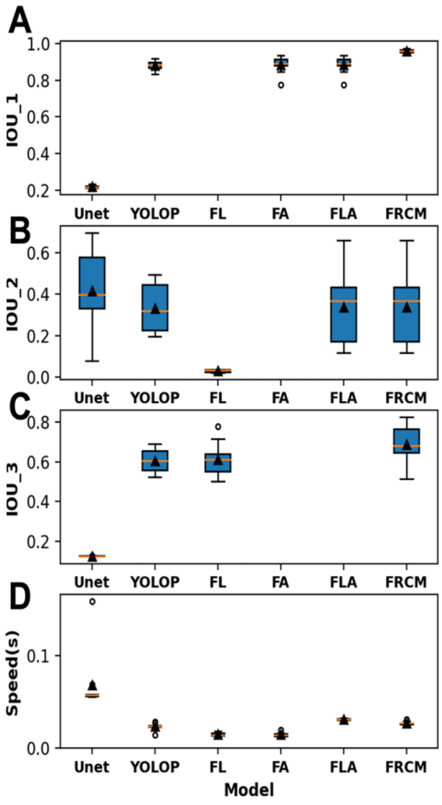

Figure 3. Segmentation performance comparison by IOU and Speed. It is noted that FL (feed line only) and FA (feed area only) are ablation models of FRCM. FLA denotes that the combination of FL and FA.

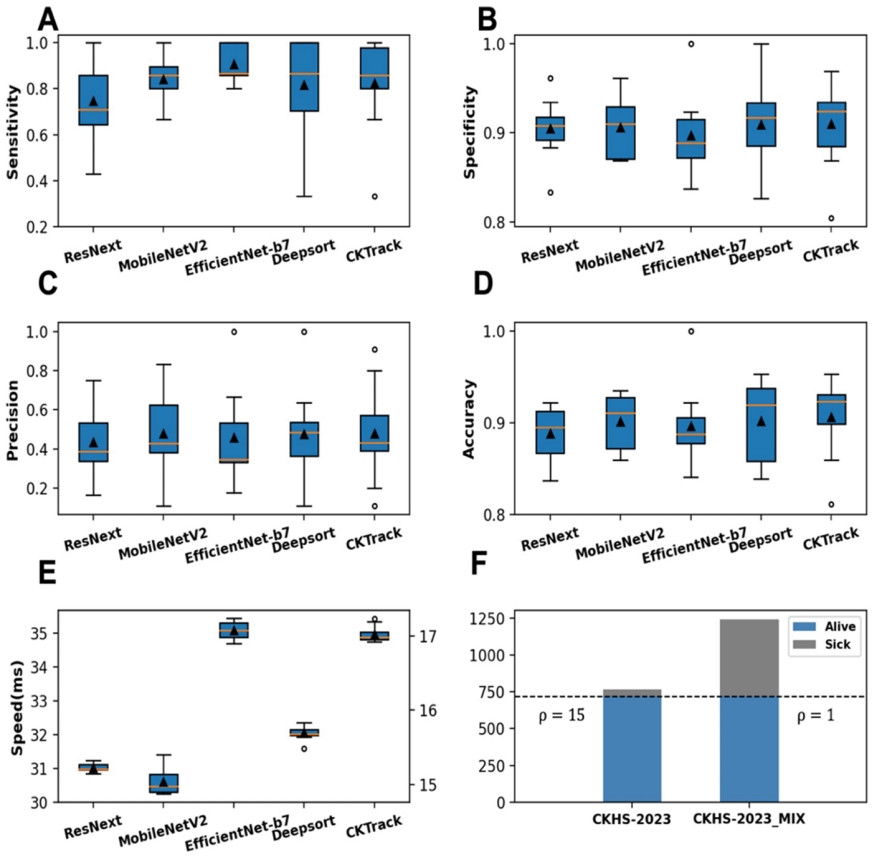

Figure 4. A-E: Inference performance comparison for the original classifier of ResNext, MobileNetV2, EfficientNet-b7, and tracking method of Deepsort, CKTrack by Sensitivity, Specificity, Precision, Accuracy and Speed. F: Overall datasets volume for CKHS-2023 and CKHS-2023-MIX.

b) Moving normal vector method

Firstly, moving normal vector method (Supplementary

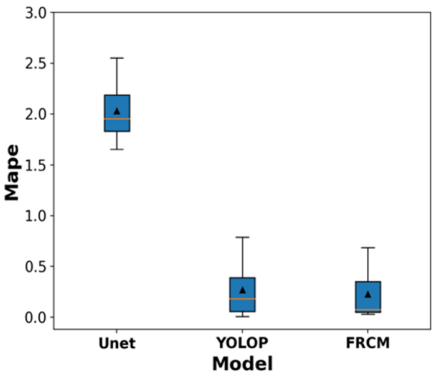

Figure 5. Segmentation performance comparison by Mape.

TABLE II) fixes the abscissa of image. Secondly, moving normal vector method traverses the number of segmented pixels in the feed area when the image is moving.Finally, based on the coordinates of the moving normal vector and the cage, we obtainthe feed area of each cage by constraining results.

D. Web server construction

The study develops a web server (Smart Chicken Farming Platform) with three components, which are IOT platform, main system, and AI system (Supplementary Figure 2).

In IOT platform, we use Mysql database [21] to store the device information of the camera sensors installed in the farm and employ S3 cloud storage [22] to store the video files recorded by the cameras.

In the main system, the back-end of the platform firstly parses the video data of the S3 cloud storage, and then stores the parsing results into the MySQL database. After the front-end of the platform sends requested data to the back-end, it reads the relevant static resources (such as digital twins, guiding animations, etc.) from the CDC object storage [23] service. Lastly, it visualizes these data.

In the AI system, the AI algorithm firstly pulls video data for inference from S3 cloud storage through kafka middleware [24]. And then, the inference results of the AI algorithm return to the main system through asynchronous interface interaction.

III. Results

A. The performance comparison among CKTrack and classical methods

This section is to answer the first scientific question: how to build up such a lightweight object detection algorithm that can identify the health status of chickens with limited training data and computing capacity.

Firstly, we compare the identification results of CKTrack and traditional classification models [25-27] in Section III.A.1.

1) Identification performance comparison

Figure 4 compares the identification effect of CKTrack, ResNext [25], MobileNetV2 [26], EfficientNet-b7 [27] and Deepsort [3] by sensitivity, specificity, precision and accuracy. The details of statistical testing[28-37] are shown in Supplementary TABLE III.

Figure 4A&B demonstrate that CKTrack performs slightly worse than EfficientNet-b7 and Deepsort in sensitivity and specificity, respectively. Figure 4C&D demonstrate that CKTrack outperforms other models in precision and accuracy.

Next, we will compare the inference speed among CKTrack and traditional classification models in Section III.A.2.

2) Speed performance comparison

Figure 4E compares the inference speed among CKTrack, ResNext [25], MobileNetV2 [26], EfficientNet-b7 [27]and Deepsort [3] which demonstrates that the inference speed of CKTrack is only slower than the MobileNetV2.

3) Dataset volume comparison

Finally, we compare the volume of training data requested by CKTrack and traditional classification models [25-27] in III.A.3.

Because datasets that are evenly distributed could meet the training request for deep learning models[29, 38-41], datasets with long-tailed distributions may not work as well as those with even distributions.

Since Figure 4F shows that CKHS-2023 dataset is long-tailed distributed with $\rho = 15$ (Eq. 10), it cannot effectively support the training procedure for deep learning model [25-27].

To support the training procedure for the traditional deep learning model [25-27], we build up a big CKHS-2023-MIX dataset for deep learning based models by obtaining the samples of sick and dead chickens from the Internet. Figure 4F also shows that CKTrack can work with a small dataset CKHS-2023 dataset.

B. Results of Feed Residuals computing methods

This section is to answer the second research question: How to simultaneously carry out low-latency refinement segmentation for continuous and discrete instances for the low-resolution greyscale moving images to accurately obtain the single cage feed residuals.

Firstly, section III.B.1 compares the segmentation results of FRCM and classical segmentation models for low resolution greyscale continuous and discrete instances by CKFR-2023 dataset.

1) Segmentation performance comparison

Supplementary TABLE IV uses four measurements (IOU_1, IOU_2, IOU_3, and MAPE) to assess the accuracy of the segmentation model.

Figure 3A-C compares the performance of Unet[11], YOLOP [24] , FA, FL, FLA by IOU_1, IOU_2 and IOU_3, which turns out that FRCM exceeds the classical segmentation model and the ablated models of FRCM. Next, we will compare the inference speed of FRCM and classical segmentation models in Section III.B.2.

2) Inference speed performance comparison

Figure 3D compares inference speed among FRCM, classical segmentation model, and the ablated model of FRCM, which demonstrates FRCM is slightly slower than the ablated model of FRCM with FL and FA, but faster than FLA. Finally, we compared the predictive accuracy between FRCM and classical segmentation models [11, 24] for single cage feed residuals in Section III.B.3.

3) Feed residual estimation comparison

Figure 5 compares the MAPE (predictive accuracy) between FRCM and classical segmentation models (Unet and YOLOP) [11, 24], which shows that the predictive accuracy of FRCM is better than the classical segmentation models [11, 24].

C. Visualization of Smart chicken farming platform

This section is to answer the third scientific question: how t o establish a visualized smart breeding system for chicken farm ing. Here, we built up a smart farming platform (https://wisdomxiaoji.msxf.com/wisdom-web/chickenCage?siteId=12&site=qd) with a cage chicken farming module (Supplementary Figure 3 A) and a free-range chicken farming module (Supplementary F igure 3B).

The platform is established using python, Java and Vue architecture. And the cage chicken farming module employs Unity-3D [42] to build up the digital twin system as shown in Supplementary Figure 3A, to enhance the management of chicken farms by farmers. The central area of P1 of Supplementary Figure 3A indicates the virtual projection of the breeding farm. Furthermore, we have updated the presentation of the FRCM algorithm's inference results and the automatic

inspection of the chicken coop, providing videos and reports in the relevant sections of Supplementary Figure 3A. In addition, the system is set up by B/S mode and accessed by users through SaaS (Software as a Service) [43]. Subsequently, asynchronous processing combined with middleware is used to manage the interaction of data processing such as video sequences.

IV. Discussion and Conclusion

It is very potential to develop the digital villages to promote the smart agriculture. To help farmers conveniently manage the large-scale chicken farms, we developed a CKTrack method to detect the sick and dead chickens; we innovated the feed residuals computing method (FRCM) for feed residual estimation; we built up a smart chicken farming platform with rich functions and data visualization for chicken management.

Figure 4 A-D and Supplementary TABLE III demonstrated that CKTrack is statistically better than the traditional classification models to identify sick and dead chicken. We explain this phenomenon from the following perspectives.

Firstly, since the samples of sick and dead chickens from CKHS-2023-MIX used in the traditional classification model do not come from actual farming scenarios, its classification effect is not better than CKTrack. Secondly, because CKTrack only uses the CKHS-2023 data to train the target detection task, and tracks the location of chickens by Kalman filtering, it is more accurate to identify the the sick and dead chickens than traditional classification models for our scenario. Thirdly, since CKTrack replaces the multi-tasking strategy of detection with the target detection with tracking strategy, the inference speed is faster than the traditional classification models. Although the inference speed of MobileNetV2 built with a small number of convolutional layers is slightly faster than the CKTrack, it is much worse than CKTrack to identify the sick and dead chickens (Figure 4C). Thus, we conclude that the CKTrack outperforms other classical classification model for our chicken faming study.

Since the FRCM we proposed employs multi-task learning strategy to simultaneously process the continuous (feed line) and discrete (feed area) instances as shown in Figure 3A, Figure 3B demonstrate that FRCM is better than the traditional segmentation models [11, 24]. On the other hand, the entire feed residuals estimation requests to cascaded connect FA to FL. Figure 3D demonstrates although FA can significantly reduce the inference time, the inference speed of the FLA is still slower than the FRCM. Therefore, we conclude that FRCM outperformances the current segmentation applications [11, 24] for our scenario.

Moreover, Supplementary Figure 3 turns out that the smart chicken farming platform not only provides farmers with early warning services for sick and dead chickens, but also can visualize the chicken quantity inventory and feed residuals.

Although this study made a great progress in developing a smart chicken farm, it still has the following shortcomings. Firstly, our developed CKTrack can only identify the location of sick and dead chickens, but it cannot predict the risk of infectious diseases such as avian influenza. Secondly, the generalized capability of FRCM is not good enough, which needs us re-collecting data for model training and testing, once we change the farming scenarios or the position of the camera. Finally, it costs too much to draw 3D models manually for the construction of smart chicken farming platform. For these reasons, we not only plan to build up an avian influenza predictive model [44] for the sick and dead chickens, but also prepare to replace the FRCM by Swin-T model with strong generalized ability [45] as well as consider employing 3D reconstruction [46] and rendering [47]to reduce the labor cost for the construction of the smart chicken farming platform .

Supplementary information

Supplementary information accompanies this paper at https://github.com/qcc-gif/Supplementary-for-BIBM_yang2023.git.

Acknowledgments

This work was supported by grants from National Natural Science Foundation of China [62372316], National Science and Technology Major Project (Nos. 2021YFF1201200 and 2018ZX10201002, China), Sichuan Science and Technology Program (2022YFS0048), Chongqing Technology Innovation and Application Development Project (CSTB2022TIAD-KPX0067).

References

[1] J. Han et al., "Evaluation of computer vision for detecting agonistic behavior of pigs in a single-space feeding stall through blocked cross-validation strategies," Computers and Electronics in Agriculture, vol. 204, p. 107520, 2023, doi: 10.1016/j.compag.2022.107520.

[2] G. H. Aly, M. A. E.-R. Marey, S. El-Sayed Amin, and M. F. Tolba, "YOLO V3 and YOLO V4 for Masses Detection in Mammograms with ResNet and Inception for Masses Classification," pp. 145-153, 2021.

[3] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," 2017 IEEE International Conference on Image Processing (ICIP), pp. 3645-3649, 2017, doi: 10.1109/icip.2017.8296962.

[4] J. Gao, P. Liu, G.-D. Liu, and L. Zhang, "Robust Needle Localization and Enhancement Algorithm for Ultrasound by Deep Learning and Beam Steering Methods," (in English), J Comput Sci Tech-Ch, vol. 36, no. 2, pp. 334-346, Apr 2021, doi: 10.1007/s11390-021-0861-7.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans Pattern Anal Mach Intell, vol. 39, no. 6, pp. 1137-1149, Jun 2017, doi: 10.1109/TPAMI.2016.2577031.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788, 2016, doi: 10.1109/cvpr.2016.91.

[7] S. Liu, Y. Wang, Q. Yu, H. Liu, and Z. Peng, "CEAM-YOLOv7: Improved YOLOv7 Based on Channel Expansion and Attention Mechanism for Driver Distraction Behavior Detection," IEEE Access, vol. 10, pp. 129116-129124, 2022, doi: 10.1109/ACCESS.2022.3228331.

[8] G. F. Welch, "Kalman filter," Computer Vision: A Reference Guide, pp. 1-3, 2020, doi: "https://doi.org/10.1016/B978-0-12-817588-0.00010-6".

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 2016, doi: 10.1109/cvpr.2016.90.

[10] A. M. Hafiz and G. M. Bhat, "A survey on instance segmentation: state of the art," International Journal of Multimedia Information Retrieval, vol.9, no. 3, pp. 171-189, 2020, doi: 10.1007/s13735-020-00195-x.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III

18, pp. 234-241, 2015, doi: https://doi.org/10.1007/978-3-319-24574-4_28.

[12] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2980-2988, 2017, doi: 10.1109/iccv.2017.322.

[13] X. Chen et al., "Robust Human Matting via Semantic Guidance," Proceedings of the Asian Conference on Computer Vision, pp. 2984-2999, 2022, doi: https://doi.org/10.1007/978-3-031-26284-5_37.

[14] Hamzah, I. W. Ordiyasa, and M. H. R. Najib, "DeepMask: face mask detection using GAN algorithm," Iran Journal of Computer Science, vol. 6, no. 1, pp. 13-19, 2022, doi: 10.1007/s42044-022-00114-9.

[15] J. Dai, K. He, Y. Li, S. Ren, and J. Sun, "Instance-sensitive fully convolutional networks," Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14, pp. 534-549, 2016, doi: https://doi.org/10.1007/978-3-319-46466-4_32.

[16] Z. Guo, "Design and practice of environmental regulation and conservation piggery(in Chinese)," Livestock and Poultry Industry, no. 05, pp. 20-21, 2014, doi: 10.19567/j.cnki.1008-0414.2014.05.012.

[17] H. W. Kuhn, "The Hungarian method for the assignment problem," Naval Research Logistics Quarterly, vol. 2, no. 1-2, pp. 83-97, 1955, doi: 10.1002/nav.3800020109.

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, no. 6088, pp. 533-536, 1986, doi: 10.1038/323533a0.

[19] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of Localization Confidence for Accurate Object Detection," vol. 11218, pp. 816-832, 2018, doi: https://doi.org/10.1007/978-3-030-01264-9_48.

[20] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," Neural Netw, vol. 106, pp. 249-259, Oct 2018, doi: 10.1016/j.neunet.2018.07.011.

[21] M. T. Ahammed and I. Khan, "Ensuring power quality and demand-side management through IoT-based smart meters in a developing country," Energy, vol. 250, 2022, doi: "https://doi.org/10.1016/j.energy.2022.123747".

[22] M. R. Palankar, A. Iamnitchi, M. Ripeanu, and S. Garfinkel, "Amazon S3 for science grids," Proceedings of the 2008 international workshop on Data-aware distributed computing, pp. 55-64, 2008, doi: 10.1145/1383519.1383526.

[23] M. B. Bokade, S. S. Dhande, and H. R. Vyavahare, "Framework Of Change Data Capture And Real Time Data Warehouse," Esrsa Publications, no. 4, 2013, doi: 10.17577/IJERTV2IS4698.

[24] D. Wu et al., "YOLOP: You Only Look Once for Panoptic Driving Perception," Machine Intelligence Research, vol. 19, no. 6, pp. 550-562, 2022, doi: 10.1007/s11633-022-1339-y.

[25] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated Residual Transformations for Deep Neural Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5987-5995, 2017, doi: 10.1109/cvpr.2017.634.

[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4510-4520, 2018, doi: 10.1109/cvpr.2018.00474.

[27] B. Koonce, "EfficientNet," in Convolutional Neural Networks with Swift for Tensorflow. Berkeley, CA: Apress, 2021, ch. Chapter 10, pp. 109-123.

[28] Y. Xia et al., "Exploring the key genes and signaling transduction pathways related to the survival time of glioblastoma multiforme patients by a novel survival analysis model," BMC Genomics, vol. 18, no. Suppl 1, p. 950, Jan 25 2017, doi: 10.1186/s12864-016-3256-3.

[29] Y. You, L. Zhang, P. Tao, S. Liu, and L. Chen, "Spatiotemporal Transformer Neural Network for Time-Series Forecasting," Entropy (Basel), vol. 24, no. 11, p. 1651, Nov 14 2022, doi: 10.3390/e24111651.

[30] Y. You et al., "Developing the novel bioinformatics algorithms to systematically investigate the connections among survival time, key genes and proteins for Glioblastoma multiforme," BMC Bioinformatics, journal article vol. 21, no. Suppl 13, p. 383, Sep 17 2020, doi: 10.1186/s12859-020-03674-4.

[31] L. Zhang, S. Fan, J. Vera, and X. Lai, "A network medicine approach for identifying diagnostic and prognostic biomarkers and exploring drug repurposing in human cancer," Comput Struct Biotechnol J, vol. 21, pp. 34-45, 2022/11/29/ 2023, doi: 10.1016/j.csbj.2022.11.037.

[32] L. Zhang, W. Bai, N. Yuan, and Z. Du, "Comprehensively benchmarking applications for detecting copy number variation," PLoS Comput Biol, vol. 15, no. 5, p. e1007069, May 2019, doi: 10.1371/journal.pcbi.1007069.

[33] L. Zhang, Z. Dai, J. Yu, and M. Xiao, "CpG-island-based annotation and analysis of human housekeeping genes," Brief Bioinform, vol. 22, no. 1, pp. 515-525, Jan 18 2021, doi: 10.1093/bib/bbz134.

[34] L. Zhang et al., "Revealing dynamic regulations and the related key proteins of myeloma-initiating cells by integrating experimental data into a systems biological model," Bioinformatics, vol. 37, no. 11, pp. 1554-1561, Jul 12 2021, doi: 10.1093/bioinformatics/btz542.

[35] L. Zhang et al., "MCDB: A comprehensive curated mitotic catastrophe database for retrieval, protein sequence alignment, and target prediction," Acta Pharm Sin B, vol. 11, no. 10, pp. 3092-3104, Oct 2021, doi: 10.1016/j.apsb.2021.05.032.

[36] L. Zhang et al., "Bioinformatic analysis of chromatin organization and biased expression of duplicated genes between two poplars with a common whole-genome duplication," Hortic Res, vol. 8, no. 1, p. 62, Mar 10 2021, doi: 10.1038/s41438-021-00494-2.

[37] Q. Zhang, H. Zhang, K. Zhou, and L. Zhang, "Developing a Physiological Signal-Based, Mean Threshold and Decision-Level Fusion Algorithm (PMD) for Emotion Recognition," (in English), Tsinghua Science and Technology, vol. 28, no. 4, pp. 673-685, 2023, doi: 10.26599/tst.2022.9010038.

[38] J. Gao, Q. Lao, Q. Kang, P. Liu, L. Zhang, and K. Li, "Unsupervised Cross-disease Domain Adaptation by Lesion Scale Matching," MICCAI 2022, pp. 660-670, 2022.

[39] J. Gao et al., "Anatomically Guided Cross-Domain Repair and Screening for Ultrasound Fetal Biometry," (in English), IEEE Journal of Biomedical and Health Informatics, vol. , no. , pp. 1-12, Apr 2023, doi: 10.1109/jbhi.2023.3298096.

[40] X. Lai et al., "A disease network-based deep learning approach for characterizing melanoma," Int J Cancer, vol. 150, no. 6, pp. 1029-1044, Mar 15 2022, doi: 10.1002/ijc.33860.

[41] H. Song et al., "Denoising of MR and CT images using cascaded multi-supervision convolutional neural networks with progressive training," (in English), Neurocomputing, vol. 469, pp. 354-365, Jan 16 2022, doi: 10.1016/j.neucom.2020.10.118.

[42] P. Bourke, "iDome: Immersive gaming with the Unity3D game engine," 2nd Annual International Conferences on Computer Games, Multimedia and Allied Technology (CGAT 2009), 2009, doi: 10.5176/978-981-08-3190-5_458.

[43] W. W. Wu, L. W. Lan, and Y. T. Lee, "Exploring decisive factors affecting an organization's SaaS adoption: A case study," International Journal of Information Management, vol. 31, no. 6, pp. 556-563, 2011, doi: "https://doi.org/10.1016/j.ijinfomgt.2011.02.007".

[44] M. Xiao et al., "2019nCoVAS: developing the web service for epidemic transmission prediction, genome analysis, and psychological stress assessment for 2019-nCoV," IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 18, no. 4, pp. 1250-1261, 2021.

[45] Z. Mayer, J. Kahn, Y. Hou, M. Götz, R. Volk, and F. Schultmann, "Deep learning approaches to building rooftop thermal bridge detection from aerial images," Automation in Construction, vol. 146, p. 104690, 2023/02/01/ 2023, doi: 10.1016/j.autcon.2022.104690.

[46] M. Zollhöfer et al., "State of the Art on 3D Reconstruction with RGB-D Cameras," Computer Graphics Forum, vol. 37, no. 2, pp. 625-652, 2018, doi: 10.1111/cgf.13386.

[47] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF," Communications of the ACM, vol. 65, no. 1, pp. 99-106, 2021, doi: 10.1145/3503250.