Optimized Model for Texture Image Recognition with Adaptive Trilinear Pooling

Aiwu Chen
College of Intelligent Manufacturing, Hunan University of Science and Engineering, Yongzhou, China
E-mail: aiwuchen@outlook.com

Keywords: Trilinear pooling, model pruning, image texture, YOLOv5, CAE

Received: July 31, 2025

Existing techniques often focus on single aspects such as accuracy or response time, with limited efforts made toward optimizing multiple indicators simultaneously. Therefore, this paper proposes an image recognition model based on trilinear pooling, which combines multiple algorithms and employs model pruning to meet the dual demands of accuracy and efficiency. The experiment was run on an NVIDIA RTX 4090 GPU, using AudioSet's balanced training subset for training. The validation set was the Eval subset, and the test set consisted of 10 common sounds, each with 100 samples per class. Train for 100 epochs using the Adam optimizer. Use Detection Transformer, Cascade Single Shot MultiBox Detector, and Efficient Hybrid Backbone Object Detection as baselines. The results indicate that the proposed model achieved an accuracy of 92.7% at the end of the iteration. Within a 95% confidence level, the confidence intervals for images of bird sounds, car sounds, and natural environment sounds are (0.91, 0.99), (0.88, 1.00), and (0.90, 1.00), respectively. The model's response time ranges from 18.5 ms to 23.1 ms during testing. The average accuracy across 200 tests remains above 90%, and the model consistently outperforms comparison models under varying levels of Gaussian noise and lighting conditions. These results demonstrate that the proposed method effectively improves image recognition performance in terms of both accuracy and efficiency. This work offers a new perspective and practical example for optimizing image recognition, which is expected to support the development of more accurate and efficient intelligent technologies.

Povzetek: Članek predlaga trilinearno združevanje z obrezovanjem modela za slikovno prepoznavo, ki združi več algoritmov in na AudioSet doseže visoko natančnost ob nizki zakasnitvi ter robustnosti na šum in osvetlitev.

1 Introduction

With the development of artificial intelligence, various automation technologies based on image recognition have emerged, which heavily rely on image recognition to make accuracy a key factor in determining the performance of the entire system or device [1]. Although many scholars have conducted in-depth studies on optimizing image processing techniques, most existing methods struggle to address both long response times and low accuracy simultaneously [2]. Therefore, there is an urgent need for an optimized image processing model that improves both accuracy and efficiency. Intelligent image recognition relies on deep learning algorithms to extract and analyze one or more types of data from the texture [3]. Trilinear pooling (TP) is a feature fusion method that utilizes outer product operations among three sets of feature vectors to capture interactions between different features, thereby facilitating a deeper understanding of relationships across various dimensions [4]. Model pruning (MP) is a lightweight technique that enhances computational speed by removing redundant nodes or channels with minimal impact on accuracy [5]. This study introduces an image recognition method based on TP and builds a recognition model by integrating various deep learning algorithms and applying MP. The innovation lies in using a comprehensive analysis of multi-level image textures to replace traditional single-level recognition, thereby improving accuracy. At the same time, the coupling of multiple deep learning algorithms helps enhance recognition efficiency. The goal is to address the limitations of current image processing techniques, including low accuracy and slow performance, and to promote further advancement of image recognition-based technologies.

2 Related works

Due to its higher precision in extracting complex semantic features compared to general fusion methods, TP has been widely used in various research fields. For example, He et al. applied TP to fuse multimodal data, using trajectory and natural language features to support scene modeling and enhance visual performance. Experimental results showed

that this method was effective [6]. Sun et al. proposed a new deep learning architecture for behavior recognition based on visual input, combining posture and appearance feature sequences, and formulated linear, bilinear, and TP methods. The results showed that the area under the subject's working feature curve of the proposed method could reach 0.92 [7]. Lee et al. proposed an intelligent model for body detection using CT images. They treated gender, age, and weight as three distinct features and applied TP to enhance detection accuracy. Testing confirmed the effectiveness of the model [8]. To balance recognition accuracy and efficiency in driver action identification, Hu et al. proposed a spatiotemporal adaptive module based on TP. They trained it using data extracted from a high-capacity spatiotemporal deep learning model, which resulted in a recognition accuracy of 98.7% [9]. Grimm et al. utilized TP to integrate multimodal data from cameras, LiDAR, and radar sensors, thereby enhancing the accuracy of automatic vehicle distance labeling. Experimental results demonstrated that their method improved the feasibility of this technology [10].

As image recognition technology has advanced, both domestic and international researchers have conducted extensive studies, producing relatively mature theories and practical approaches. For instance, Wang et al. proposed a deep image compression scheme to enhance the generalization ability of image recognition. They improved the Bjøntegaard delta by reducing the parameter space of the visual model and combining it [11]. Vasanth et al. developed an image recognition method based on multivariate correlation analysis to improve face recognition accuracy. By extracting geometric feature points and low-level features simultaneously, they achieved an accuracy of over 95% across four datasets [12]. Hassan et al. proposed an optimization algorithm that combines stochastic gradient descent, momentum-based optimization, and deep ensemble strategies to enhance the accuracy of skin disease image recognition. Although their method effectively enhanced accuracy, it lacked analysis of response time during the recognition process [13]. To improve accuracy in foggy image recognition, Sahu et al. proposed a cascaded defogging method based on an adaptively parameterized dual-channel simplified pulse-coupled neural network. Extensive experiments showed that the method outperformed conventional qualitative and quantitative approaches. However, the study did not examine recognition efficiency [14]. Frants et al. addressed foggy image recognition by developing a solution based on a quaternion neural network architecture. By applying a pixel-wise quaternion loss function and a quaternion normalization layer, they significantly improved image quality [15]. The summary table of the above-mentioned related work is shown in Table 1.

Table 1: Summary of related works

Author	Task	Dataset	Metrics	Limitations
He et al. (2024) [6]	Video Visual Relationship Detection	ImageNet-VidVRD	Accuracy	Only focus on precision optimization
Sun et al. (2023) [7]	Gait freeze detection	Freezing of Gait dataset	Area under the curve = 0.92	Unanalyzed computational efficiency
Lee et al. (2022) [8]	CT image analysis	CT dataset	Dice coefficient	Lack of efficiency evaluation
Hu et al. (2024) [9]	Driver behavior recognition	Self-made driving dataset	The accuracy rate is 98.7%	Relying on dedicated hardware platforms
Grimm et al. (2022) [10]	Multimodal fusion of autonomous driving	nuScenes	Estimation accuracy	No specific delay reported
Wang et al. (2022) [11]	Machine vision	Kodak, CLIC	Bjøntegaard delta	Ignore recognition accuracy
Vasanth et al. (2022) [12]	Face recognition	Four publicly available datasets	Accuracy > 95%	Response time not analyzed
Hassan et al. (2023) [13]	Skin disease image recognition	ISIC 2019	Accuracy	Not considering the time cost
Sahu et al. (2022) [14]	Image dehazing and recognition	RESIDE, O-HAZE	Peak Signal-to-Noise Ratio	Unanalyzed computational efficiency
Frants et al. (2023) [15]	Image dehazing	RESIDE, HazeRD	Structural Similarity Index Measure	Not considering the time cost

In summary, existing image recognition methods have achieved certain advancements in theory and technique. However, existing methods generally have two limitations. Firstly, most studies only focus on optimizing a single performance indicator, lacking collaborative optimization of multidimensional objectives. Secondly, there is a general lack of systematic efficiency evaluation and statistical verification. Therefore, this study proposes a balanced optimization scheme that balances accuracy and efficiency by introducing a collaborative design of trilinear pooling and model pruning, thereby filling the gap in existing technology in terms of accuracy, efficiency, balanced optimization, and rigorous evaluation. This study aims to provide new solutions for real-time image recognition applications in resource-constrained scenarios, thereby promoting the practical application of high-performance visual recognition systems.

3 Construction of the image recognition model based on adaptive TP and MP

3.1 Design of adaptive TP-based feature fusion and classification algorithm

Image recognition often focuses on analyzing one or several types of features and completing recognition based on their feature values. To improve the accuracy of image recognition, it is necessary to extract and analyze as much information as possible [16]. TP extracts deep semantic relationships between different types of data through outer product computation. It is suitable for classification tasks involving multi-dimensional data. The spatial attention mechanism enhances specific types of information in feature maps by assigning weights, while filtering or hiding less important information [17]. It adaptively distinguishes texture features in images to prepare them for TP outer product computation. Therefore, this study puts forward an adaptive TP algorithm based on the spatial attention mechanism. The algorithm's structure is illustrated in Figure 1.

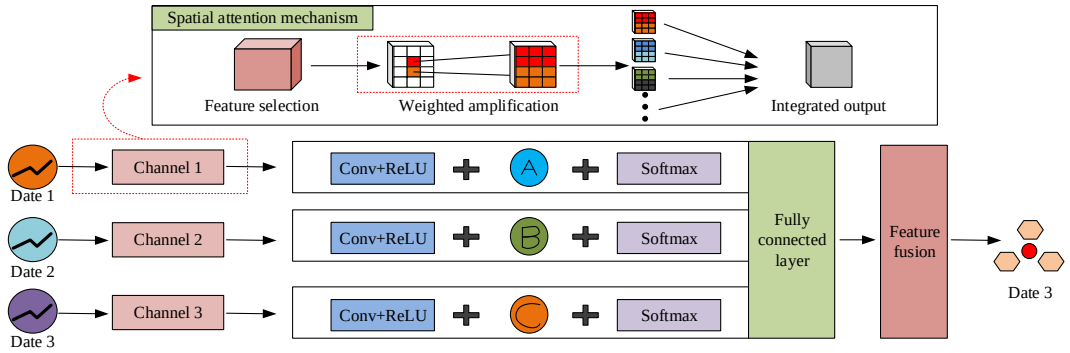


Figure 1: TP-based image feature fusion algorithm with spatial attention (Source from: author self-drawn)

In Figure 1, the spatial attention mechanism is applied before TP. It adaptively enhances the feature map through weighted processing before the outer product computation, as shown in Equation (1).

$$Ms(F') = F \otimes Ms(F) \quad (1)$$

In Equation (1), F represents the feature values, $Ms(F)$ is the feature map, and $Ms(F')$ is the weighted feature map. This study uses average pooling and max pooling for weighting, as shown in Equation (2).

$$Mc(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \quad (2)$$

In Equation (2), σ is the activation function, and $AvgPool$ and $MaxPool$ represent average pooling and max pooling, respectively. The feature map then undergoes convolutional processing, which works in conjunction with the ReLU function. The ReLU function is shown in Equation (3).

$$f(z_i) = \begin{cases} z_i, & z_i > 0 \\ 0, & z_i < 0 \end{cases} \quad (3)$$

In Equation (3), z represents the feature vector and i indicates the index of the vector. Then, different strategies are applied to different types of features. The results are smoothed using the Softmax function, as shown in Equation (4).

$$\lambda(z_i) = \frac{e^z i}{\sum_{j=1}^{K} e^z j} \quad (4)$$

In Equation (4), K and j represent the total number of index categories and the index of the vector, respectively. A fully connected layer then integrates the feature information, as shown in Equation (5).

$$C = W * z_i + b \quad (5)$$

In Equation (5), W is the weight of the fully connected layer and b is the bias term. Finally, the outer product computation obtains high-order relationships among the three types of feature information, as shown in Equation (6).

$$V = X \otimes Y \otimes Z \quad (6)$$

In Equation (6), X, Y, and Z represent the feature values from the three channels. However, this step only generates high-order fused features. Dimensionality reduction and classification are still needed for image recognition. Classification Autoencoder (CAE) is a variant of autoencoder that can reconstruct input data and perform complex classification tasks [18]. CAE's encoder, decoder, and classifier can perform dimensionality reduction, normalization, and classification. Therefore, this study uses CAE for the subsequent image recognition process. After receiving the fused image feature data, the decoder performs dimensionality reduction and maps it to the latent space of the encoder. The process assumes that the data follow a distribution pattern in the latent space, as shown in Equation (7).

$$P(z) \sim (\mu, \delta^2) \quad (7)$$

In Equation (7), z represents the random data, μ is the center of the data distribution, and δ^2 is the squared deviation between z and the center. The total probability of the data distribution in the space is calculated next, as shown in Equation (8).

$$P(X) = \sum P(X|Z)P(Z) \quad (8)$$

In Equation (8), $P(Z)$ is the total space where the data exist, and $P(X|Z)$ is the probability that the data fall exactly in the given region. To obtain the probability of the data's location in the space, an integral computation is performed, as shown in Equation (9).

$$P(Z) = \int_{x^i} P(z|x^i)P(x^i) \quad (9)$$

In Equation (9), $P(z|x^i)$ indicates the correspondence between a specific position and the position in the Gaussian distribution space, and its value is shown in Equation (10).

$$P(z|x^i) \sim G(\mu^i, \delta^{2(i)} I) \quad (10)$$

In Equation (10), G represents the Gaussian distribution, and I is the corresponding vector matrix of the data. Finally, the loss function is calculated to determine the effect of the normalization process, as shown in Equation (11).

$$L(\mu, \delta, x^i) = \log P(x^i) \geq D_{KL}(Q_\theta(z|x^i) \| P(z)) + E_{Q\theta(z|x^i)} \log P\phi(x^i|z) \quad (11)$$

In Equation (11), D_{KL} represents relative entropy, and Q_θ is the posterior approximation distribution. The classifier then completes image recognition based on the magnitude of the values. Ultimately, an adaptive TP algorithm, named TSC (Three-channel Pooling Based on Spatial Attention Mechanism and CAE), is proposed. The workflow is shown in Figure 2.

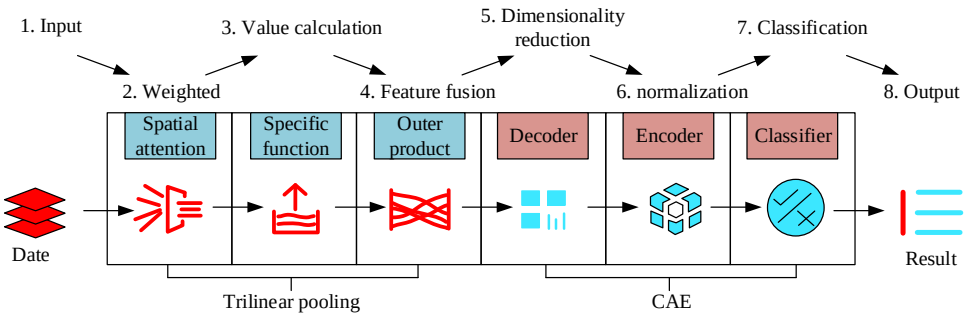


Figure 2: Workflow of the TSC algorithm (Source from: author self-drawn)

In Figure 2, the TSC algorithm comprises six components: weighted processing, feature value calculation, outer product computation, dimensionality reduction, normalization, and classification. After receiving the feature map, the spatial attention mechanism performs adaptive weighting, generating three types of feature maps with different characteristics. Then, the tri-linear channel module calculates the feature values using a specific method, and the outer product module obtains high-order interaction information. This information is then passed in code form to the CAE structure, where the decoder, encoder, and classifier carry out dimensionality reduction, normalization, and classification. The final recognition result identifies the target image content. The CAE encoder and decoder used in this study are both 4-layer fully connected networks. The encoder dimensions are [1024, 512, 256], the latent spatial dimension is 128, and the decoder is its mirrored structure. Except for the output layer, which uses the Sigmoid activation function, all other layers use the ReLU function, and the loss function consists of mean square reconstruction loss and classification cross-entropy loss. For the feature vectors U, V, and W from three channels, the classical trilinear interaction tensor is computed through the outer product. To avoid excessive computation and spatial complexity, this study adopts a low-rank projection strategy. Firstly, project U, V, and W into a low-dimensional space through independent linear layers. Subsequently, the dimensionality-reduced trilinear interaction is computed and

mapped to the final fused feature through flattening and a linear projection layer.

3.2 Construction of the image processing model based on TSC and MP

The TSC algorithm only completes image processing and recognition tasks by fusing existing feature maps. It does not extract feature maps on its own. You Only Look Once version 5 (YOLOv5) is a lightweight image recognition algorithm. Its backbone and neck are responsible for the layered extraction of image features and mainly contribute to improving recognition efficiency [19]. Therefore, this study proposes utilizing the front part of the YOLOv5 network to extract multi-scale feature maps from the target image for the fusion and recognition process in TSC. The trimmed YOLOv5 and the C3 module structure are shown in Figure 3.

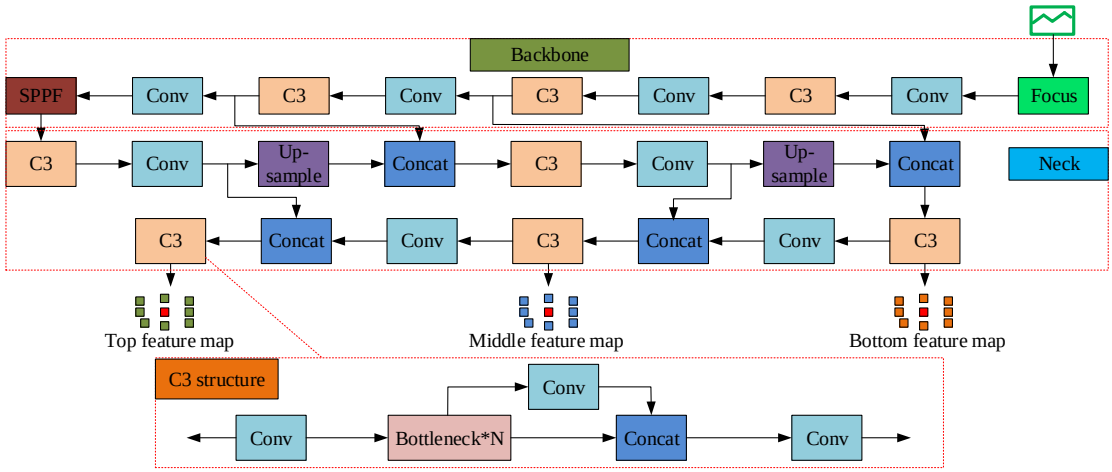


Figure 3: Structure diagram of the partial YOLOv5 network and the C3 module (Source from: author self-drawn)

In Figure 3, the trimmed YOLOv5 network includes a backbone and a neck. The modules in the backbone, such as SPPF, C3, and Conv, work together to extract image information. The main function of the neck is to extract the same feature information between adjacent layers and output it through the C3 module. The C3 module is composed of Conv, Concat, and N Bottleneck modules. The computation process in the Bottleneck module is shown in Equation (12).

$$y = \phi(W_2 * \gamma(W_1 * x + b_1) + b_2) + shortcut(x) \quad (12)$$

In Equation (12), ϕ and γ are activation functions. γ and b represent the weight matrix and bias, $shortcut$ represent the branch operations with different input dimensions. The computation process in the Concat module is shown in Equation (13).

$$C_{r,m,n} = \begin{cases} A_{r,m,n}, 0 \le r < C_A \\ B_{r,m,n}, C_A \le r < C_A + C_B \end{cases} \quad (13)$$

In Equation (13), r, m, and n represent the dimensions, height, and width used during the fusion of feature maps, A and B represent the two layers to be connected. These modules together enable the YOLOv5 backbone and neck to extract high-, mid-, and low-level feature maps. However, the three-channel image recognition method that integrates the front end of YOLOv5 with TSC combines multiple deep learning algorithms. It may increase the operational time cost. MP is a model optimization technique that enhances computational efficiency by evaluating the importance of different parameters and structures, and removing components with minimal impact on the results [20]. Pruning improves model feedback speed while maintaining a certain level of accuracy. This study adopts structured channel pruning as its core strategy, with the importance criterion based on the L1 norm amplitude of channel weights. The pruning plan is iterative, consisting of a total of three iterations. Each iteration prunes 15% of the global channel count, with a final target sparsity of approximately 40%. After each pruning, the model is fine-tuned for 5 epochs with a learning rate set to one-tenth of the initial learning rate. Through this process, the total number of parameters and Giga Floating point Operations Per Second (GFLOPs) of the trimmed model can be significantly reduced. In the trimmed YOLOv5s backbone network, channel simplification was mainly carried out on the C3 module and Conv module in Backbone and Neck. The number of output channels for the 2nd, 4th, and 6th C3 modules in the backbone network has decreased from 128, 256, and 512 to 96, 192, and 384, respectively. The C3 module, after the Concat layer used for feature fusion in the neck network, has reduced its output channel count from 256 to 192. All hidden layer channels in Bottleneck modules are retained in the same proportion. The detailed process is shown in Figure 4.

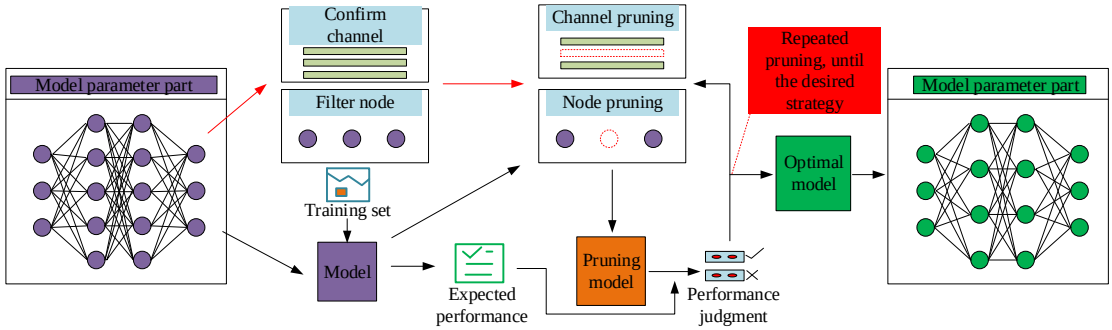


Figure 4: MP process flowchart (Source from: author self-drawn)

In Figure 4, the pruning process first selects redundant nodes and channels in the model. Then, it trains the model using the training set to obtain an expected accuracy. Next, the model removes the nodes and channels identified as unimportant. The pruned model is verified. If it reaches the expected accuracy, the model is output. If it does not meet the target, the process selects other nodes and channels to prune and verifies again until the accuracy meets expectations. During pruning, when the number of channels is reduced, the data may become unstable, and the accuracy may drop significantly. To address this issue, batch normalization is employed to adjust the data distribution after pruning, as illustrated in Equation (14).

$$x'_i = \frac{x_i - E[x]}{\sqrt{Var[x]}} \quad (14)$$

In Equation (14), x'_i is the final data. x_i is the input data. $E[x]$ is the mean value, and $Var[x]$ is the variance. To avoid data being overly uniform, the model data must be adjusted again, as shown in Equation (15).

$$y_i = \gamma x_i + \beta \quad (15)$$

In Equation (15), γ is the scaling factor, and β is the offset. Finally, the study integrates the YOLOv5 backbone, neck, and TSC algorithm to construct a pruned image processing model, named Y-TSC. The structure and workflow of the model are shown in Figure 5.

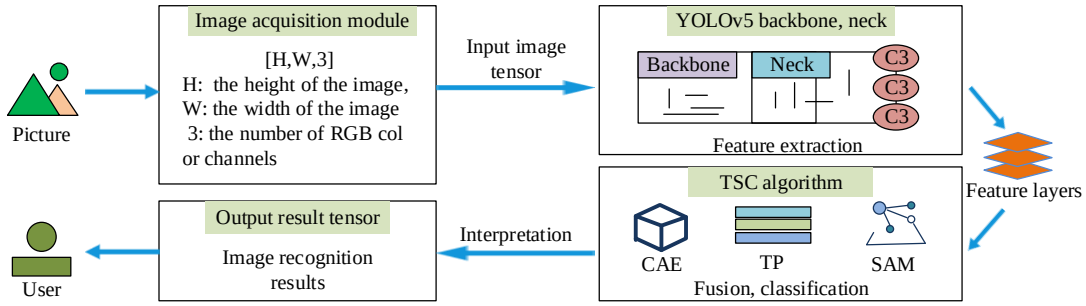


Figure 5: Pruned image processing model integrating YOLOv5 backbone and TSC (Source from: author self-drawn)

In Figure 5, the Y-TSC model after pruning consists of four components: an image acquisition module, a texture feature extraction module, a feature fusion and recognition module, and a result output device. The physical sensor uses cameras and other devices to convert real-world images into digital images that the algorithm can recognize. The texture feature extraction module is composed of the trimmed YOLOv5. It extracts different texture features from low, mid, and high levels to be used in the following TP process. The feature fusion and recognition module is composed of the TSC algorithm. This module applies the spatial attention mechanism to adaptively perform outer product operations on different-dimensional feature maps and finally completes the image recognition process. The result output device includes monitors, speakers, and other human–machine interface devices that convert digital information into physical signals that users can understand. This completes the final result output.

4 Performance validation of the TP-based image recognition model

4.1 Performance validation of the TP-based TSC algorithm

To verify the effectiveness of the Y-TSC model, the study utilized a balanced training subset of AudioSet for model training, which comprises approximately 22000 audio clips covering 527 categories. The validation set uses the Eval subset, which contains approximately 20000 samples. During the testing, 10 common sounds (such as bird songs, cars, natural environments, and human voices) were selected, and 100 samples were randomly selected from each category for recognition testing. Can be obtained from

https://research.google.com/audioset/. The study used Detection Transformer (DETR), Cascade Single Shot MultiBox Detector (CASSD), and Efficient Hybrid Backbone Object Detection (EHBD) as baselines. All baseline models were adjusted for spectral classification tasks and run under identical conditions to ensure fair comparisons. The specific implementation is based on a public repository. DETR utilizes PyTorch from Facebook Research (version 2.0) to adjust the model by replacing the detection header with a fully connected classification header. ResNet-50 and EfficientNet-B0 are implemented using the PyTorch vision library (version 0.15). All models maintain hyperparameter parity. The audio clips are uniformly converted to mono at a sampling rate of 16kHz, and a spectrogram is generated using the short-time Fourier transform. The window length is set to 1024, the hop count is set to 512, and the frequency range is set to 0-8kHz. Logarithmic Mel spectrograms (128 Mel filters) are used. The image resolution is uniformly adjusted to 224 × 224. Data augmentation includes time masking (maximum width: 20 frames) and frequency masking (maximum bandwidth: 10 Mel frequency bands), as well as normalization of mean and variance. Training was conducted for a total of 100 epochs using the Adam optimizer, with weight decay of 1e-4 and momentum parameters β1 and β2 set to 0.9 and 0.999, respectively. The learning rate employed a cosine annealing scheduling strategy, starting from an initial value of 1e-4 and reaching a minimum of 1e-6. Train using fixed random seeds (42) to ensure reproducibility of results, and initiate early stopping when the validation set loss does not decrease for 10 consecutive epochs. All report results are the average of 3 independent runs. The operating system was Ubuntu 22.04.3 LTS, the programming language was Python 3.9, and the deep learning framework was PyTorch-2.0. The CPU used was AMD Ryzen 9 7950X3D, and it is equipped with an NVIDIA GeForce RTX 4090 (24GB VRAM) GPU for model training and inference. When the training batch size is 32, the throughput of a single GPU is about 285 samples/second, with a peak memory usage of 18.2GB. It takes approximately 12.5 hours to complete a full 100-epoch training. The parameter settings for each algorithm are shown in Table 2.

In the training set, the accuracy, loss rate, and ROC curve of each method are shown in Figure 6.

Table 2: Algorithm parameter settings

Parameter	Description	Value
Lr	Learning rate	1e-4
Batch_size	Lot size	32
Hidden_dims_SA	Hidden layer dimension of the spatial attention module	256
Hidden_dims_CAE	Hidden layer dimension of the classification autoencoder	512
Kernel_size	Convolution kernel size	3
Concat_size	Fusion dimension	256
Feature_Dim_Per_Channel	Single-channel input feature dimension	800
Input_Sequence_Length	Model total input feature dimension	2400
Input_size	Input image size	640

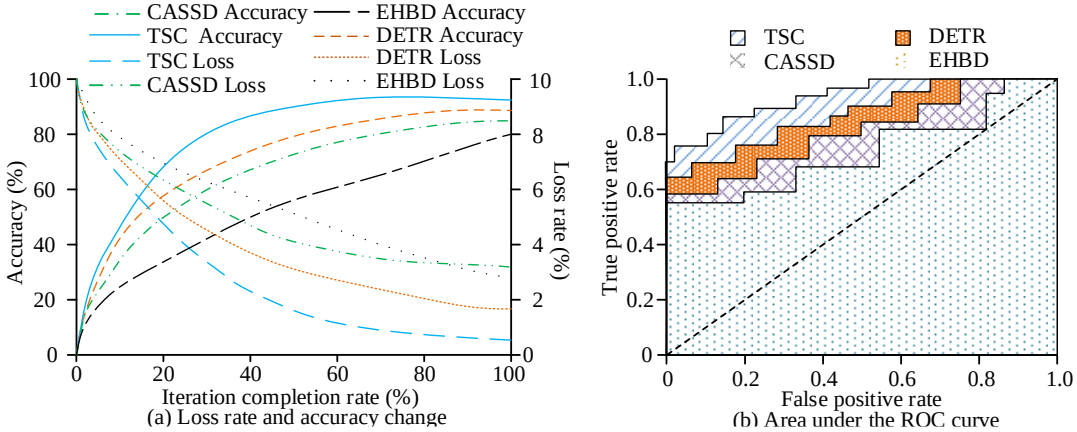


Figure 6: Comparison of accuracy (%), loss rate (%), and ROC curve of various algorithms on a balanced training subset (sample size 22000) (source from: author self-drawn)

In Figure 6(a), the TSC algorithm converged at 60% and achieved an accuracy of 92.7% with a loss rate of 0.5%. The DETR algorithm reached an accuracy of 89.2% and a loss rate of 1.8%. The CASSD algorithm showed an accuracy of 84.5% with a loss rate of 3.2%, while the EHBD algorithm had an accuracy of 80.0% and a loss rate of 1.4%. The accuracy of the standard classifiers ResNet-50 and EfficientNet-B0 under the same conditions was 85.6% and 87.3%, respectively, with loss rates of 2.0% and 1.7%, respectively. Figure 6(b) showed that the ROC curve of the TSC algorithm completely enclosed the curves of all comparison algorithms. The area under the TSC ROC curve was 0.942, compared to 0.875 for DETR, 0.791 for CASSD, and 0.741 for EHBD.

One hundred sound spectrograms were selected from different categories for classification tests. The sampling method involved randomly selecting 100 spectrograms from the test set of each category to ensure category balance and overall representativeness, without reuse. The comparison results are shown in Figure 7.

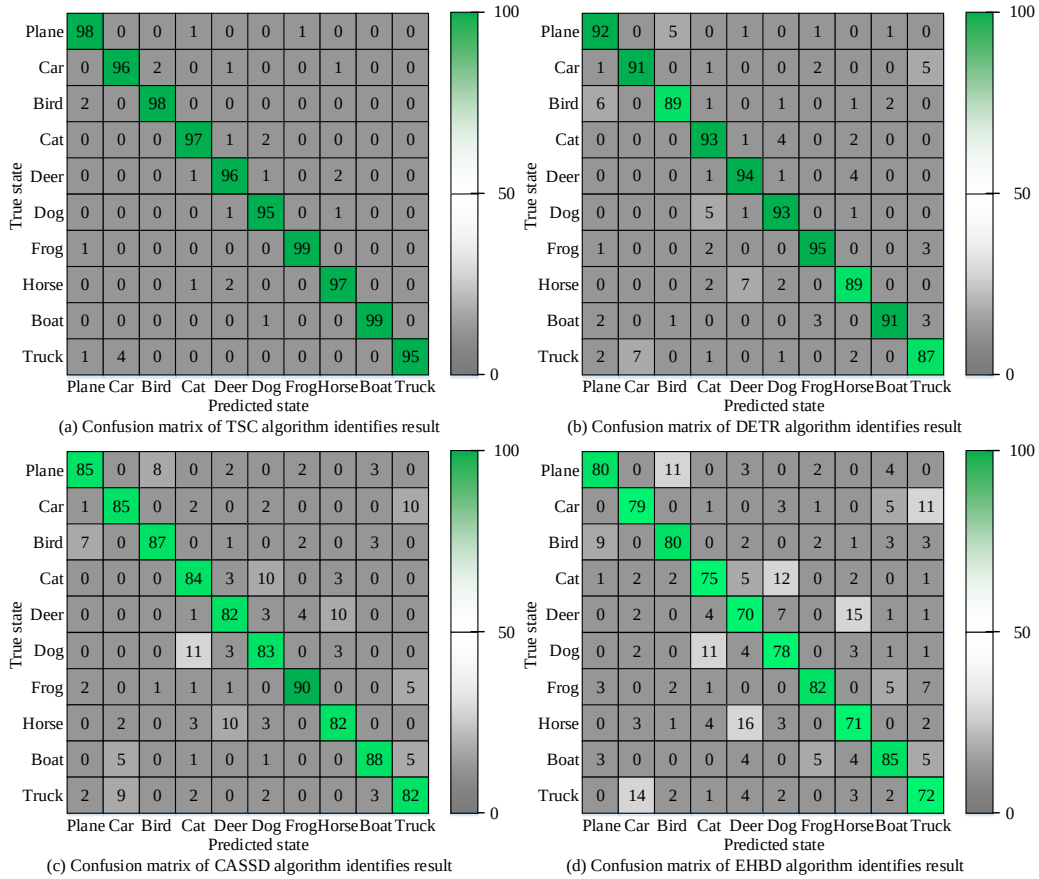


Figure 7: Classification test results of 100 spectrograms using various algorithms(Source from: author self-drawn)

As shown in Figure 7(a), the TSC algorithm correctly classified over 95% of the sound spectrograms in each category. Among them, the number of correctly classified sound spectrograms for frogs and boats was the highest, with both achieving a 99% accuracy. After the calculation, the average recognition rate of the TSC algorithm in all categories has reached 97.0%. Figure 7(b) showed that the DETR algorithm achieved an average recognition rate of 91.4%. Figures 7(c) and 7(d) showed that the CASSD and EHBD algorithms had average recognition rates of 84.8% and 77.2%, respectively. These results demonstrated that the TSC algorithm exhibited significantly better image recognition performance than the comparison algorithms.

To further verify the recognition ability of the TSC algorithm, this study selected and generated 1000 spectrograms for each category from the dataset (a total of 3000) to calculate the confidence interval, and compared the confidence intervals of each algorithm at different confidence levels. The confidence interval was calculated using the Wilson Score method, with 1000 test samples (instead of the original 100) for each category, to provide more stable interval estimates. To ensure the reliability of the confidence interval, the original probability output by the model was calibrated using the Platt scaling method to improve the consistency between the probability value and the actual confidence level. The results are shown in Table 3.

Table 3: Confidence intervals of dogs, cats, and birds under different confidence levels

Category	Image category	Accuracy (%)	Confidence			
			80%	85%	90%	95%
TSC	Bird sounds	95.3	(0.93,0.98)	(0.92,0.98)	(0.91,0.99)	(0.91,0.99)
	Car sounds	96.9	(0.91,1.00)	(0.90,1.00)	(0.89,1.00)	(0.88,1.00)
	Natural environment sounds	98.1	(0.92,1.00)	(0.92,1.00)	(0.91,1.00)	(0.90,1.00)
DETR	Bird sounds	93.1	(0.88,0.98)	(0.87,0.99)	(0.86,1.00)	(0.85,1.00)
	Car sounds	92.9	(0.86,0.94)	(0.86,0.94)	(0.85,0.95)	(0.84,0.96)
	Natural environment sounds	89.5	(0.76,0.90)	(0.75,0.91)	(0.74,0.92)	(0.72,0.94)
CASSD	Bird sounds	83.4	(0.76,0.90)	(0.75,0.91)	(0.74,0.92)	(0.72,0.94)
	Car sounds	84.5	(0.77,0.91)	(0.76,0.92)	(0.75,0.93)	(0.73,0.95)
	Natural environment sounds	86.8	(0.80,0.90)	(0.79,0.91)	(0.78,0.91)	(0.78,0.92)
EHBD	Bird sounds	78.69	(0.71,0.86)	(0.70,0.87)	(0.69,0.88)	(0.68,0.90)
	Car sounds	75.6	(0.68,0.82)	(0.67,0.83)	(0.66,0.84)	(0.67,0.83)
	Natural environment sounds	80.2	(0.75,0.85)	(0.74,0.86)	(0.73,0.87)	(0.72,0.88)

The accuracy in Table 3 is the Top-1 accuracy for each category, calculated based on the test samples of each category. The TSC algorithm achieved recognition accuracies of 95.3% for bird sounds, 96.9% for car sounds, and 98.1% for natural environment sounds. At the 95% confidence level, the confidence intervals were (0.91, 0.99) for bird sounds, (0.88, 1.00) for car sounds, and (0.90, 1.00) for natural environment sounds. In contrast, for the DETR algorithm, the confidence intervals at the same confidence level were (0.85, 1.00), (0.84, 0.96), and (0.72, 0.94), respectively. For the CASSD algorithm, the intervals were (0.72, 0.94), (0.73, 0.95), and (0.78, 0.92); and for the EHBD algorithm, they were (0.68, 0.90), (0.67, 0.83), and (0.72, 0.88). The upper limit of the partial confidence interval is close to 1.00, reflecting the high accuracy of the model in these categories. To avoid interval estimation bias caused by small sample sizes, this study employed the bootstrap sampling method (repeated 1000 times) to verify interval stability, and the results were consistent with the reports in the table.

4.2 Practical validation of the TP-based model

Based on the validation of the TSC algorithm, the performance of the Y-TSC model was further evaluated. A set of real-world images was used as the sample for testing. The performance of each model was assessed based on the quality of output images and processing time. Models based on DETR, CASSD, and EHBD were used for comparison. To verify whether the Y-TSC model had advantages in processing speed, a group of 200 images was used to test the processing efficiency of each model. The results are shown in Figure 8.

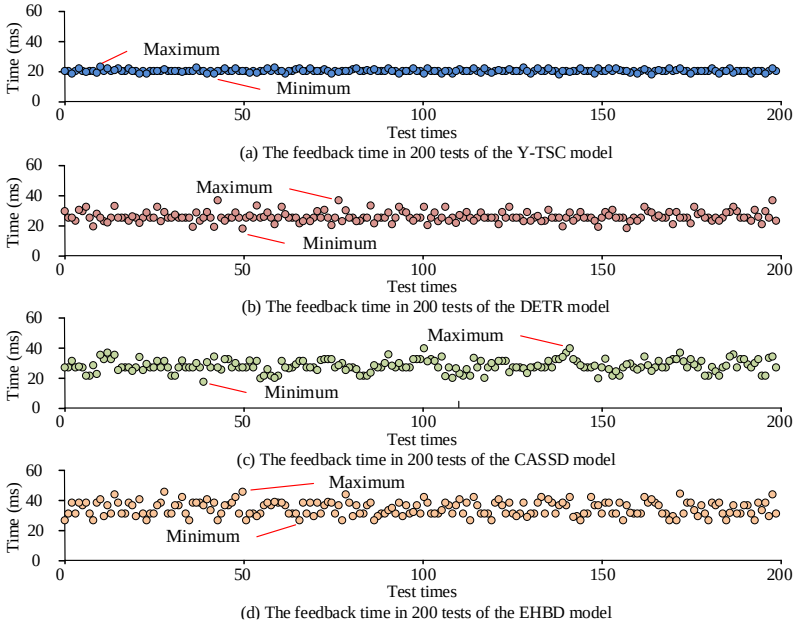


Figure 8: Feedback time of each model for 200 images (ms) (Source from: author self-drawn)

As shown in Figure 8(a), the feedback time of the Y-TSC model was concentrated around 20 ms, with a maximum of 23.1 ms and a minimum of 18.5 ms. In contrast, Figures 8(b), 8(c), and 8(d) showed that the response time distributions of the comparison models were more scattered. The DETR model had a maximum response time of 39.58 ms and a minimum of 18.6 ms. The CASSD model showed a maximum of 42.1 ms and a minimum of 18.6 ms, while the EHBD model ranged from 22.2 ms to 43.1 ms. These results indicated that the Y-TSC model significantly outperformed the comparison models in processing efficiency.

To further evaluate the accuracy of the Y-TSC model, the feature recognition accuracy of each model was compared across 200 tests, covering low-level, mid-level, and high-level features, as well as overall recognition accuracy. The results are shown in Figure 9.

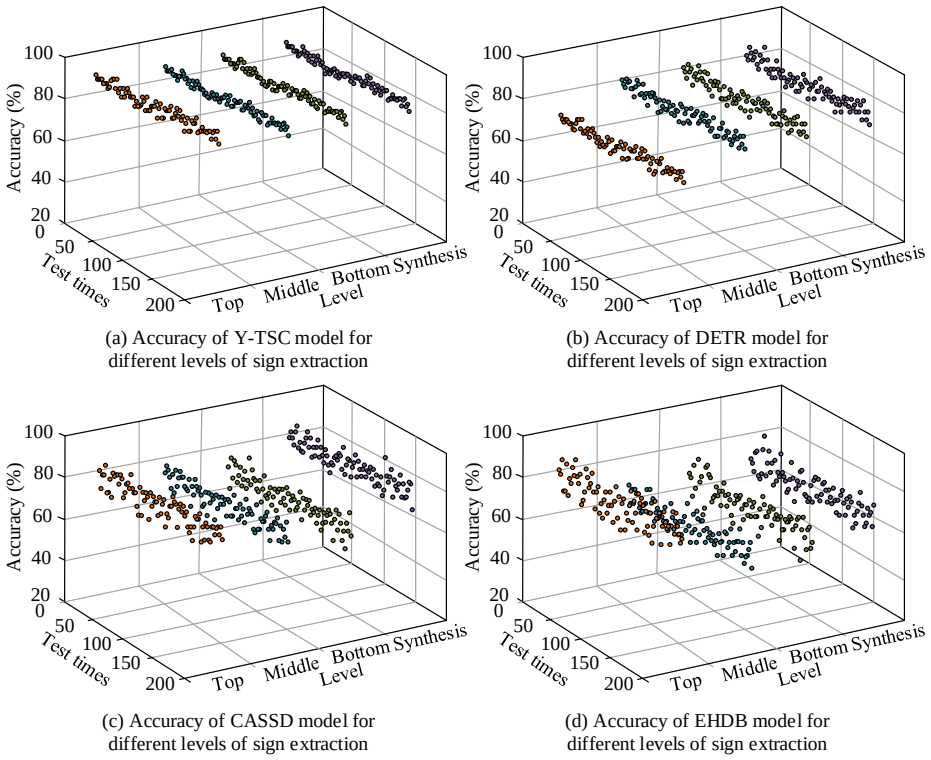
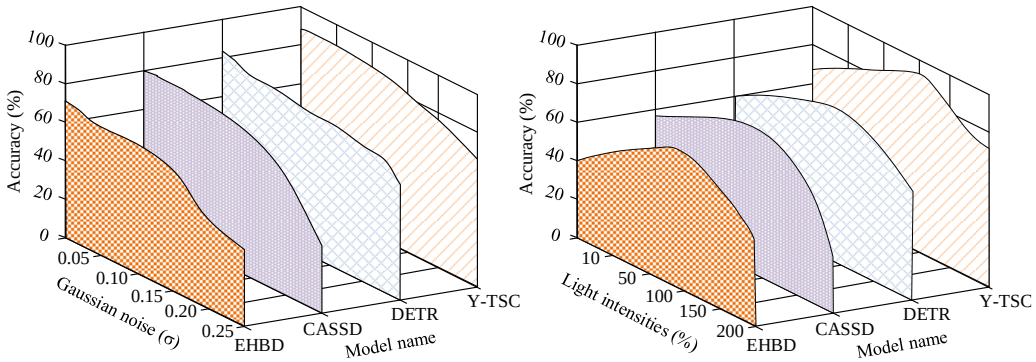


Figure 9: Recognition accuracy of each model on 200 images (%) (Source from: author self-drawn)

As shown in Figure 9(a), the Y-TSC model consistently maintained a feature recognition accuracy above 90% for each level, with an overall accuracy also at 90%. The test results were mostly aligned, indicating high model stability. In Figure 9(b), although the DETR model maintained an overall accuracy above 80%, its accuracy for low-level features remained around 70%, which might have limited its overall performance. Figures 9(c) and 9(d) showed that both the CASSD and EHBD models achieved significantly lower recognition accuracy across all layers, with more dispersed data points. These findings suggested that the Y-TSC model provided higher accuracy.

To verify the robustness of the models, all models were tested under different levels of Gaussian noise and lighting intensity. The results are presented in Figure 10.



(a) Accuracy under different Gaussian noise intensities (b) Accuracy under different light intensities intensities

Figure 10: Accuracy (%) of each model under different Gaussian noise and lighting conditions (Source from: author self-drawn)

As shown in Figure 10(a), the accuracy of all models decreased with increasing Gaussian noise. The accuracy of the Y-TSC model decreased from 90.2% at a 0.05σ noise level to 71.2% at a 0.25σ noise level. In comparison, the accuracy of the DETR model declined from 82.5% to 70.6%, the CASSD model from 79.5% to 40.7%, and the EHBD model from 70.7% to 52.3%. Figure 10(b) showed that all models experienced decreased accuracy under both low and high lighting conditions. At a 10% lighting intensity, the Y-TSC model achieved an accuracy of 69.2%, which was significantly higher than that of the comparison models. At 200% lighting intensity, the Y-TSC model still maintained a high accuracy of 71.2%, again surpassing the others. In summary, the Y-TSC model consistently outperformed the comparison models under high noise, highlighting, and low-lighting conditions, demonstrating stronger robustness.

Ablation experiments were conducted to quantitatively evaluate the independent contributions of each innovative component in the Y-TSC model. Five control models were designed in total: (a) Baseline: using only the trimmed YOLOv5 backbone network. (b) Add a spatial attention module on the basis of (a). (c) On the basis of (b), upgrade the feature fusion method from bilinear pooling to trilinear pooling; (d) On the basis of (c), introduce CAE to replace the original classifier; (e) Complete the Y-TSC model. Using average recognition accuracy, model parameter count, GFLOPs, and CPU latency as evaluation metrics. All models were independently run 5 times, presenting indicator results in the form of mean ± standard deviation. The results of the ablation experiment are shown in Table 4.

Table 4: Results of the ablation experiment

Models	Accuracy/%	Parameter count/M	GFLOPs	CPU latency/ms
a	84.5±0.3*	4.1	9.8	45.3±0.5*
b	86.1±0.4*	4.3	10.4	48.2±0.6*
c	89.2±0.2*	4.7	12.9	55.7±0.6*
d	91.8±0.3*	7.5	16.7	68.8±0.8*
e	92.7±0.2	4.5	10.1	50.4±0.5

Note: "*" indicates significant differences compared to the complete Y-TSC model, $P<0.01$.

As shown in Table 4, the recognition accuracy of model (a) is the lowest, only 84.5%. The recognition accuracy of the complete model (e) is the highest, reaching 92.7%, significantly higher than the other four models ($P<0.01$). And the parameter count, GFLOPs, and CPU latency of the complete model (e) are lower than those of the (d) model. This indicates that the components proposed by this study institute have made significant contributions to the model's performance, successfully achieving a balance between

accuracy and efficiency, resulting in the best overall performance.

5 Discussion

To address the challenge in existing image processing technologies of balancing both accuracy and efficiency, this study proposed a method for image recognition based on TP by analyzing image textures at three levels: low, middle, and high. The complete image recognition process was implemented through the proposed Y-TSC model, which integrated a Coupled Spatial Attention Mechanism, a CAE, and the YOLOv5 backbone. Experimental results showed that the TSC algorithm, used for image feature fusion and recognition, converged at 60% training progress with an accuracy of 92.7%, a loss rate of 0.5%, and a ROC value of 0.942. On the AudioSet dataset, the TSC algorithm achieved an average recognition accuracy of 97.0% across ten image categories. In the performance evaluation of the Y-TSC model, the feedback time never exceeded 23.1 ms, which was significantly lower than that of the comparison models. The Y-TSC model effectively extracted texture features from the low, middle, and high levels of images, enabling accurate image recognition. In the robustness tests, the Y-TSC model demonstrated strong resistance to Gaussian noise, maintaining an accuracy of 71.2% even under a noise intensity of 0.25σ. Under lighting conditions with only 10% of normal brightness, the model achieved an accuracy of 69.2%, and under 200% lighting intensity, it reached 71.2%.

The performance differences between this study and existing SOTA methods mainly stem from differences in architecture design, training strategies, and computational budgets. Compared to He et al. [6], who only used trilinear pooling for multimodal fusion without optimizing computational efficiency, this study introduces low-rank projection and structured pruning to reduce GFLOPs to 10.1 while maintaining feature interaction capability. Unlike Hu et al. [9], who relied on Jetson TX2 dedicated hardware to achieve real-time performance, this study achieved an inference speed of 23.1 ms on a general-purpose GPU through global channel pruning and progressive fine-tuning strategies. These improvements in architecture and training system have enabled the proposed Y-TSC model to achieve better overall performance under the same computational budget.

6 Conclusion

In summary, the proposed method and model successfully met the requirements of both accuracy and efficiency in the image recognition process, demonstrating a certain degree of resistance to interference. However, this study did not evaluate the resource and energy consumption involved in model construction, nor did it assess the economic feasibility of the recognition process. Future research will focus on addressing these issues and continuously improving and optimizing the model.

Funding

The research is supported by the Research Foundation of the Natural Science Foundation of Hunan Province, (Grant No. 2024JJ7189); the Social Science Project of Hunan Provincial Achievement Review Association, (Grant No.XSP24YBC319)； Hunan Province General Higher Education Teaching Reform Research Project (HNJG-20231101); Hunan Province General Higher Education Teaching Reform Research Project (HNJG-20231094).

References

[1] Putri R K, Athoillah M. Detection of facial mask using deep learning classification algorithm. Journal of Data Science and Intelligent Systems, 2024, 2(1): 58-63. https://doi.org/10.47852/bonviewJDSIS32021067

[2] Premnath S P, Gowr P S, Ananth J P, Arumugam S R. Image enhancement and blur pixel identification with optimization-enabled deep learning for image restoration. Signal, Image and Video Processing, 2024, 18(5): 4525-4540. https://doi.org/10.1007/s11760-024-03092-6

[3] Kumar HN N, Kumar A S, Prasad MS G, Shah M A. Automatic facial expression recognition combining texture and shape features from prominent facial regions. IET Image Processing, 2023, 17(4): 1111-1125. https://doi.org/10.1049/ipr2.12700

[4] SHI J, XU Y, CAO B. Fine-grained Image Classification Based on Adaptive Trilinear Pooling Network. Computer Engineering, 2023, 49(5): 239-246.

[5] Mpia H N, Syasimwa L M, Muyisa D M. Comparative machine learning models for predicting loan fructification in a semi-urban area. Archives of Advanced Engineering Science, 2025, 3(2): 124-134. https://doi.org/10.47852/bonviewAAES42022418

[6] He Y, Gan M G, Ma Q. Online video visual relation detection with hierarchical multi-modal fusion. Multimedia Tools and Applications, 2024, 83(24): 65707-65727. https://doi.org/10.1007/s11042-023-15310-3

[7] Sun R, Hu K, Martens K A E, Hagenbuchner M, Tsoi A C, Bennamoun M, Lewis S J G, Wang Z. Higher order polynomial transformer for fine-grained freezing of gait detection. IEEE transactions on neural networks and learning systems, 2023, 35(9): 12746-12759. https://doi.org/10.1109/TNNLS.2023.3264647

[8] Lee S B, Cho Y J, Yoon S H, Lee Y Y, Kim S H, Lee S, Cheon J. E. Automated segmentation of whole-body CT images for body composition analysis in pediatric patients using a deep neural network. European Radiology, 2022, 32(12): 8463-8472. https://doi.org/10.1007/s00330-022-08829-w

[9] Hu Y, Shuai Z, Yang H, Wan G, Zhang Y, Xie C, Lu X. ESDAR-net: towards high-accuracy and real-time driver action recognition for embedded systems. Multimedia Tools and Applications, 2024, 83(6): 18281-18307.

https://doi.org/10.1007/s11042-023-15777-0

[10] Grimm C, Fei T, Warsitz E, Farhoud R, Breddermann T, Haeb-Umbach R. Warping of radar data into camera image for cross-modal supervision in automotive applications. IEEE Transactions on Vehicular Technology, 2022, 71(9): 9435-9449. https://doi.org/10.1109/TVT.2022.3182411

[11] Wang S, Wang Z, Wang S, Ye Y. Deep image compression toward machine vision: A unified optimization framework. IEEE Transactions on Circuits and Systems for Video Technology, 2022, 33(6): 2979-2989. https://doi.org/10.1109/TCSVT.2022.3230843

[12] Vasanthi M, Seetharaman K. Facial image recognition for biometric authentication systems using a combination of geometrical feature points and low-level visual features. Journal of King Saud University-Computer and Information Sciences, 2022, 34(7): 4109-4121. https://doi.org/10.1016/j.jksuci.2020.11.028

[13] Hassan E, Shams M Y, Hikal N A, Elmougy S. The effect of choosing optimizer algorithms to improve computer vision tasks: a comparative study. Multimedia Tools and Applications, 2023, 82(11): 16591-16633. https://doi.org/10.1007/s11042-022-13820-0

[14] G Sahu, Seal A, Bhattacharjee D, Frischer R, Krejcar O. A novel parameter adaptive dual channel MSPCNN based single image dehazing for intelligent transportation systems. IEEE Transactions on Intelligent Transportation Systems, 2022, 24(3): 3027-3047. https://doi.org/10.1109/TITS.2022.3225797

[15] Frants V, Agaian S, Panetta K. QCNN-H: Single-image dehazing using quaternion neural networks. IEEE Transactions on Cybernetics, 2023, 53(9): 5448-5458. https://doi.org/10.1109/TCYB.2023.3238640

[16] Patil N. An enhanced segmentation technique and improved support vector machine classifier for facial image recognition. International Journal of Intelligent Computing and Cybernetics, 2022, 15(2): 302-317. https://doi.org/10.1108/IJICC-08-2021-0172

[17] Nath M, Mitra P, Kumar D. A novel residual learning-based deep learning model integrated with attention mechanism and SVM for identifying tea plant diseases. International Journal of Computers and Applications, 2023, 45(6): 471-484. https://doi.org/10.1080/1206212X.2023.2235750

[18] Puzyrev V, Elders C. Unsupervised seismic facies classification using deep convolutional autoencoder. Geophysics, 2022, 87(4): 125-132. https://doi.org/10.1190/geo2021-0016.1

[19] Zhou M, Wu L, Liu S, Li J. UAV forest fire detection based on lightweight YOLOv5 model. Multimedia Tools and Applications, 2024, 83(22): 61777-61788. https://doi.org/10.1007/s11042-023-15770-7

[20] Saadallah A, Jakobs M, Morik K. Explainable online ensemble of deep neural network pruning for time series forecasting. Machine Learning, 2022, 111(9): 3459-3487. https://doi.org/10.1007/s10994-022-06218-4