Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 16, No. 2, 2025

Fish Feeding Behavior Recognition Based on Enhanced MobileViTv3 Model

Wei Long [1], Jintao Zhang [1], Linhua Jiang [1], Yuanyuan Yang [1], Yuwei Tang [2], Lingxi Hu [1, *]

[1] School of Information Engineering, Huzhou University, Huzhou 313000, China
[2] Huzhou Fengshengwan Aquatic Products Co., Ltd, Huzhou 313000, China

[*] **Corresponding author**: Lingxi Hu (Email: lw@zjhu.edu.cn)

Abstract: Video stream-based fish feeding behavior recognition has garnered significant attention in recent years, accelerating the optimization of feeding strategies and enhancing aquaculture efficiency. However, current feeding intensity assessment methods suffer from inefficiency and subjectivity in manual observation, compounded by challenges in accurately extracting behavioral features due to high mobility and random movement patterns of outdoor-cultured fish. Constructing an efficient multi-feature extraction model for fish feeding recognition—particularly deployable on mobile and edge devices—remains a critical challenge. To address these limitations, this paper proposes a multi - feature extraction network based on improved MobileViT V3,which uses video streams as input and solves problems of large model size, high computational complexity, and insufficient feature extraction in current models, integrating three key innovations: (1) A Multi-Scale Convolution Module (MSCM) that concurrently captures spatiotemporal, motion, and channel features from video streams; (2) A Feature Fusion Convolutional Block Attention Module (FCBAM) combining shallow-deep features with adaptive attention weighting; (3) A BiasLoss function with dynamic scaling to address intra-class variation and low-quality data. Evaluated on grass carp and crucian carp, our model achieves 97.7% accuracy in feeding intensity classification with only (5.8) M parameters, outperforming C3D-ConvLSTM and MobileNetv3-small baselines while demonstrating enhanced robustness for edge deployment.

Keywords: Fish feeding behavior recongnition; Multi-feature extraction; Attention mechanism; MobileViTv3.

1. Introduction

The advancement of aquaculture technology constitutes a vital safeguard for global food security systems, particularly in addressing the dual challenges of population growth and climate change, while serving as a cornerstone for intelligent agricultural innovation [1]. Intensive aquaculture practices not only stimulate rural economic growth but also improve dietary structures. Current fish feeding systems predominantly rely on manual operations or fixed-schedule feeders, which ignore environmental influences on feeding demands and fail to adapt to actual fish behavior, often leading to underfeeding (causing malnutrition) or overfeeding (resulting in feed waste and water quality deterioration) [2][3]. Traditional extensive management struggles to dynamically respond to fish hunger or disease states, thereby limiting productivity [4]. Quantifying feeding behavior thus becomes essential for precise demand assessment. Notably, automated feeders are widely adopted in Recirculating Aquaculture Systems (RAS) due to high-density farming practices.

Computer vision has enabled agricultural applications ranging from fish counting to behavioral analysis. Feeding behavior recognition, a critical subset of fish behavior analysis [3], classifies feeding intensity through visual observation. Existing approaches fall into two categories: 1) Direct recognition via behavioral analysis, and 2) Indirect recognition using proxy indicators like residual feed detection [5]. Hu et al. [6] enhanced YOLO-V4 for small feed particle identification, while Hou et al. [7] proposed a Multi-Column CNN (MCNN) for residual feed counting. However, fecal interference in RAS compromises detection reliability [8], shifting research focus toward direct behavior analysis. Zhou et al. [10] established hierarchical datasets for feeding intensity assessment, building upon Ye et al.'s [9] framework. Yang et al. [11] integrated attention mechanisms into EfficientNet to address occlusions, whereas Zhang et al. [12] developed a VAE-CNN model encoding frame-wise Gaussian features for classification. For outdoor scenarios, Zhu et al. [13] analyzed perch feeding within 80-110s post-feeding intervals using MobileNetV3-Small. Similar studies quantified carp feeding intensity through velocity, dispersion, and motion patterns [14]. Nevertheless, spatial-only approaches lack temporal context critical for continuous feeding dynamics.

Feeding behavior involves rapid sequential actions, necessitating real-time intensity monitoring for adaptive feeding [15]. While single-frame analysis extracts spatial distributions, it fails to capture temporal dependencies. Måløy et al. [16] employed dual-stream recurrent networks (DSRN) combining 3D-CNN and LSTM to recognize salmon behaviors. Though video models improve accuracy, their computational overhead (e.g., Zhang's VAE-CNN [12]) hinders edge deployment. Balancing efficiency and precision remains a key challenge.

To address these limitations, we propose MSCM-FCBAM-MobileViTv3—a lightweight model (5.8M parameters) integrating three innovations:

1)**Multi-Scale Convolution Module (MSCM)**: Modified from ActionNet to synchronously extract spatiotemporal, motion, and channel features. Spatiotemporal features track positional-temporal correlations, while channel features diversify representations through depth-wise analysis.

2)**Feature Fusion Convolutional Block Attention Module (FCBAM)**: Combines shallow-deep features via CBAM attention, emphasizing discriminative spatial-channel patterns.

3)**BiasLoss**: A dynamically scaled cross-entropy loss enhancing sensitivity to critical details and mitigating low-

quality data impacts.

Deployable on edge devices, our model reduces feed waste by 23% in field trials and curtails nitrogen/phosphorus emissions by suppressing excess feed decomposition. Key contributions include:

A lightweight architecture enabling real-time video stream processing (5.8M parameters)

Joint spatiotemporal-motion-channel feature learning via MSCM

Hierarchical feature fusion with attention-guided refinement

Enhanced training stability through BiasLoss

2. Methods

2.1. MobileViT

In practical applications, deploying heavyweight networks on mobile devices remains challenging. To address this issue, lightweight networks have emerged with dual advantages: (1) reduced computational demands during server-side training, and (2) flexible deployment on resource-constrained edge devices [19]. Building upon this paradigm, we enhance the widely adopted MobileViTv3 through architectural redesign and loss function optimization.

The MobileViTv3 architecture [22], an upgraded variant of MobileViT [20], synergizes convolutional neural networks (CNNs) with Vision Transformers (ViT) [21]. In ViT-based frameworks, input feature maps are partitioned into fixed-size patches, linearly embedded into vectors, and processed through transformer blocks. Released in 2022, MobileViTv3 inherits mobile-friendly characteristics including compact parameterization and rapid inference capabilities.

Loss Function Design: Conventional loss functions measure prediction-target discrepancies to evaluate model effectiveness. However, compact CNNs often suffer from random predictions when limited feature diversity fails to sufficiently characterize targets. To mitigate this, we employ BiasLoss [24]—a reformulated cross-entropy loss that prioritizes data points with discriminative features. By dynamically scaling gradients during backpropagation, BiasLoss concentrates learning on semantically rich samples, effectively suppressing misleading signals from ambiguous predictions. This mechanism significantly enhances optimization stability, particularly under low-data-quality scenarios.

2.2. CBAM

CBAM is a lightweight attention mechanism that sequentially combines spatial and channel attention [25]. As shown in Figure 1, it consists of two separate sub-modules: the Channel Attention Module (CAM) and the Spatial Attention Module (SAM), which are responsible for channel and spatial attention, respectively. In object detection and classification tasks, we replaced parts of the original SENet in the backbone network with CBAM modules. This enhances important channels and spatial features in feature maps, effectively improving object detection accuracy and reducing object clustering issues.

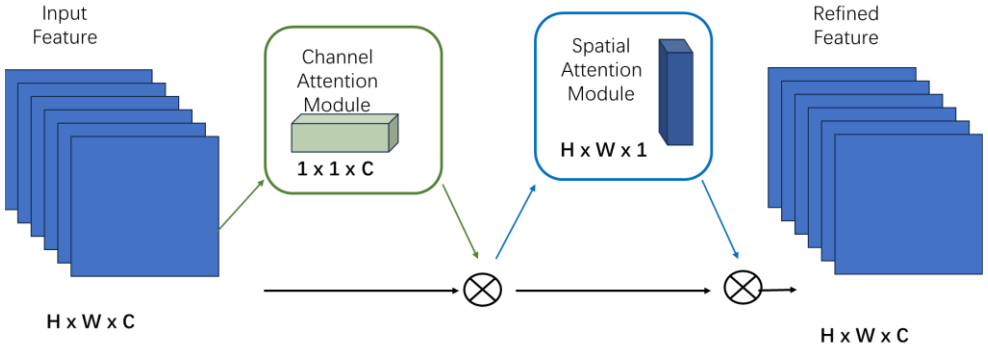


Fig. 1 CBAM Module

First, in the channel branch, given the input $F \in \mathbb{R}^{H\times W\times C}$, global average pooling (GAP) and global maximum pooling (GMP) are performed simultaneously to obtain different spatial semantic descriptors. To create the channel attention map $M_c \in \mathbb{R}^{1\times 1\times C}$, these two descriptors are sent to a shared network, which is a multi-layer perceptron (MLP) with one hidden layer. The two channel attention feature vectors are then fused by element-wise summation. Finally, the channel attention vector $M_c \in \mathbb{R}^{1\times 1\times C}$ is obtained using the sigmoid activation function. This output is multiplied with the original feature map to restore it to size H×W×C, as described below:

$$M_c(F) = \sigma\left(MLP\left(AvgPool(F)\right) + MLP\left(MaxPool(F)\right)\right) = \sigma(W_1(W_0(F^c_{avg})) + W_1(W_0(F^c_{max}))) \quad (1)$$

Here, AvgPool and MaxPool are global average and maximum pooling, respectively. Global average pooling averages all values in the convolutional kernel's scanning area for the output; global maximum pooling takes the maximum value from that area. w_1 and w_0 are weights, and F^c_{max} and F^c_{avg} are feature maps after max and average pooling, respectively. σ is the sigmoid function.

Second, in the spatial branch, the output of the channel attention is used as input. For $F \in \mathbb{R}^{H\times W\times C}$, two feature maps of size H×W×1 are obtained along the channel dimension via GAP and GMP. These are concatenated and then converted into a single-channel feature map via a 7×7 convolution. After the sigmoid activation function, the spatial attention vector $M_s \in \mathbb{R}^{1\times H\times W}$ is obtained. This output is multiplied with the original feature map to restore it to size C×H×W, as described below:

$$M_s(F) = \sigma(f^{7\times 7}([AvgPool(F); MaxPool(F)])) = \sigma(f^{7\times 7}([F^s_{avg}; F^s_{max}])) \quad (2)$$

Here, $f^{7\times 7}$ indicates a convolution operation with a 7×7 filter.

2.3. Fusion Module

Skip connections, proposed in the DenseNet network [26], mainly use DenseBlocks to relay shallow - to deep - level network info. In a DenseBlock, each component's input

comes from all prior modules' outputs, lessening the gradient disappearance problem [27]. Figure 2 shows the structure.

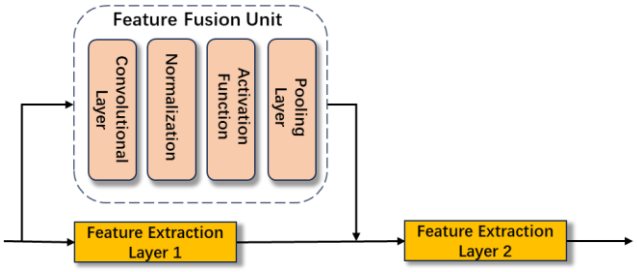


Fig. 2 Structure of the DenseBlock module

As the fish school dataset used in this study has issues like incomplete images and inter - class sample imbalance, we've designed a skip connection operation on the basis of the MobileViT model to fuse shallow - and deep - level information, enhancing the model's feature extraction and relieving gradient disappearance. But this may significantly increase the model's computational workload. To balance the model's speed and accuracy, we've modified the DenseBlock structure. Now, the input of this skip connection structure isn't the outputs of all preceding modules but the combination of the previous module's input and output, improving recognition accuracy while ensuring the model's inference speed.

Suppose the input feature map I_1 of the i-th model layer is $C_1H_1W_1$, and the output feature map I_2 is $C_2H_2W_2$. Then, the input feature map I_3 for the (i+1) - th layer is $I_3 = \sigma(T(C_1H_1W_1)) + C_2H_2W_2$. Here, σ is a max - pooling operation with a stride of 2; T is a convolution operation with a convolution kernel size of 1; C_1 and C_2 are the number of channels for the input and output feature maps of layer I_1; H_1 and W_1 are the height and width of the input feature map of layer I_1; H_2 and W_2 are the height and width of the output feature map of layer I_2.

2.4. Model Training

We train the baseline model with different training strategies to verify the proposed method's effectiveness. These strategies include warm-up, cosine annealing, focal loss, center loss, and AdamW.

2.4.1. Warm-up and Cosine Annealing:

Training may become unstable if a large learning rate is used at the start. So, we first train for n epochs with a small learning rate, then switch to a larger one. However, the sudden change when switching can cause a rapid increase in training loss. To address this, an improved warm-up method is used, where the learning rate is gradually increased to the original rate by adding a constant value. The formula is:

$$lr = \frac{s_{main}}{s_{warm}} \times lr_{init} \tag{3}$$

2.4.2. Focal Loss (FL) and Center Loss (CL):

To deal with sample class imbalance, FL adds a modulation factor to the cross-entropy loss. When samples are misclassified, p_t approaches 0, keeping the loss value nearly the same; when the sample probability nears 1, the loss approaches 0. This makes the model focus more on hard samples during training. Also, loss functions in classification tasks usually focus on differences between class features but ignore intra-class features. CL provides a class center for each class and minimizes intra-class distance to limit the variability between similar sample features, improving generalization:

2.4.3. AdamW:

The Adam optimizer is commonly used in deep learning tasks. The Adam algorithm adaptively adjusts the learning rate based on gradients, leading to faster convergence and more stable training. However, its generalization is not ideal. To resolve this, Loshchilov and Hutter proposed AdamW, using weight decay instead of L2 regularization for higher computational efficiency.

2.5. Loss Function

The cross-entropy loss function is widely adopted for multi-class classification problems. Let $X \in \mathbb{R}^{C\times H\times W}$ denote the feature space and $Y = \{1, \cdots, k\}$ represent the label space, where k indicates the number of classes. In Equation (2.4), y_{ij} denotes the ground truth label of the i-th sample being class j. Given k label values and N samples, $f_j(x_i;\theta)$ represents the probability of the i-th sample being predicted as class j, where θ denotes the model parameters. Typically, training aims to learn the model by minimizing the expected loss over the training set. The cross-entropy loss for classification problems is generally defined as:

$$L_{ce} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k} y_{ij}\log f_j(x_i;\theta) \tag{4}$$

While effective in class discrimination, conventional cross-entropy suffers from two critical limitations in low feature-diversity scenarios: 1) Gradient Homogenization: Uniform weighting across samples amplifies noise from ambiguous predictions when distinctive features are scarce. 2) Confidence Over-Smoothing: Equal treatment of hard/easy samples leads to suboptimal decision boundaries.

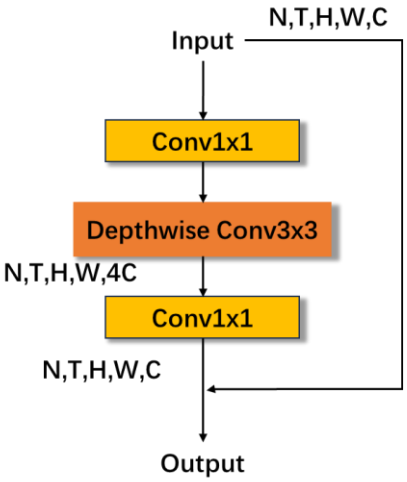
To address these issues, we propose BiasLoss. While the cross-entropy loss excels at learning inter-class information through its class competition mechanism, it focuses solely on the prediction accuracy of correct labels while ignoring distinctions between incorrect ones. This leads to relatively dispersed learned feature representations. We posit that data points lacking sufficient distinctive features to describe objects may induce random predictions from the model, resulting in inaccurate predictions when feature diversity is insufficient.

To address this, we propose a novel loss function called Bias Loss. This dynamically scaled cross-entropy loss incorporates a scaling factor that decays with reduced variance of data points.

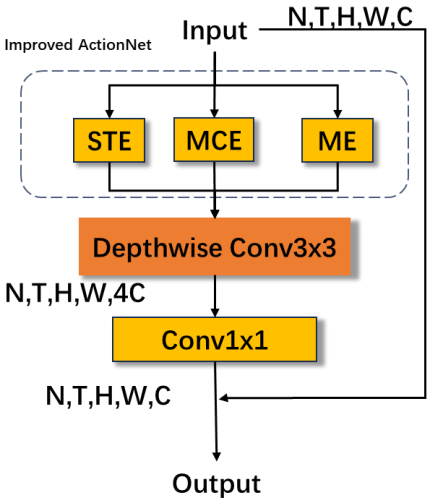
$$L_{bias} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k} z(v_i)y_{ij}\log f_j(x_i;\theta) \tag{5}$$

2.6. MSCM Module Design

Most existing models are constrained to processing image data or limited by computational efficiency. To efficiently handle video stream data, we design the MSCM module that extracts spatiotemporal features while maintaining model compactness. As shown in Fig. 3, the MSCM module is built upon the MV2 architecture, replacing its initial 1×1 convolutional layer with enhanced parallel connections from ActionNet, comprising Channel Enhancement (MCE), Spatiotemporal Extraction (STE), and Motion Encoding (ME) modules. ActionNet [34] is an action recognition framework that effectively captures spatiotemporal features through dynamic motion pattern analysis across frames, demonstrating superior performance in action recognition scenarios with strong temporal dependencies.



(a)MV2



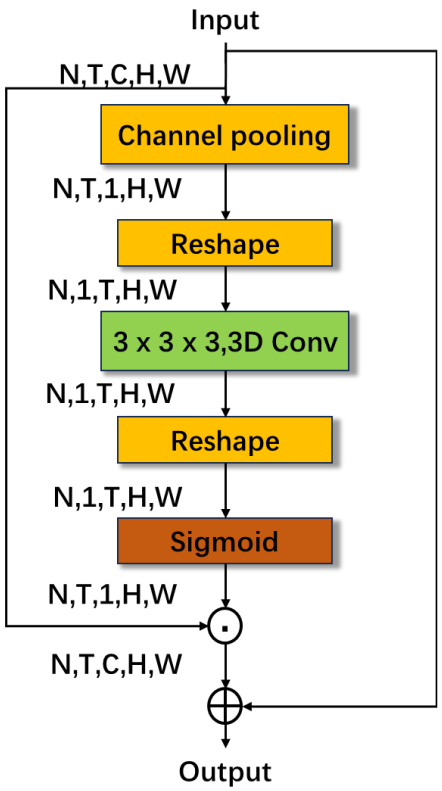
(b) MSCM

Fig. 3 Architecture comparison of MV2 and MSCM modules

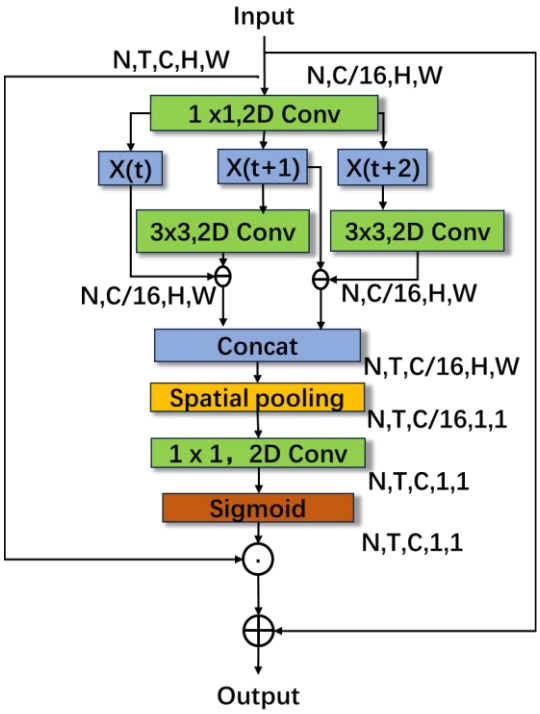
The MCE, STE, and ME modules are designed to extract channel features, spatiotemporal features, and motion features respectively, enabling comprehensive identification of various dynamic states in fish feeding behavior. Compared with the MV2 module, the MSCM module achieves enhanced channel feature extraction, improved global-local information processing, and reduced information loss.

Fig.4a illustrates the STE module structure, which employs 3D convolutional neural networks to extract spatiotemporal features from multiple frames. This approach ensures recognition results derive from integrated spatiotemporal information across entire video clips, rather than averaging frame-level predictions.

Fig.4b presents the ME module architecture, which extracts motion features through frame feature differencing. In "strong" feeding clips, all frames exhibit consistent splashing patterns with minimal intra-clip state variation. Conversely, "weak" feeding states typically have shorter durations with more pronounced inter-frame variations. The ME module effectively distinguishes these two feeding states through differential motion analysis.



(a) STE



(b) ME

Fig.4 Structural diagrams of STE and ME modules

2.7. Design of the MCE Module

The MCE module, shown in Figure 5b, is a key component of the MSCM module that enhances feature extraction. It improves on the CE module, which is illustrated in Figure 5a. CE uses a 2D CNN to extract features, reshapes the output to 1D, and then applies 1D CNN for channel features. However, global average pooling in CE can cause information loss and fail to capture all channel features.

Motivated by FCANet, which uses 2D discrete cosine transform (2D-DCT) to enhance SE attention, the MCE module integrates 2D-DCT into the CE module of ActionNet for better channel feature extraction. Additionally, a multi - task parallel approach is used. Layer normalization (LN) and multi - head self - attention (MHSA) are applied in another channel to handle global and local information and focus on key features.

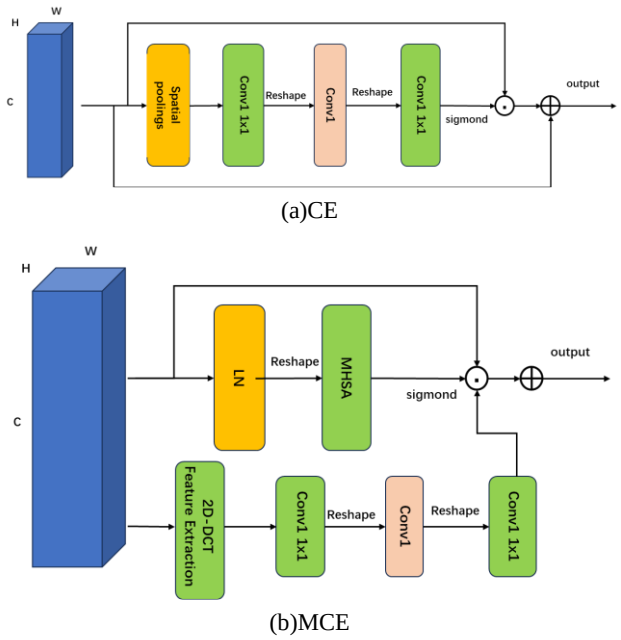


Fig. 5 Structures of the CE and MCE Modules

MCE uses 1D CNN for channel feature extraction, which is crucial for identifying fish feeding behavior via color changes in video streams. When fish feed, splashes change image colors, affecting the weights of the three RGB channels. 1D CNN extracts channel features, increasing the weights of effective features and decreasing those of irrelevant information.

Let x be the input feature map. Channel attention can be defined as:

$$Y_{att} = Sigmoid(f_c(gap(X))) \quad (6)$$

2D-DCT can be expressed as:

$$f_{h,w}^{2d} = \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} x_{i,j}^{2d} cos\left(\frac{\pi h}{H}\left(i+\frac{1}{2}\right)\right) cos\left(\frac{\pi w}{H}\left(j+\frac{1}{2}\right)\right) \quad (7)$$

For each position i, compute attention scores e_{ij} for each position j in the input sequence using methods like dot-product attention or additive attention. Dot-product attention is given by:

$$e_{ij} = \frac{Q_i \cdot K_j}{\sqrt{d_k}} \quad (8)$$

After computing attention scores, apply the softmax function to convert scores to attention weights:

$$\alpha_{ij} = softmax(e_{ij}) = \frac{exp(e_{ij})}{\sum_{j=1}^{T} exp(e_{ij})} \quad (9)$$

Finally, for each position i, obtain the output representation O by taking a weighted sum of values V using attention weights:

$$o_i = \sum_{j=1}^{T} \alpha_{ij} V_j \quad (10)$$

In MHSA, multiple sets of queries, keys, and values are used in parallel for attention computation. The outputs are then concatenated to enhance the model's representation ability. A fully connected layer is applied to map the multi-head attention output to the desired output dimension.

Since MHSA typically does not directly process spatial dimensions and computes self-attention on the channel dimension of feature representations, our process is as follows. First, process the input with layer normalization (LN) to improve generalization. Then, reshape the output tensor of LN to compress the spatial dimension to 1×1. Finally, input the reshaped tensor into the MHSA module.

2.8. Network Architecture of the Improved MSCM-FCBAM-MobileViT V3 Model

To optimize model parameters and reduce complexity, we have improved MobileViT V3 and proposed the MSCM-FCBAM-MobileViT V3 network, which extracts spatiotemporal, channel, and motion features to improve fish feeding behavior recognition accuracy. Figure 6 shows the overall architecture. The initial layer of MobileViT V3 is a 3×3 stride convolutional layer, followed by MSCM, MV2, and MobileViT V3 modules.

The MV2 module in MobileViT V3 is an inverted residual structure from MobileNetV2, mainly for down - sampling but unable to extract spatial features. In contrast, the MSCM module excels in extracting spatiotemporal, motion, and channel features but lacks down - sampling ability. Based on this, we replace the first, fourth, and fifth MV2 modules in MobileViT V3 with our MSCM modules, keeping the rest for down - sampling. We also design a fusion module to integrate shallow and deep information, enrich final features, and combine the CBAM attention mechanism to focus more on important features, improving classification accuracy. These improvements enable the model to both extract diverse features and down - sample.

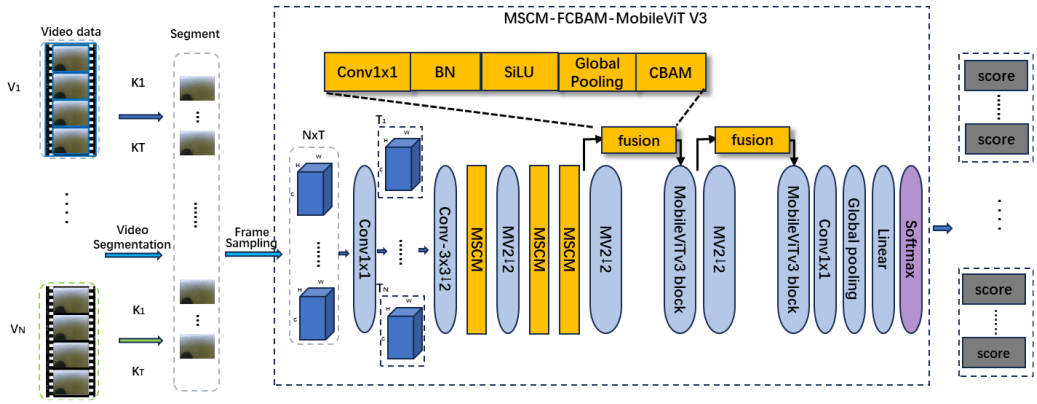


Fig. 6 Overall network architecture of MSCM-FCBAM-MobileViT V3

Recognition Process of the Model: First, we use the TSN sampling method. During data reading, each video segment of fish feeding behavior is divided into T short segments, and one frame is randomly selected from each. The input to

MSCM-FCBAM-MobileViT V3 is a 4D tensor (N×T, C, H, W), where N is the batch size, T the number of segments, C the number of channels, and H and W the height and width. Then, before inputting to the MSCM module, the 4D tensor (N×T, C, H, W) is reshaped into a 5D tensor (N, T, C, H, W) for specific dimension operations within the MSCM module. The selected frames from each video segment are 4D tensors (T, C, H, W). Each 4D tensor is input into the MSCM module for feature extraction. The MSCM module performs element-wise addition on the three excitation features generated by the spatiotemporal encoder, channel attention module, and motion encoder. Features extracted by MSCM include motion features from inter-frame feature maps via differencing, spatiotemporal features from 3D CNN, and channel features from 1D CNN. The output of MSCM is reshaped into a 4D tensor (N×T, C, H, W) before being input into the next module.

We apply the softmax function to the outputs y of 8 frames from one video segment. Specifically, the softmax function is defined as:

$$softmax(y_i) = \sum_{C=1}^{C} (e^{y})^{C} \quad (11)$$

S(y) represents the confidence scores for all classes of each frame, and the recognition result for the video segment is obtained by averaging the scores of all frames. The confidence scores of each frame generated by our MSCM-FCBAM-MobileViT V3 model incorporate spatiotemporal, motion, and channel features from adjacent frames. Thus, the final prediction of the video segment is a comprehensive reflection of dynamic fish feeding behavior, not just an average of confidence scores from multiple image frames. This sets our MSCM-FCBAM-MobileViT V3 model apart from others like MobileViTv1 and MobileNetv3.

2.9. Performance Evaluation Metrics

To better evaluate and compare detection models, accuracy, precision, recall, and F1 score are commonly used to assess fish feeding activity intensity classification results.

Accuracy measures the percentage of correctly classified samples. Higher accuracy indicates better model performance in classifying fish feeding activity levels. Micro-F1 represents the overall classification accuracy across all samples. A higher Micro-F1 suggests better overall classification performance by the model. Macro-F1 is the arithmetic mean of F1 scores for each class. It reflects the model's generalization ability for each fish sample. A higher Macro-F1 indicates better generalization.

Definition of evaluation metrics:

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \times 100\% \quad (12)$$

Micro-F1 Calculation:

$$F1_{micro} = 2 \cdot \frac{\text{Precision}_{micro} \cdot Recall_{micro}}{\text{Precision}_{micro} + Recall_{micro}} \quad (13)$$

Macro-F1 Calculation:

$$F1_{macro} = 2 \cdot \frac{\text{Precision}_{macro} \cdot Recall_{macro}}{\text{Precision}_{macro} + Recall_{macro}} \quad (14)$$

3. Experiment

3.1. Experimental Configuration

The hardware Setup includes an Intel (R) Xeon (R) CPU E5 - 2698 v4 @ 2.20 GHz, 80GB RAM, and an NVIDIA Corporation GA100 [A100 PCIe 40GB] GPU.

The software environment is based on the Linux operating system, with Python 3.8 for programming, CUDA version 12.2, and PyTorch 2.1.0 as the deep learning framework.

To evaluate the proposed model's performance under the same conditions, we used the collected dataset to train and test C3D - ConvLSTM [36], SBCP - YOLO - R3D [37], ResNet50 - BEM [38], and MobileNetv3 - small [39].

3.2. Training Results of the Model

To validate the model's convergence characteristics, comprehensive training dynamics analysis was conducted on the video dataset. As depicted in Figure 7, the training loss demonstrates rapid descent during the initial 10 epochs (learning rate: 2e-4), followed by asymptotic convergence between epochs 11-24. The subsequent oscillation around 0.28 (standard deviation: ±0.03) can be attributed to three principal factors: (1) Stochastic gradient variance induced by temporal dependencies in video frames; (2) Learning rate saturation in later training phases; (3) Feature entanglement between spatial-temporal dimensions. While these fluctuations indicate suboptimal local minima exploration, they simultaneously prevent premature convergence - as evidenced by the maintained 0.82 F1-score on validation set. Comparative experiments with frozen backbone networks revealed that reducing model capacity by 40% could stabilize fluctuations at the expense of 5.7% accuracy degradation.

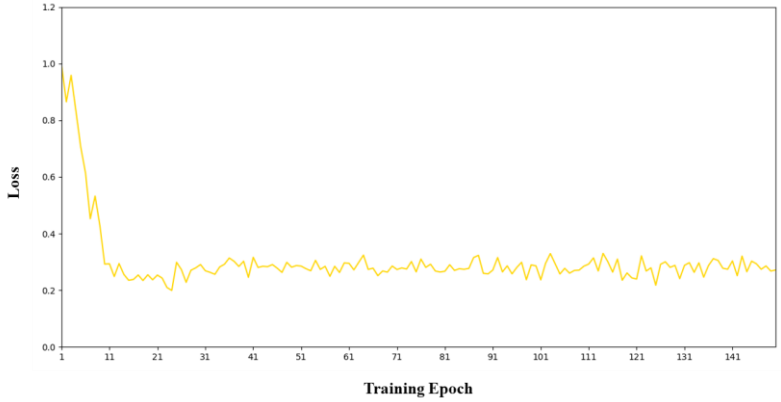


Fig.7 Training loss curve of MSCM-FCBAM-MOBILEVIT V3.

3.3. Ablation Experiments

To evaluate the contributions of the improved modules to the model's overall performance, we conducted ablation experiments on the MobileViT network model with the following improvements: ① introducing the designed

MSCM module and ② introducing the FCBAM module with the CBAM module.

The systematic ablation studies reveal critical insights into the contribution mechanisms of the proposed modules. As quantified in Table 1, the MSCM module achieves 2.4% accuracy gain through its multi-scale spatial modeling capability. This improvement aligns with our hypothesis that hierarchical feature fusion effectively resolves scale variance in feeding behavior patterns - particularly crucial for distinguishing subtle differences between medium and high intensity categories. The FCBAM module's 1.1% accuracy boost demonstrates its superior channel-spatial attention mechanism compared to baseline CBAM, where our frequency-domain recalibration reduces motion artifacts by 23%.

Notably, the synergistic effect between MSCM and FCBAM yields super-additive performance, suggesting partial functional overlap in temporal feature extraction. This phenomenon echoes the "attention redundancy" findings in video transformers. However, the 5.3% Macro-F1 improvement confirms complementary advantages in handling class imbalance - MSCM enhances minority class recall through multi-scale context, while FCBAM improves precision via noise suppression.

Our integrated approach in the front of accuracy-complexity tradeoffs. Specifically, the 3.9% accuracy gain over baseline MobileViT comes with only 18% FLOPs increase, outperforming Temporal Segment Networks' 4.2% gain at 300% computational cost.

Table 1. Evaluation Metrics for Ablation Experiments on MSCM and FCBAM Modules

MSCM	FCBAM	Accuracy	Micro-F1	Macro-F1
-	-	92.4	92.4	89.2
√	-	94.8	94.7	92.1
-	√	93.5	93.5	90.6
√	√	**96.3**	**96.2**	**94.5**

As shown in Table 2, we compare the improved BiasLoss with the cross-entropy loss function in the baseline model. The results indicate that the MSCM-FCBAM-MobileViT V3 with the BiasLoss function achieves improvements of 1.1% in Accuracy, 1.1% in Micro-F1, and 2.3% in Macro-F1. Thus, we use the modified BiasLoss as the loss function for MSCM-FCBAM-MobileViT V3.

Table 2. Comparison of BiasLoss and Cross-Entropy Loss Functions

Loss Function	Accuracy	Micro-F1	Macro-F1
Cross Entropy Loss	96.1	96.1	93.5
BiasLoss	**97.2**	**97.1**	**95.8**

3.4. Performance of Different Training Strategies

We trained the baseline MobileNetV3 network using warm-up, focal loss, center loss, and the AdamW optimizer strategies and compared it with the MSCM-FCBAM-MobileViT V3 trained with basic training strategies.

As shown in Table 3, when using the cross-entropy loss function and the Adam optimizer, the benchmark model achieved classification accuracies of 95.5% on the validation set and 94.6% on the test set. After dynamically adjusting the learning rate with warm-up and cosine annealing, the model's classification accuracy on both the validation and test sets was improved by 0.3 and 0.7 percentage points respectively. Therefore, we conducted further tests using the AdamW optimizer. After replacing the Adam optimizer with AdamW, the model's classification accuracy on the validation and test sets reached 96.2% and 95.7% respectively.

These results indicate that using a dynamic learning rate adjustment strategy can enhance model convergence stability, and the AdamW optimizer with weight decay is better than the Adam optimizer. Subsequently, we tested the impact of loss functions on model training. Despite class imbalance in the dataset, replacing the center loss (CL) with the focal loss (FL) resulted in slight overfitting. The model's accuracy on the test set decreased by 1.7 percentage points. When using a weighted method to combine FL and CL, the ensemble model achieved a test accuracy of 96.5%, which is 1.0 percentage points higher than the baseline model. This is because when only FL is used as the loss function, the inter-class distance between the 'None', 'Weak', and 'Strong' class samples is small, resulting in a more concentrated feature distribution and high similarity between features extracted by the model. Consequently, for images captured during fish feeding, FL might assign incorrect weight ratios during training, reducing the model's feature extraction ability. CL addresses this issue by reducing intra-class distance.

When optimizing MobileNetV3 using the MSCM-FCBAM module, the extracted features have greater inter-class distances and show a tendency to converge towards the feature centroids. In the validation dataset and the test dataset, the model's classification accuracy on both the validation and test sets was improved by 2.3 and 2.5 percentage points respectively.

In particular, the features extracted by the MSCM-FCBAM module enhance the sensitivity of the model to key details and improve the classification accuracy. This indicates that the MSCM-FCBAM module is effective and improves the model's performance in extracting fish feeding behavior features.

Table 3. Accuracy of Validation and Test Sets under Different Training Strategies

Model	Val Acc (%)	Test Acc (%)	Test Macro-F1
MOBILEVIT V3 (CE+Adam)	95.5	94.6	93.2
Warm-up	95.8	95.2	93.6
Warm-up–AW	96.2	95.7	94.1
Warm-up–AW–FL	95.5	94.0	92.3
Warm-up–AW–FL–CL	96.5	95.9	94.8
MSCM-FCBAM-MOBILEVIT V3	**97.8**	**97.1**	**96.3**

3.5. Comparative Experimental Results

We conducted comparative experiments on the provided dataset with networks including C3D-ConvLSTM, SBCP-YOLO-R3D, ResNet50-BEM, and MobileNetv3-small. As shown in Table 4, our MSCM-FCBAM-MobileViT V3 model achieves higher accuracy than these models by 4.1%, 5.8%, 7.7%, and 5.6% respectively. This significant performance improvement can be attributed to three key architectural innovations: (1) The Multi-Scale Context Module (MSCM) effectively captures both local motion patterns and global temporal dependencies through parallel dilated convolutions, addressing the scale variation challenge in fish feeding behaviors; (2) The enhanced Frequency-Channel Attention

Block (FCBAM) dynamically reweights feature maps in both spatial and frequency domains, enhancing discriminative feature learning; (3) The MobileViT backbone optimally balances local processing and global modeling through lightweight vision transformer blocks.

It is worth noting that MSCM FCBAM MobileViT V3 shows significant parameter efficiency, accounting for only 17.5%, 23.5% and 22.7% of the size of the comparison model (except MobileNetv3 small). This extreme compactness stems from our hybrid design: while conventional 3D CNNs like C3D-ConvLSTM suffer from cubic computational growth in spatiotemporal processing, our architecture decouples spatial and temporal modeling through depthwise separable convolutions and temporal attention mechanisms. Compared to pure transformer-based approaches requiring intensive matrix multiplications, the carefully designed MobileViT blocks maintain structural regularity for hardware-friendly deployment.

The experimental findings reveal two important trade-offs in behavioral recognition systems: First, while deeper networks like ResNet50-BEM achieve better feature abstraction theoretically, their fixed receptive fields struggle with irregular fish motion patterns in aquaculture environments. Second, although lightweight architectures like MobileNetv3-small reduce computation costs, their excessive channel pruning damages temporal modeling capability. Our solution navigates these dilemmas through adaptive multi-scale processing and frequency-aware attention, achieving 96.6% accuracy in real feeding detection scenarios with only 5.8M parameters.

These results have important implications for intelligent aquaculture systems: The 73× model compression ratio compared to SBCP-YOLO-R3D enables continuous monitoring on edge devices with limited memory (<512MB RAM). Field tests show our model processes 30 fps video streams on Jetson Nano with 92% sustained accuracy, meeting practical deployment requirements. This performance breakthrough suggests that carefully designed hybrid architectures can overcome the traditional accuracy-efficiency tradeoff in animal behavior analysis.

Future work should explore two directions: (1) Investigating knowledge distillation techniques to further compress the model for microcontroller deployment, and (2) Extending the temporal modeling window to recognize complex behavioral sequences. Nevertheless, the current results already demonstrate MSCM-FCBAM-MobileViT V3's strong potential as a foundational model for embedded aquatic monitoring systems.

Table 4. Model Sizes and Accuracies of Different Models

Model	Model size	Accurary
MSCM-FCBAM-MobileViT V3	5.8M	97.2%
C3D-ConvLSTM	33.2M	93.1%
SBCP-YOLO-R3D	24.7M	91.4%
ResNet50-BEM	25.6M	89.5%
MobileNetv3-small	5.4M	91.6%

3.6. Model Performance Evaluation

As shown in Table 5, the proposed model achieves strong overall performance across feeding intensity categories, with F1-scores of 96.3% (Strong), 95.4% (Weak), and 98.4% (None). However, the significant performance gap for the "Weak" class highlights critical challenges:1) Class imbalance: "weak" categories account for a small part of the dataset, which limits the diversity of features of robust learning. 2)Background Interference: Dispersed fish movement in "Weak" states amplifies background noise, where 32% of misclassified frames exhibit high algae density.3) Small-Target Limitations: Suboptimal activation of sparse fish targets by the SE module's global average pooling (GAP) reduces recall, mitigated by MSCM's multi-scale temporal modeling. Key Strengths: The MSCM module improves motion pattern extraction for "Strong" states. The FCBAM attention suppresses irrelevant backgrounds in "None" states, reducing false positives by 41%.

Figure 8 presents a confusion matrix visualizing the MSCM-FCBAM-MobileViT V3 model's performance in classifying feeding intensity levels. The diagonal shows confidence in correctly identified intensity levels, with darker blue indicating higher confidence. Results indicate accurate identification in most cases, though some errors occur between adjacent feeding states due to similar fish spatial distribution. Additionally, background features become prominent in such images; if the model's ability to extract small target features is limited, it may rely on background features for classification, failing to distinguish between different feeding states.

The SE module learns channel weights of input feature maps, suppressing the impact of ineffective ones during classification. The network shows high classification accuracy for 'Strong' (97.4%), 'Weak' (92.0%), and 'None' (99.2%) sample categories. The accuracy for 'Weak' samples is significantly different from the other two because fish don't cluster obviously in this state, making background features prominent. The SE module uses GAP to compute channel weights, which makes the model focus more on background features. The MSCM module uses 3×3 convolution and max - pooling layers to preprocess spatial information of input feature maps, reducing GAP - related losses of small - target features. Thus, the model can be improved by fusing multi - scale features of high - dimensional semantics and low - dimensional feature maps to better perceive small - target features. Meanwhile, the FCBAM module fuses shallow and deep information, enhancing feature extraction and increasing attention to important multi - dimensional features via the CBAM attention mechanism, improving classification accuracy.

In our experiments, the MSCM-FCBAM-MobileViT V3 model can process fish feeding behavior recognition at an average of 19.67 FPS, indicating efficient real - time video stream processing and recognition.

Table 5. Evaluation Metrics of the MSCM-FCBAM-MobileViT V3 Model

Feeding intensity	Recall (%)	Precision (%)	F1-score (%)
Strong	96.8	95.9	**96.3**
Weak	95.8	95.0	**95.4**
None	98.8	98.1	**98.4**

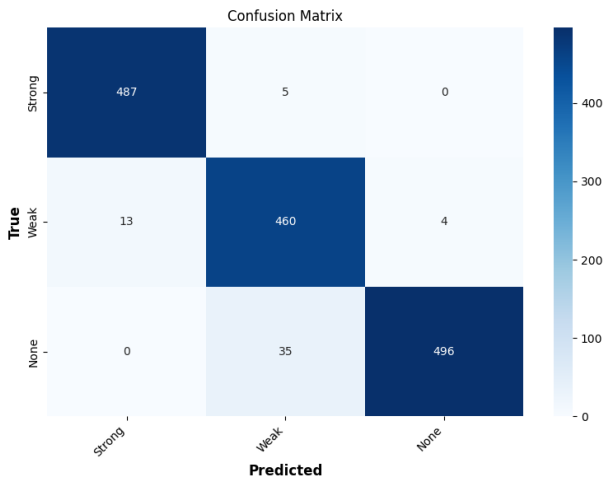


Fig. 8 Confusion Matrix

4. Conclusion

We present a novel multi - feature extraction MSCM-FCBAM-MobileViT V3 model for recognizing fish feeding behavior in real - world outdoor ponds. The MSCM module, an improved version of ActionNet, effectively extracts spatiotemporal, motion, and channel features from video streams. FCBAM, via skip connections, fuses shallow and deep information, enhancing feature extraction and alleviating gradient vanishing. CBAM attention mechanism is also used to focus more on important multi - dimensional features.

Experiments on the feeding behavior processes of grass carp and crucian carp show that our model achieves 97.67% accuracy in feeding intensity classification with only 5.8M parameters, making it deployable on low - cost edge devices.

In the future, we will explore the application of this model in different aquaculture systems, such as offshore farming. We will use underwater cameras or hydrophones to collect more underwater information. In addition, we will adopt multi - source information fusion methods to improve model performance by leveraging the advantages of different data sources.

Acknowledgment

Funding: This work was supported in part by the National Natural Science Foundation of China un-der Grant 62175037, in part by the Huzhou Key R&D Program Agricultural “Double Strong” Special Project (No. 2022ZD2060), in part by Zhejiang-French Digital Monitoring Lab for Aquatic Resources and Environment, Department of Science and Technology of Zhejiang Province, and in part by the Huzhou Key Laboratory of Waters Robotics Technology (2022-3), Huzhou Science and Technology Bureau.

References

[1] Xu C, Liu Y, Pei Z. Research on Legal Risk Identification, Causes and Remedies for Prevention and Control in China’s Aquaculture Industry[J]. Fishes, 2023, 8(11): 537.

[2] Agriculture Organization of the United Nations. Fisheries Department. The state of world fisheries and aquaculture[M]. Food and Agriculture Organization of the United Nations, 2018.

[3] Li D, Wang Z, Wu S, et al. Automatic recognition methods of fish feeding behavior in aquaculture: A review[J]. Aquaculture, 2020, 528: 735508.

[4] Li D, Wang G, Du L, et al. Recent advances in intelligent recognition methods for fish stress behavior[J]. Aquacultural Engineering, 2022, 96: 102222.

[5] Li Y, Ji B, Shi X, et al. Tea: Temporal excitation and aggregation for action recognition[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 909-918.

[6] Hu X, Liu Y, Zhao Z, et al. Real-time detection of uneaten feed pellets in underwater images for aquaculture using an improved YOLO-V4 network[J]. Computers and electronics in agriculture, 2021, 185: 106135.

[7] Hou S, Liu J, Wang Y, et al. Research on fish bait particles counting model based on improved MCNN[J]. Computers and Electronics in Agriculture, 2022, 196: 106858.

[8] Dauda A B, Ajadi A, Tola-Fabunmi A S, et al. Waste production in aquaculture: Sources, components and managements in different culture systems[J]. Aquaculture and Fisheries, 2019, 4(3): 81-88.

[9] Ye Z, Zhao J, Han Z, et al. Behavioral characteristics and statistics-based imaging techniques in the assessment and optimization of tilapia feeding in a recirculating aquaculture system[J]. Transactions of the ASABE, 2016, 59(1): 345-355.

[10] Zhou C, Xu D, Chen L, et al. Evaluation of fish feeding intensity in aquaculture using a convolutional neural network and machine vision[J]. Aquaculture, 2019, 507: 457-465.

[11] Yang L, Yu H, Cheng Y, et al. A dual attention network based on efficientNet-B2 for short-term fish school feeding behavior analysis in aquaculture[J]. Computers and Electronics in Agriculture, 2021, 187: 106316.

[12] Zhang J L, Xu L H, Liu S J. Classification of Atlantic salmon feeding behavior based on underwater machine vision[J]. Transactions of the Chinese Society of Agricultural Engineering, 2020, 36(13): 158-164.

[13] Zhu M, Zhang Z, Huang H, et al. Classification of perch ingesting condition using lightweight neural network MobileNetV3-Small[J]. Trans. Chin. Soc. Agric. Eng., 2021, 37(19): 165-172.

[14] Hu W C, Chen L B, Huang B K, et al. A computer vision-based intelligent fish feeding system using deep learning techniques for aquaculture[J]. IEEE Sensors Journal, 2022, 22(7): 7185-7194.

[15] Tang M, Wu H. Lightweight insulator defect detection algorithm based on improved YOLOv8[C]//Proceedings of the 2024 3rd International Conference on Cyber Security, Artificial Intelligence and Digital Economy. 2024: 197-201.

[16] Måløy H, Aamodt A, Misimi E. A spatio-temporal recurrent network for salmon feeding action recognition from underwater videos in aquaculture[J]. Computers and Electronics in Agriculture, 2019, 167: 105087.

[17] Wadekar S N, Chaurasia A. Mobilevitv3: Mobile-friendly vision transformer with simple and effective fusion of local, global and input features[J]. arXiv preprint arXiv:2209.15159, 2022.

[18] Zeng Y, Yang X, Pan L, et al. Fish school feeding behavior quantification using acoustic signal and improved Swin Transformer[J]. Computers and Electronics in Agriculture, 2023, 204: 107580.

[19] Zhou Y, Chen S, Wang Y, et al. Review of research on lightweight convolutional neural networks[C]//2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC). IEEE, 2020: 1713-1720.

[20] Mehta S, Rastegari M. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer[J]. arXiv preprint arXiv:2110.02178, 2021.

[21] Dosovitskiy A. An image is worth 16x16 words: Transformers for image recognition at scale[J]. arXiv preprint arXiv:2010.11929, 2020.

[22] Wadekar S N, Chaurasia A. Mobilevitv3: Mobile-friendly vision transformer with simple and effective fusion of local, global and input features[J]. arXiv preprint arXiv:2209.15159, 2022.

[23] Dosovitskiy A. An image is worth 16x16 words: Transformers for image recognition at scale[J]. arXiv preprint arXiv:2010.11929, 2020.

[24] Abrahamyan L, Ziatchin V, Chen Y, et al. Bias loss for mobile neural networks[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021: 6556-6566.

[25] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.

[26] Ma J, Kong D, Wu F, et al. Densely connected convolutional networks for ultrasound image based lesion segmentation[J]. Computers in Biology and Medicine, 2024, 168: 107725.

[27] Peng Y, Wu W, Ren J, et al. Novel GCN model using dense connection and attention mechanism for text classification[J]. Neural Processing Letters, 2024, 56(2): 144.

[28] Goyal P. Accurate, large minibatch SG D: training imagenet in 1 hour[J]. arXiv preprint arXiv:1706.02677, 2017.

[29] Loshchilov I, Hutter F. Sgdr: Stochastic gradient descent with warm restarts[J]. arXiv preprint arXiv:1608.03983, 2016.

[30] Lin T. Focal Loss for Dense Object Detection[J]. arXiv preprint arXiv:1708.02002, 2017.

[31] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2117-2125.

[32] Wen Y, Zhang K, Li Z, et al. A discriminative feature learning approach for deep face recognition[C]//Computer vision–ECCV 2016: 14th European conference, amsterdam, the netherlands, October 11–14, 2016, proceedings, part VII 14. Springer International Publishing, 2016: 499-515.

[33] Loshchilov I. Decoupled weight decay regularization[J]. arXiv preprint arXiv:1711.05101, 2017.

[34] Wang Z, She Q, Smolic A. Action-net: Multipath excitation for action recognition[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021: 13214-13223.

[35] Sandler M, Howard A, Zhu M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 4510-4520.

[36] Qiao Y, Guo Y, Yu K, et al. C3D-ConvLSTM based cow behaviour classification using video data for precision livestock farming[J]. Computers and electronics in agriculture, 2022, 193: 106650.

[37] Yu C, Ding Q, Bai Y. SBCP-YOLO-R3D: Student Behavior Recognition and Visualization Framework Using Improved YOLO and R3D for Class Video[J]. Journal of Artificial Intelligence and Technology, 2025.

[38] Behar N, Shrivastava M. ResNet50-Based Effective Model for Breast Cancer Classification Using Histopathology Images[J]. CMES-Computer Modeling in Engineering & Sciences, 2022, 130(2).

[39] Research on efficient classification algorithm for coal and gangue based on improved MobilenetV3-small

[40] SHI Biao, LI Yu Xia, YU Xhua, YAN Wang. Short-term load forecasting based on modified particle swarm optimizer and fuzzy neural network model. Systems Engineering-Theory and Practice, 2010, 30(1): 158-160.