Fusion CNN-Transformer Model for Target Counting in Complex Scenarios

Xingyuan He[1], Ruiying Wang[2*], Ting Cao[2], Weiyu Liang[3], Yimin Fan[4]
[1]Information Management Center, Shijiazhuang Institute of Railway Technology, Shijiazhuang 050001, China
[2]Finance Department, Shijiazhuang Institute of Railway Technology, Shijiazhuang 050001, China
[3]Information Engineering Department, Shijiazhuang Institute of Railway Technology, Shijiazhuang 050001, China
[4]Economic Management Department, Shijiazhuang Institute of Railway Technology, Shijiazhuang 050001, China
E-mail: hexingyuan1232022@126.com, wangruiying2005@126.com, caoting20095522@126.com, liangweiyu0314@163.com, fym2006@126.com
*Corresponding author

Keywords: convolutional neural network, attention mechanism, computer counting, target counting, fully self attention network

Received: October 12, 2024

To overcome the shortcomings of traditional manual counting methods, which are labor-intensive, resource-consuming, and inefficient, this study introduces a computer-based counting model. This model integrates convolutional neural networks (CNNs) with Transformer networks to efficiently recognize and count specific target objects in large-scale data scenarios. This approach leverages CNNs for local feature extraction and incorporates Transformer networks to capture long-range global information, achieving a synergistic effect. The methodology includes key steps such as "CNN for feature extraction and Transformer for global attention." The experiment outcomes show that the model has an average absolute error of 10.13, a root mean square error of 12.08, an average counting accuracy of 98.6%, a peak signal-to-noise ratio of 23.75, a structural similarity of 0.933, a coefficient of determination of 0.901, an average counting time of about 6.58ms per image, and a parameter count of 3.21 in target counting. It can also recognize and respond well to high complexity scenes while maintaining high accuracy. Compared to the CNN model, the research model reduces the error rate by 13.4%, indicating that the fusion of CNN and Transformer networks is effective in object counting for computer vision tasks. This result indicates that the model integrating convolutional neural networks and fully self attention networks can be well applied to computer recognition and object counting.

Povzetek: Predstavljen je hibridni model CNN-Transformer za štetje tarč v kompleksnih scenarijih. Model združuje CNN za ekstrakcijo lokalnih značilnosti in transformer za zajemanje globalnih informacij.

1 Introduction

Traditional counting relies on manual operation, with low processing power and efficiency, and often requires a lot of manpower and time to identify large-scale data [1-3]. However, as computer technology advances, in recent years, many researchers have begun to rely on computer vision technology to handle the matter of object detection and identification counting in the context of big data. At present, the application of computer counting has spread to many fields, such as road vehicle recognition and counting in vehicle transportation systems, melon and fruit counting in large-scale agricultural and forestry production, livestock counting, and colony counting in laboratories, etc. [4-5]. With the advancement of computer vision technology, an increasing number of computers counting algorithms and models have been developed and applied. Leong J M et al. developed a fish counting system based on convolutional neural network (CNN) to assist hatchery staff in counting fish from images. During the process, contrast limited adaptive histogram equalization was also used to enhance the captured images, and a YOLOv5 form of deep learning architecture was incorporated to generate a model that can recognize and compute fish on the images. The experimental results showed that the recall rate of the model reached 65.5% [6]. Chen G et al. proposed a new efficient deep learning model called Density Transformer for automatically counting trees from aerial images. This architecture includes a multi-receptive field CNN for extracting visual feature representations from local patches and their extensive contexts, and a Transformer encoder for transmitting contextual information between relevant locations. The experimental results showed that the research model achieved the highest accuracy on both datasets, significantly better than most other methods [7]. Miao Z et al. proposed a weakly-supervised method that effectively combines multi-level dilated convolution and Transformer methods to achieve end-to-end crowd counting. The experimental results showed that on four well-known benchmark population counting datasets, this method outperformed other weakly supervised methods and was comparable to fully supervised methods [8]. Liu et al. proposed a multi-receptive field extraction deep learning method grounded on YOLOX

(MRF-YOLO) for detecting and counting small targets, and validated it on the cotton bolls dataset of a cotton farm. The results indicated that the average accuracy of the model rose by 14.86%, with a mean square error of 1.06 and a coefficient of determination of 0.92. The model could be well applied to a wide range of small target crop detection [9]. Shen L et al. constructed a YOLOv5s cluster detection model grounded on channel pruning algorithm and applied it to counting grape clusters in the field. The research results showed that the mAP reached 82.3%, the average inference time per image was 6.1 ms, the average counting accuracy was 84.9%, the video processing speed was 50.4 frames per second, and the model parameters and complexness were effectively reduced while guaranteeing perception precision. This model could be well applied to counting stacked grape clusters [10].

Despite the notable achievements of the aforementioned studies in their respective application scenarios, the field of computer counting still faces several challenges and limitations. In particular, mainstream models like YOLO frequently produce false positives and negatives when confronted with small, densely packed targets, largely attributed to their limited capacity in managing complex scenes and dealing with target occlusion. Furthermore, many existing counting models struggle to balance local and global feature information. Local features are crucial for accurately identifying individual targets, while global features aid in understanding the entire scene and the distribution of targets. However, existing models often fail to achieve a balance between the two, resulting in insufficient flexibility and accuracy during counting.

In response to these limitations, this study proposes a computer counting algorithm that integrates CNN and Transformer networks. This algorithm aims to combine the advantages of CNNs in local feature extraction with the capabilities of Transformers in global feature capture and sequence modeling, thereby enhancing the accuracy and flexibility of computer counting. By introducing the Transformer module, it is hoped to enhance the model's understanding of global contextual information while leveraging the convolutional operations of CNNs to precisely capture the local features of targets. This fusion strategy is expected to address the shortcomings of existing models when dealing with small and densely packed targets, while also improving the counting performance of the model in complex scenarios.

2 Methods and materials

2.1 Counting algorithm integrating CNN

Computer counting refers to the collection of information through computer vision mechanisms, in order to achieve the effect of calculating or counting quantities. This method is often applied in the area of image processing, such as vehicle counting, crowd counting, cell counting, etc. CNN, as a type of deep learning algorithm, is commonly applied in image recognition in the area of computer vision. It simulates the way neurons in the human brain process information, especially the working mode of the visual cortex, and abstracts and extracts feature layer by layer from input data to achieve automatic processing and recognition simulation of grid structured data such as images [11-12]. These features provide detailed object and element information for subsequent counting tasks. CNN is mainly composed of three parts: convolutional layer, pooling layer (also known as downsampling layer), and fully connected layer. Its structure is represented in Figure 1.

In Figure 1, the first layer performs convolution operation on the input image to get a feature map (FM) with a depth of 3. Then pooling operation is constructed on the obtained FM to get a novel FM. The convolution pooling joint operation will be repeated until an FM with a depth of 5 is obtained. This operation process can extract input data features layer by layer. As the number of convolutional and pooling layers rises, the model's ability to interpret and express data gradually improves. Finally, the obtained latest round of FMs is expanded and connected into vectors by rows, and passed into a fully connected layer. the internal hierarchical structure of CNN is analyzed. Part 1: convolutional layers, as shown in formula (1).

Table 1: Literature review table.

Literature	Method	Major contribution	There are problems
Leong J M et al. [6]	CNN-YOLOv5	Assist the staff of the hatchery in counting fish from the images	The recall rate of the model is not high
Chen G et al. [7]	Deep learning models, a multi receptive field CNN	Can achieve automatic calculation of trees in aerial images	The accuracy value is only slightly higher than the general model
Miao Z et al. [8]	Weak supervision law, Transformer	Effectively combining multilevel expansion convolution and Transformer methods to achieve end-to-end population counting.	The research dataset is limited to the population, and the generalization application of counting methods still needs to be considered
Liu et al [9]	YOLOX（MRF-YOLO）	Design proposes a multi receptive field extraction deep learning method for detecting and counting small targets	Mean square error is relatively high
Shen L et al. [10]	YOLOv5s cluster detection model	constructed a detection model and applied it to the counting of grape clusters in the field.	The average counting accuracy is slightly lower and the inference time is slightly longer

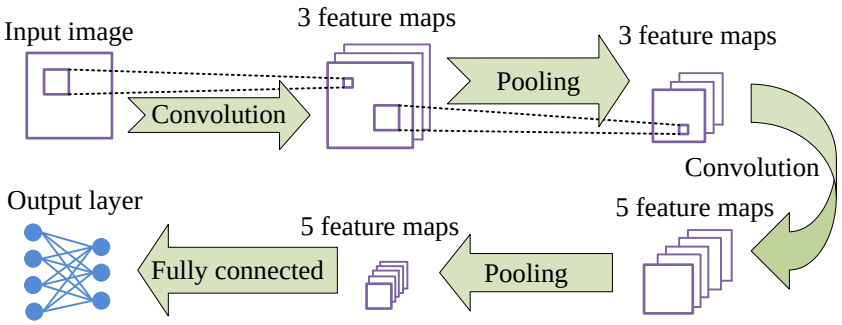


Figure 1: CNN structure diagram.

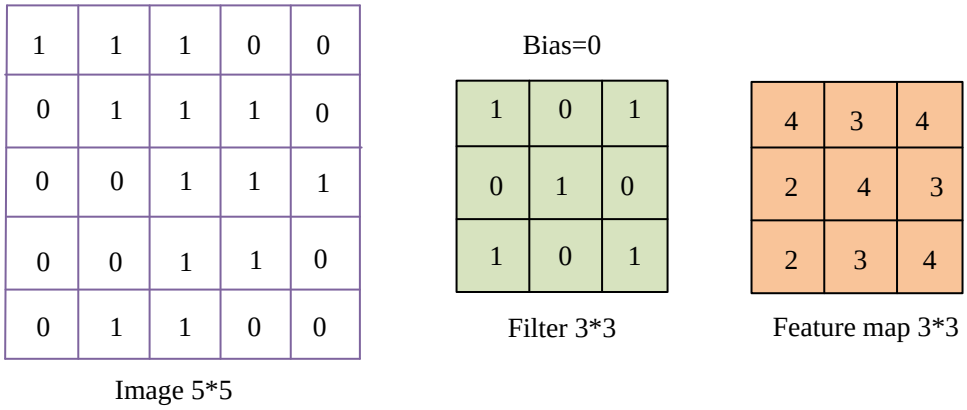


Figure 2: Example of convolution operation.

$$s(i,j)=(X*W)(i,j)=\sum_m\sum_n x(i-m,j-n)w(m,n) \quad (1)$$

$$X'=\frac{(X+2p-W)}{k}+1 \quad (2)$$

Formula (1) represents two-dimensional convolution. Among them, W is the convolution kernel (also known as the weight matrix or filter), X is the input matrix (also known as the input FM), and $s(i,j)$ means the value of the output matrix at position (i,j). $w(m,n)$ means the value of convolution kernel W at position (m,n). $x(i-m,j-n)$ represents the elements of the input matrix X that are accessed in the convolution operation. * Represents convolution. The essence represented by this formula as a whole is to multiply and add the elements at different positions of the matrix and convolution kernel matrix of different parts of the image, as shown in Figure 2.

Figure 2 gives an illustration of convolution process. An image is input and converted into a matrix. In the example, the matrix corresponding to the image is 5×5, and a 3×3 convolution kernel is utilized for convolution to acquire a 3×3 FM. However, not all sliding steps are 1 and need to be adjusted according to the situation. If the sliding stride is greater than 1, there may be a situation where the convolution kernel cannot slide exactly to the edge. In this case, it is necessary to add zeros to the outermost layer of the matrix, as shown in formula (2).

In formula (2), the strid is k and the zero-padding layer is p. The second part is pooling. The pooling layer cuts the dimensionality of FMs while preserving important details through downsampling operations. Pooling can be divided into two types: maximum pooling and average pooling. Compared to max pooling, average pooling can preserve more detailed information. The third part is the fully connected layer, as shown in formula (3).

$$\begin{cases} Y=\varphi(V) \\ V=conv2(W,X,"valid")+b \\ E=\frac{1}{2}\left\|d-y^L\right\|_2^2 \end{cases} \quad (3)$$

In formula (3), *conv* represents the convolution function, *valid* represents the type of convolution operation, b is the bias vector, φ is the activation function, E is the total error, d represents the expected output vector, y means the output node vector, and L means the amount of layers. Figure 3 shows a fully connected diagram.

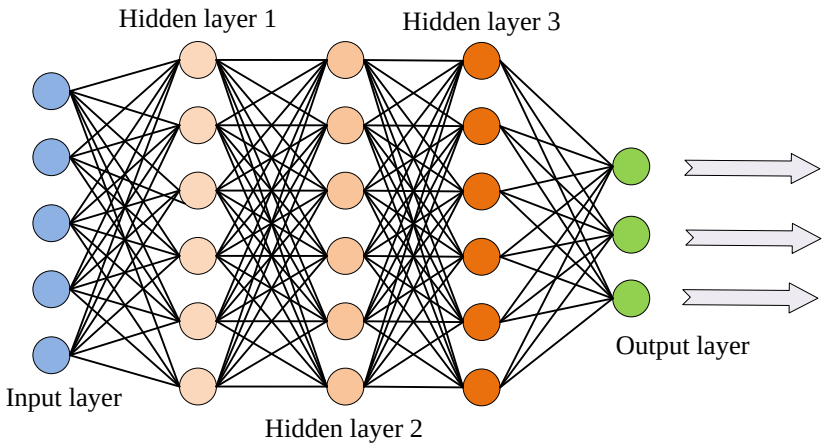


Figure 3: Fully connected layer operation process.

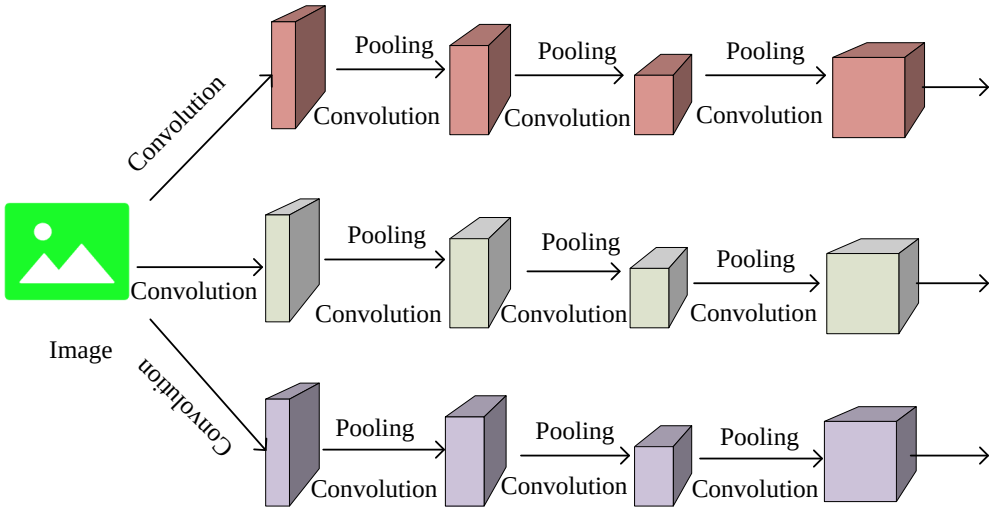


Figure 4: Network structure of FE module.

Figure 3 illustrates the classification function of the fully connected layer, which takes all local detail features as input to the input layer, passes through multiple hidden layers (including linear transformation, nonlinear activation, etc.), and finally generates prediction results through the output layer. However, when CNN is integrated with counting algorithms, it mainly focuses on FE and classification [13]. When the object overlap and coincidence rate of the counted image are high, it is very easy to encounter the problem of varying visual perception depth in comparison with the initial image, which makes it difficult to recognize or misidentify [14]. The counting algorithm that integrates CNN can improve the FE module of the original counting algorithm, helping to enhance the algorithm's ability to capture feature information, as shown in Figure 4.

Figure 4 gives the structure of the FE module that integrates CNN counting algorithm. The FE module includes three parallel CNN networks, with each column's filter (i.e. convolution kernel) having a different size of local receptive field. This produces different feature information extraction effects for counting objects of different distances and sizes, providing higher quality FMs for subsequent network modules and ultimately improving the quality of the algorithm's counting results. In short, integrating the powerful FE capabilities of CNN can effectively enhance computer vision technology and achieve automatic counting of specific objects in images or videos.

2.2 Counting algorithm integrating CNN transformer

Although CNN has strong local FE and parameter sharing capabilities, it can decrease the amount of model parameters and is widely used in image classification and object detection, thereby improving computer vision counting. However, CNN based counting algorithms lack modeling of global information, and CNN assumes that image features have spatial invariance. Therefore, once the target object undergoes deformation or positional changes, it will affect the final counting results [15]. Based on this, the study intends to introduce Transformer on the basis of CNN's counting algorithm. Transformer excels in global information modeling, complementing CNN and Transformer to raise the precision and validity of counting tasks, as represented in Figure 5.

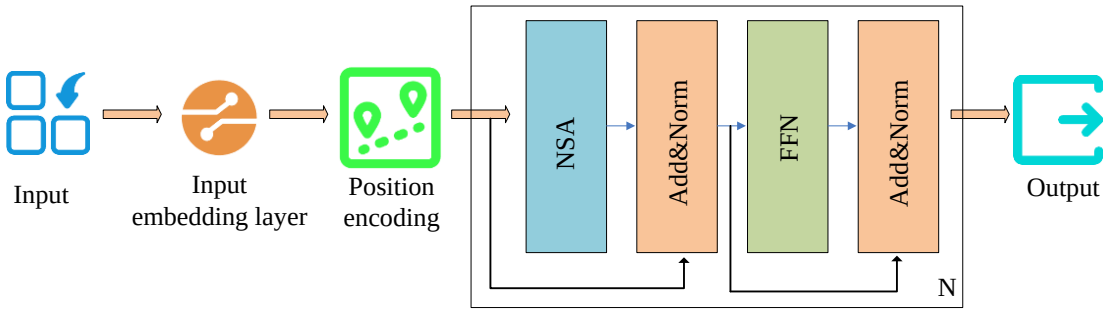


Figure 5: Schematic diagram of transformer structure.

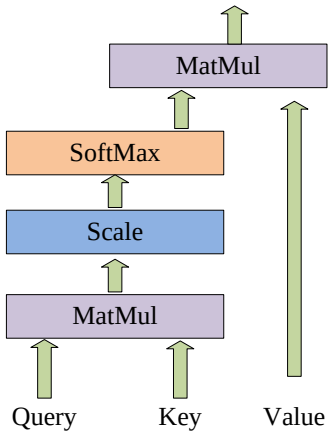


Figure 6: Self attention mechanism calculation process.

From Figure 5, it can be seen that Transformer is mainly composed of Position Embedding, Multi-Head Self Attention (MSA) mechanism, Residual Structure (Add), Normalization (Norm), and FeedForward Network (FFN) [16]. The entire processing flow is to first feed the input data into an input embedding layer composed of transition matrices and convert it into an initial tensor. Then positional encoding information is added to the tensor to generate a new tensor. The new tensor is immediately transmitted to the FE module for further processing. In the FE module, the FE process is repeated N times, each iteration aims at extracting deeper and more abstract characteristics from the input data, ensuring that the model can seize intricate patterns and structures in the data until the optimal result is output. Among them, the position code is shown in formula (4).

$$\begin{cases} PE_{(position,2i)} = \sin\left(\dfrac{position}{10000^{\frac{2i}{d_m}}}\right) \\ PE_{(position,2i+1)} = \cos\left(\dfrac{position}{10000^{\frac{2i}{d_m}}}\right) \end{cases} \tag{4}$$

In formula (4), PE is the position encoding, and the system in formula (4) is the commonly used position encoding, namely sine cosine position encoding. It represents the relative or absolute positional relationship between pixels. The function of position encoding is to enable the model to obtain effective position information. Among them, *position* represents the position of the input element, i means the specific dimension of the element, and d_m represents the dimension of the input. The Transformer model's essential feature is the self-attention mechanism, enabling it to consider all other elements while processing a single element in the sequence, thereby capturing long-range dependencies in the sequence. The computation process is shown in Figure 6.

In Figure 6, it can be seen that *Query*, *Key*, and *Value* are matrices composed of vectors q_i, k_i, and v_i. *Query* and *Key* obtain an output vector sequence containing rich contextual information through matrix multiplication, scaling, SoftMax, and quadratic matrix multiplication, while *Value* directly outputs the sequence through matrix multiplication. The specific first step calculation is shown in formula (5).

$$a_i = Wx_i \tag{5}$$

In formula (5), a_i is the middle tensor, W is the learning matrix, and x_i is the input tensor. Each input tensor is first multiplied by a W matrix and encoded to obtain the intermediate tensor. Multiplying each intermediate tensor with different learning matrices yields the desired vector, as shown in formula (6).

$$\begin{aligned} &q_i = W_q, \quad a_i k_i = W_k, \\ &a_i v_i = W_v a_i, \quad (i = 0,1,2,...,d) \end{aligned} \tag{6}$$

Among them, q_i , k_i , and v_i represent the vectors corresponding to Query, Key, and Value. W_q , W_k , and W_i are corresponding learnable matrices. d is the dimension of the input vector. Among them, each vector q_i will perform attention calculation on each vector k_j (j=0, 1, 2,..., d), that is, perform similarity calculation of vector dot multiplication. Due to the fact that the dot multiplication result increases with the increase of dimension, it is necessary to compress the result and process it through Sofmax, as shown in formula (7).

$$\begin{cases} a_{i,j} = Soft\max(\dfrac{q_i \cdot k_j}{\sqrt{d}}) \\ Soft\max(y_i) = \dfrac{e^{y_i}}{\sum e^{y_i}} \end{cases} \tag{7}$$

In formula (7), $a_{i,j}$ represents the normalized probability value of the vector at position (i, j) corresponding to the Softmax function processing. The Softmax function can convert the output values of multiple classifications into a probability distribution within the range of (0,1) and equal to 1. Finally, multiply the obtained a_{ij} with all vi vectors and sum them to obtain the feature pixels, as shown in formula (8).

$$Attention(Q,K,V) = Soft\max(\frac{QK^T}{\sqrt{d}})V \tag{8}$$

Formula (8) represents the calculation of attention weights in the self attention mechanism. It is worth noting that the current attention mechanism of Transformers usually adopts the Multi Head Self Attention (MSA) mechanism, which is represented as formula (9)

$$\begin{cases} Z_i = Attention(Q_i, K_i, V_i), \quad (i = 1,2,...,h) \\ MultiHead(Q,K,V) = Concat(Z_1, Z_2, ..., Z_h)W_o \end{cases} \tag{9}$$

In formula (9), i represents the i th self attention head, h means the amount of self attention heads, and Z_i means the output matrix calculated by the i th self attention head. Compared with self attention mechanisms, multi-head attention mechanisms can independently and parallelly compute attention in different subspaces, achieving the effect of simultaneously focusing on different features of the input sequence from different perspectives. In addition, in the normalization selection of the model, Transformer adopts layer normalization, as shown in formula (10).

$$LayerNorm(x) = \gamma \cdot (\frac{x-\mu}{\sigma}) + \beta \tag{10}$$

In formula (10), x represents the mean of the input tensor, μ is the standard deviation, γ and β represent learnable parameters, and the size is usually equal to the number of channels. Layer normalization is only applicable to single sample processing and is suitable for handling long sequence data and learning global relationships from single samples. In addition, residual connections are also introduced in the Transformer module, as shown in formula (11).

$$F = Att(X) + X \tag{11}$$

In formula (11), Att represents the attention layer and F represents the output feature. The function of residual connections is to send the data from the last layer to the subsequent layer through skip connections, which simplifies the model's learning process of identity maps, thereby promoting information flow and alleviating the problems of gradient vanishing and exploding [17-18]. In summary, integrating CNN and Transformer networks to construct CNN Transformer counting algorithms can complement each other's strengths and weaknesses, improve computational flexibility, enhance global information modeling capabilities, and improve the accuracy and efficiency of counting tasks. The detailed parameter information of the model is as follows, as shown in Table 2.

3 Results

3.1 Performance analysis based on CNN-transformer counting algorithm model

To verify the capability of the model grounded on the CNN-Transformer counting algorithm, simulation experiments were conducted for validation. Common computer vision applications include counting road vehicles in traffic monitoring systems and counting bacterial colonies in laboratory culture dishes. Considering the difficulty of obtaining the dataset, the study intended to use the actual chicken feeding situation of a large-scale breeding farm in a certain area as the experimental dataset. The selection of live chicken feeding data for this large-scale breeding farm was mainly based on the following considerations: Firstly, this dataset has high practical application value and can provide strong support for precision breeding and animal health management. Secondly, compared to other scenarios, the chicken flock activities in the breeding farm are more intensive and regular, providing rich test samples for counting algorithms. Finally, the dataset exhibits high diversity in terms of image quality, lighting conditions, and background complexity, which helps to comprehensively evaluate the model's generalization ability. A total of 80 live data segments were collected, with a duration of 30-60 seconds per segment, a resolution of 1920 × 1080 pixels, and a frame rate of 25 frames per second. For the collected chicken breeding video data, images were extracted from the video at intervals of 15 frames. In order to improve the quality of

the dataset, manual inspection was used to remove excessively similar or blurry images, and data augmentation was performed on the images in the training set, including random rotation, scaling, cropping, and color transformation. In addition, to ensure the accuracy of annotation, the study adopted cross validation method, where multiple annotators independently annotate the images and ensure the annotation quality through consistency checks. Finally, 761 images were obtained, and the dataset was separated into a training set (685 images) and a testing set (76 images) in a 9:1 ratio. The parameter size was set to: Learning Rate: 0.0005; Optimizer: AdamW; Epochs: 100; Batch Size: 32. The flowchart of data processing is shown in Figure 7.

Table 2: model parameters.

CNN			
Image size	Convolutional kernel size	Number of convolution kernels	Step size and filling
224×224×3	3x3	64	1
Transformer			
Embedding dimension	Position encoding	Hidden layer dimension	Encoder layers
768	Sine/Cosine Position Encoding	2048	6

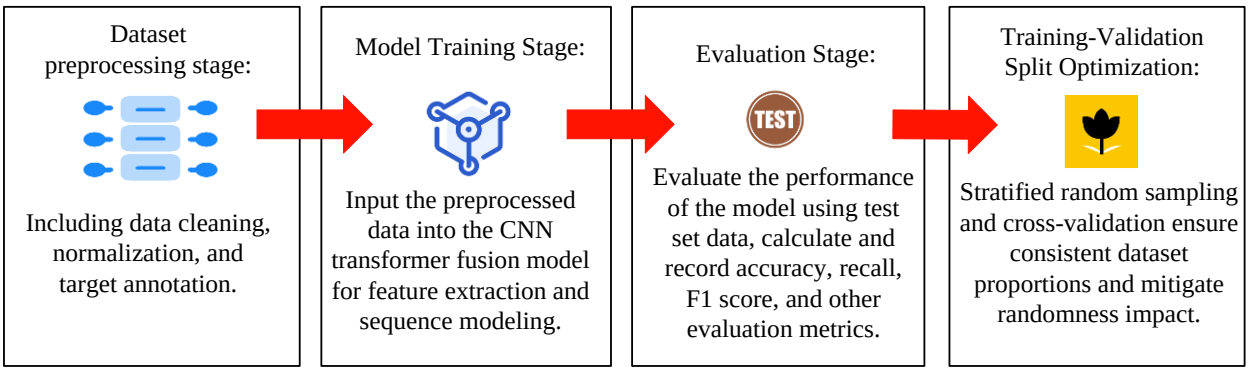
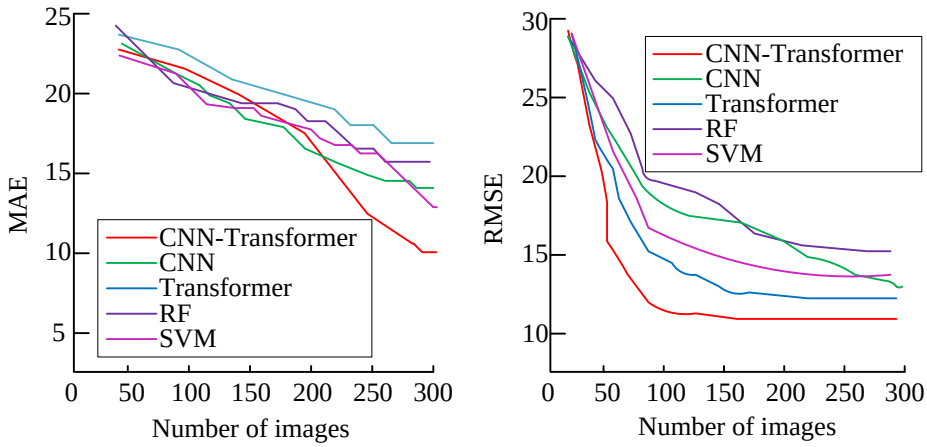


Figure 7: The flowchart of data processing.



(a) Performance indicators MAE of each model

(b) Performance of each model on the RMSE performance metric

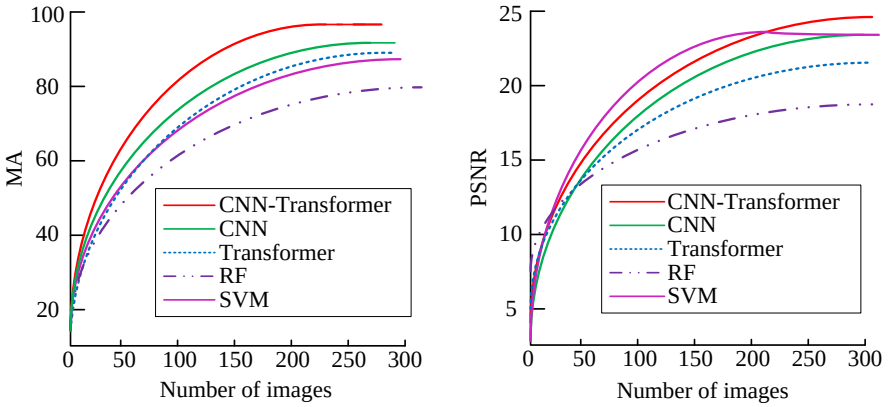
Figure 8: Performance of different algorithms on MAE and RMSE of the training set.

Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Accuracy (MA), Peak Signal to Noise Ratio (PSNR), Structural Similarity (SSIM), and Coefficient of Determination (R2) were used as evaluation metrics for model performance. MAE measures the average of the absolute differences between the predicted and actual values. In counting tasks, MAE provides a straightforward reflection of the accuracy of the model's predictions. RMSE assigns higher weights to larger errors, in counting tasks, it highlights significant deviations in predictions. PSNR in counting tasks, it can be used to measure the similarity between the reconstructed count image and the actual count image. A higher PSNR value indicates better quality of the reconstructed count image and its closeness to the actual image. To more intuitively testify the superiority of the CNN Transformer counting algorithm model, four counting algorithm models including CNN, Transformer, Support Vector Machine (SVM), and Random Forest (RF) were included as comparative algorithms. The comparison results of MAE and RMSE performance of

different algorithms in the training set are shown in Figure 8.

In Figure 8, (a) shows the ability of each model on the behaviour metric MAE. MAE is one of the key indicators for model evaluation, which calculates the mean absolute deviation between predicted and actual values, and is used to characterize the count of network models. The smaller the value, the better the performance. From Figure 8 (a), the MAE value of the CNN-Transformer fusion counting algorithm was 10.13, which was the lowest compared to the other four counting algorithms. Figure 8 (b) shows the behaviour of each model on the performance metric RMSE. RMSE was another important indicator for model evaluation, which was the average square root error between the predicted and actual values. It was used to characterize the stability of network model counting, and the smaller its value, the better the stability of the model. The Transformer model had the highest value of 17.8. From Figure 8 (b), the RMSE value of the CNN-Transformer fusion counting algorithm was 12.08, which was the lowest compared to the other four counting algorithms. The RF model had the highest value of 16.7. The comparison results of MA and PSNR performance of different algorithms in the training set are shown in Figure 9.

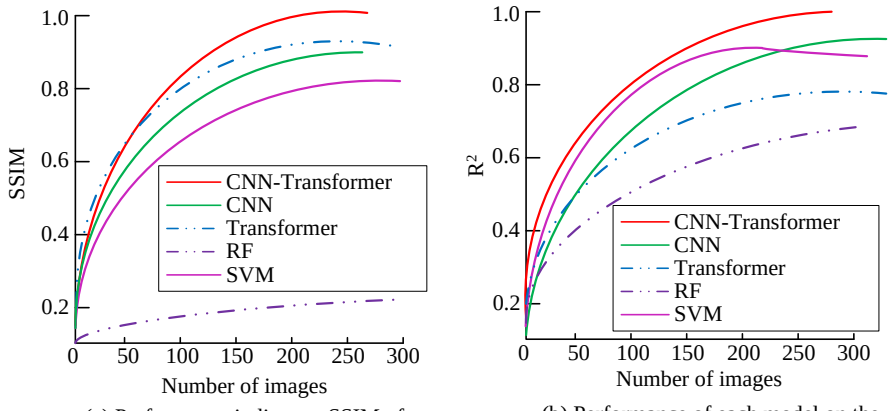
In Figure 9, (a) shows the behaviour of each model on the behaviour metric MA. The larger the MA, the higher the counting accuracy and stability of the network model. From Figure 9 (a), the MA value of the CNN-Transformer fusion counting algorithm was 98.6%, which was the highest compared to the other four counting algorithms. Figure 9 (b) shows the behaviour of each model on the behaviour metric PSNR. This indicator represents the quality of an image based on the error between corresponding pixels, so the higher the PSNR value, the higher the quality of the predicted generated image. In Figure 9 (b), the PSNR value of the CNN-Transformer fusion counting algorithm was the highest, at 23.75. Compared with the other four counting algorithms, this algorithm performed the best in image quality assessment. The comparison results of SSIM performance of different algorithms in the training set are shown in Figure 10.



(a) Performance indicators MA of each model

(b) Performance of each model on the PSNR performance metric

Figure 9: Performance of different algorithms on the MA and PSNR of the training set.



(a) Performance indicators SSIM of each model

(b) Performance of each model on the R^2 performance metric

Figure 10: Performance of different algorithms on SSIM and R2 in the training set.

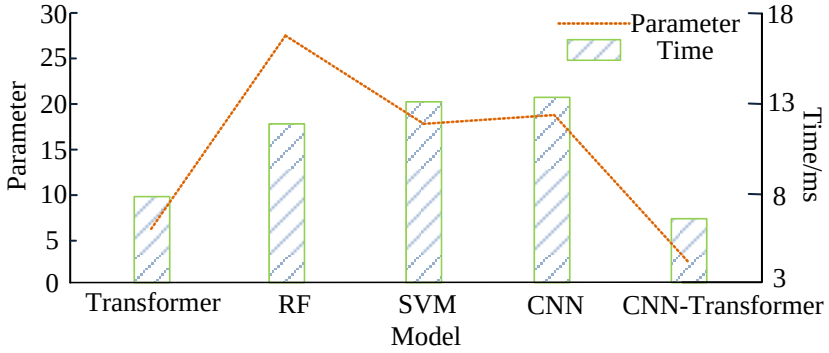


Figure 11: The counting time and parameter count of each algorithm model.

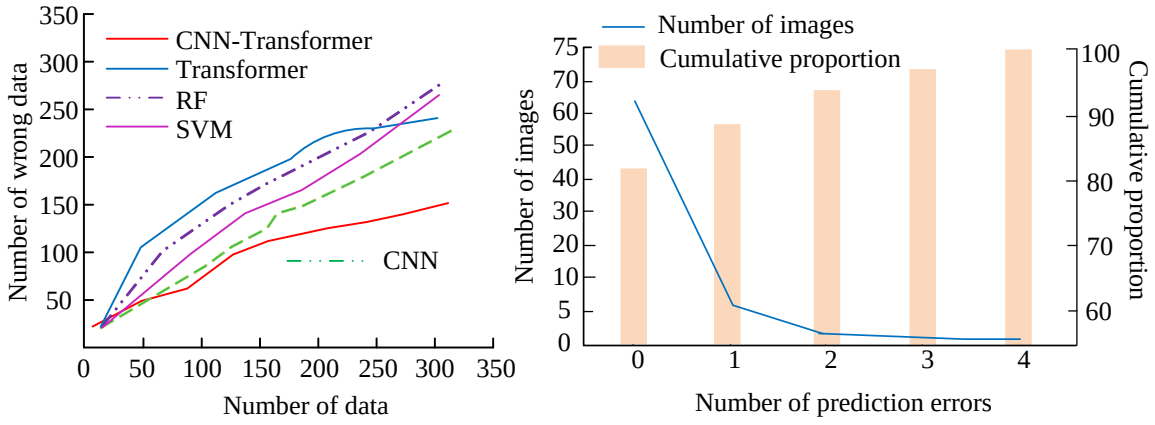
In Figure 10, (a) shows the specific situation of the training sets of five computer counting algorithms on SSIM. This indicator often considers the brightness, contrast, and structure of the image comprehensively to achieve the effect of measuring the correlation between pixels, making it closer to human subjective perception of image quality. Generally speaking, the closer the SSIM value is to 1, the higher the image quality predicted by the algorithm. From Figure 10, it is told that the SSIM value of the CNN-Transformer fusion counting algorithm was 0.933, which was closest to 1 compared to other models. In addition, compared with the other four algorithms, the convergence speed of the research algorithm was significantly higher in the SSIM image, with the convergence inflection point located around image number 40. Figure 10 (b) shows the specific situation of R2 for each model, which reflects the degree of fit of the model. From the figure, it is told that the R2 value of the CNN Transformer fusion counting algorithm was 0.901, which was closest to 1 compared to other models. Based on the above, the proposed counting algorithm that integrates CNN Transformer had good counting performance on the training set. Furthermore, to demonstrate the universality of the model application, the experiment also explored it on a publicly available dataset. This dataset is the Distribution Transformer Detection Dataset (DTD). The same performance indicators as mentioned above were selected for testing. The experimental results showed that MAE was 10.02, RMSE was 12.02, MA was 97.6%, PSNR was 23.55, SSIM was 0.934, and R2 was 0.911.

3.2 Testing and analysis based on CNN transformer counting algorithm model

In the above experiment, the proposed CNN-Transformer counting algorithm model performed well on the training set. To formalize more about the practical application ability of the model, the study intended to use a test set to analyze the model again. Among them, the study compared the recognition performance of various models by introducing the average detection time/ms and parameter quantity of a single image, as shown in Figure 11.

Figure 11 shows the specific situation of the five models in terms of time and parameters. The counting algorithm model that integrated CNN-Transformer had the shortest average counting time for a single image, about 6.58ms, and the smallest number of parameters, about 3.21. In comparison with the model with the longest average detection time for a single image, there was a difference of 6.62ms. Compared with the model corresponding to the maximum parameter count, there was a difference of 24.33. Obviously, the model proposed in the study had shorter recognition and counting time, and more efficient counting efficiency in actual counting. The above indicators reflected the overall testing performance of each model. To understand the situation of each model in counting error images, the study also tested the error counting probability of each model in the test set, recorded the image numbers of error counts in each counting algorithm, and summarized the number of times each image was counted incorrectly. The results are shown in Figure 12.

In Figure 12, (a) shows the false detection rates of different algorithms, and (b) shows the distribution of error count images. From Figure 12 (a), as the number of counting images increased, the error rates of each algorithm randomly increased. However, compared to the other four algorithms, the counting algorithm that integrated CNN-Transformer had a lower overall false detection rate. In Figure 12 (b), out of 76 test set images, 62 images were correctly counted by all models, accounting for 81.58% of the total; The number of images with an error count of less than or equal to 1 accounted for 88.15% of the entire test set. Among the five models mentioned above, there were a total of three images with a classification error rate higher than 50%. One of them was incorrectly counted by four models, indicating that this image had strong confusion and the category features might not be clear enough. The specific number of this image in the test set was 13, with 4 errors. The specific situation of the error probability of this image in the five models is represented in Table 3.



(a) False detection rates of different algorithms (b) Distribution of Error Counting Images

Figure 12: Error recognition status of each model.

Table 3: Probability of incorrect counting for figure 13 by each model.

Image number	Model	Predicted probability
13	CNN	[0.78,0.16]
	Transformer	[0.97,0.56]
	SVM	[0.93,0.64]
	RF	[0.92,0.18]
	CNN-Transformer	[0.59,0.51]

Table 3 shows the error count probabilities of each algorithm for high ambiguity image number 13. The true label of image 13 was a positive sample. From the figure, the intervals of the five counting algorithms in the two-dimensional vector were [0.77, 0.21], [0.96, 0.55], [0.92, 0.63], [0.91, 0.17], and [0.60, 0.30]. The first element in this interval was the probability of incorrectly judging a positive sample, and the second element was the probability of correctly judging a positive sample. Except for the CNN-Transformer model, all other models made incorrect judgments. Subsequently, after separate analysis, it was found that the high error rate of image number 13 was due to issues with lighting and occlusion. The CNN Transformer model combines the advantages of CNN and Transformer, using CNN to extract local features and Transformer to capture global contextual information, thus improving the model's ability to process blurry images. Overall, the counting algorithm that integrated CNN-Transformer still had good recognition and counting capabilities in high complexity scenarios.

4 Discussion

The fusion CNN-Transformer counting algorithm proposed in the study performed well in various performance analysis indicators of the training set data, with MAE of 10.13, RMSE of 12.08, MA of 98.6%, PSNR value of 23.75, and SSIM and coefficient of determination close to 1. In comparison with other algorithms, the algorithm raised in the study performed excellently in all indicators. In addition, in the test set, the experiment also compared the average single image counting time and parameter count of five counting algorithms. It was found that the CNN Transformer counting algorithm had the shortest average single image counting time of about 6.58ms, with a parameter count of 3.21 and the lowest quantity. In terms of error counting, all algorithms showed a trend where the more recognized images, the higher the false detection rate. However, for a single algorithm, the counting algorithm that integrated CNN-Transformer exhibited a lower overall false detection rate. In addition, in low feature and high ambiguity images, except for the counting algorithm that integrated CNN-Transformer, all other algorithms had incorrect recognition counts, indicating that the counting algorithm that integrated CNN and Transformer still had good counting ability in recognizing high complexity counting scenes.

The CNN Transformer model exhibited significant advantages in balancing the number of parameters, inference time, and model accuracy. In resource constrained environments such as farms and other practical application scenarios, traditional complex models often struggle to run stably due to the lack of powerful computing and storage capabilities of the devices in these scenarios. The research model, due to its limited number of parameters and fast inference speed, can adapt well to these resource constrained environments. Therefore, in practical applications, this model can accurately count the number of chickens and provide timely and accurate data support for farm managers. This helps them better understand the feeding situation, develop scientific feeding plans, and thus improve feeding efficiency and economic benefits. Meanwhile, due to the fast inference speed of the model, it can also meet the real-time requirements and provide real-time data feedback for farm managers.

In the same type of research, Zhang L et al. proposed a shrimp automatic local image-based enumerating way

utilizing lightweight YOLOv4, and constructed a local shrimp enumerating model grounded on Light-YOLOv4. The strategy underwent testing on a shrimp dataset, and the results showed that the Light-YOLOv4 local shrimp enumerating model acquired a enumerating accuracy of 92.12%, a recall rate of 94.21%, an F1 value of 93.15%, and an average accuracy mean of 93.16% [19]. Although the comprehensive counting ability of this model was superior, its average accuracy was lower than that of the model in this study. Wu Fy et al. fused the CNN Deeplab V3+model with traditional image processing algorithms and applied it to the detection and counting of banana bunches. The results showed that the final bundle perception precision was 86%, the accuracy of bacterial colony detection during harvesting was 76%, and the overall bacterial colony counting accuracy was 93.2% [20]. The results of this model were lower than the comprehensive behaviour of the model in this study.

The results of this study have significant advantages over existing technology, which may be attributed to the ability of CNN to handle local features and the modeling of global dependencies by Transformer. CNN can effectively extract local features of images, while Transformer captures global dependencies in images through its self attention mechanism. The combination of the two enables more accurate counting when dealing with complex scenes. However, this fusion also brings certain complexity, such as an increase in the number of parameters. However, this research model achieved fast inference time while maintaining a low number of parameters, indicating a good balance between complexity and efficiency.

5 Conclusion

Traditional counting relies on manual operation, with low processing power and efficiency, and often requires a lot of manpower and time to identify large-scale data. However, with the prosperity of Internet technique, computer vision technique can effectively solve this problem for object detection and counting. CNN and Transformer are representative models of deep learning. The former has good local FE ability, while the latter has a non cyclic structure based on attention mechanism and processes the entire input sequence in parallel. Based on this, the study integrated CNN with Transformer to construct a CNN-Transformer model, and explored its performance in target counting through simulation training and testing. The results showed that the model performed well in performance analysis. In testing analysis, the counting time and parameter count of the model were significantly lower than other models of the same type. However, it still performed well in low feature and high confusion image counting recognition. Although the research achieved good results, there were still some limitations, such as the lack of clear input-output mapping in the Transformer model compared to other models, which increased the difficulty of internal interpretation. In the future, efforts can be made to incorporate interpretable artificial intelligence technologies such as attention visualization or salinity maps to enhance the interpretability of models. In addition, the chicken breeding image dataset used in the study still has insufficient quantity in the context of deep learning. In the future, data augmentation techniques such as rotation, scaling, cropping, and flipping can be further adopted to increase data diversity and help models learn more robust features, thereby improving their generality.

References

[1] Muhammad Asif Khan, Hamid Menouar, and Ridha Hamila. Revisiting crowd counting: State-of-the-art, trends, and future perspectives. Image and Vision Computing, 129(1):104597, 2023. https://doi.org/10.1016/j.imavis.2022.104597

[2] Wim Bernasco, Evelien M. Hoeben, Dennis Koelma, Lasse Suonperä Liebst, Josephine Thomas, Joska Appelman, Cees G. M. Snoek, and Marie Rosenkrantz Lindegaard. Promise into practice: Application of computer vision in empirical research on social distancing. Sociological Methods & Research, 52(3):1239-1287, 2023. https://doi.org/10.1177/00491241221099554

[3] N Krishnachaithanya, Gurdit Singh, Smita Sharma, Rangisetti Dinesh, Sumeet Ramsingh Sihag, Kamna Solanki, Abhishek Agarwal, Mrinalini Rana, and Ujjwal Makkar. People counting in public spaces using deep learning-based object detection and tracking techniques. 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), 21(1):784-788, 2023. https://doi.org/10.1109/CISES58720.2023.10183503

[4] Li Zhang, Leilei Yan, Mengqian Zhang, and Jingang Lu. T2 CNN: a novel method for crowd counting via two-task convolutional neural network. The Visual Computer, 39(1):73-85, 2023. https://doi.org/10.1007/s00371-021-02313-0

[5] Shashi Bhushan Jha, and Radu F. Babiceanu. Deep CNN-based visual defect detection: Survey of current literature. Computers in Industry, 148(1):103911, 2023. https://doi.org/10.1016/j.compind.2023.103911

[6] Leong J M, Hijazi M H A, Saudi A, On C K, Fui C F, Haviluddin H. The development and usability test of an automated fish counting system based on CNN and contrast limited histogram equalization. Bulletin of Electrical Engineering and Informatics, 13(2):1128-1137, 2024. https://doi.org/10.11591/eei.v13i2.5840

[7] Chen G, Shang Y. Transformer for tree counting in aerial images. Remote Sensing,14(3):476 2022. https://doi.org/10.3390/rs14030476.

[8] Miao Z, Zhang Y, Peng Y, Peng H, Yin B. DTCC: Multi-level dilated convolution with transformer for weakly-supervised crowd counting. Computational Visual Media, 9(4): 859-873, 2023. https://doi.org/10.1007/s41095-022-0313-5

[9] Qianhui Liu, Yan Zhang, and Gongping Yang. Small unopened cotton boll counting by detection with MRF-YOLO in the wild. Computers and Electronics in Agriculture, 204(1):107576, 2023. https://doi.org/10.1016/j.compag.2022.107576

[10] Lei Shen, Jinya Su, Runtian He, Lijie Song, Rong Huang, Yulin Fang, Yuyang Song, and Baofeng Su. Real-time tracking and counting of grape clusters in the field based on channel pruning with YOLOv5s. Computers and Electronics in Agriculture, 206(1):107662, 2023. https://doi.org/10.1016/j.compag.2023.107662

[11] Yao Liu, Hongbin Pu, and Da-Wen Sun. Efficient extraction of deep image features using convolutional neural network (CNN) for applications in detecting and analysing complex food matrices. Trends in Food Science & Technology, 113:193-204, 2021. https://doi.org/10.1016/j.tifs.2021.04.042

[12] Jinzhu Lu, Lijuan Tan, and Huanyu Jiang. Review on convolutional neural network (CNN) applied to plant leaf disease classification. Agriculture, 11(8):707, 2021. https://doi.org/10.3390/agriculture11080707

[13] Xiang Chen, Hao Li, Mingqiang Li, and Jinshan Pan. Learning a sparse transformer network for effective image deraining. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 46(1):5896-5905, 2023. https://doi.org/10.48550/arXiv.2303.11950

[14] Guy Farjon, Liu Huijun, and Yael Edan. Deep-learning-based counting methods, datasets, and applications in agriculture: A review. Precision Agriculture, 24(5):1683-1711, 2023. https://doi.org/10.1007/s11119-023-10034-8

[15] Nourhan T.A. Abdelnaiem, Hossam M.A. Fahmy, and Anar A. Hady. DC-PHD: multitarget counting and tracking using binary proximity sensors. International Journal of Wireless and Mobile Computing, 16(1):44-59, 2022. https://doi.org/10.1504/IJWMC.2023.135383

[16] Xin Man, Deqiang Ouyang, Xiangpeng Li, Jingkuan Song, and Jie Shao. Scenario-aware recurrent transformer for goal-directed video captioning. ACM Transactions on Multimedia Computing Communications and Applications, 35(1):11079-11091, 2022. https://doi.org/10.1145/3503927

[17] Matteo Polsinelli, Luigi Cinque, and Giuseppe Placidi. A light CNN for detecting COVID-19 from CT scans of the chest. Pattern Recognition Letters, 140(1):95-100, 2020. https://doi.org/10.1016/j.patrec.2020.10.001

[18] Diksha Moolchandani, Anshul Kumar, and Smruti R. Sarangi. Accelerating CNN inference on ASICs: A survey. Journal of Systems Architecture, 113(1):101887, 2021. https://doi.org/10.1016/j.sysarc.2020.101887

[19] Lu Zhang, Xinhui Zhou, Beibei Li, Hongxu Zhang, and Qingling Duan. Automatic shrimp counting method using local images and lightweight YOLOv4. Biosystems Engineering, 220(1):39-54, 2022. https://doi.org/10.1016/j.biosystemseng.2022.05.011

[20] Fengyun Wu, Zhou Yang, Xingkang Mo, Zihao Wu, Wei Tang, Jieli Duan, and Xiangjun Zou. Detection and counting of banana bunches by integrating deep learning and classic image-processing algorithms. Computers and Electronics in Agriculture, 209(1):107827, 2023. https://doi.org/10.1016/j.compag.2023.107827