applied sciences

MDPI

Article

CASVM: An Efficient Deep Learning Image Classification Method Combined with SVM

Shuqiu Tan *, Jiahao Pan, Jianxun Zhang and Yahui Liu

College of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400054, China
* Correspondence: tsq@cqut.edu.cn

Abstract: Recent advances in convolutional neural networks (CNNs) for image feature extraction have achieved extraordinary performance, but back-propagation algorithms tend to fall into local minima. To alleviate this problem, this paper proposes a coordinate attention-support vector machine-convolutional neural network (CASVM). This proposed to enhance the model's ability by introducing coordinate attention while obtaining enhanced image features. Training is carried out by back-propagating the loss function of support vector machines (SVMs) to improve the generalization capability, which can effectively avoid falling into local optima. The image datasets used in this study for benchmark experiments are Fashion-MNIST, Cifar10, Cifar100, and Animal10. Experimental results show that compared with softmax, CASVM can improve the image classification accuracy of the original model under different image resolution datasets. Under the same structure, CASVM shows better performance and robustness and has higher accuracy. Under the same network parameters, the loss function of CASVM enables the model to realize a lower loss value. Among the standard CNN models, the highest accuracy rate can reach 99%, and the optimal number of accuracy indicators is 5.5 times that of softmax, whose accuracy rate can be improved by up to 56%.

Keywords: CASVM models; coordinate attention; feature learning method

check for updates

Citation: Tan, S.; Pan, J.; Zhang, J.; Liu, Y. CASVM: An Efficient Deep Learning Image Classification Method Combined with SVM. *Appl. Sci.* **2022**, *12*, 11690. https://doi.org/10.3390/app122211690

Academic Editor: Zeng Nianyin

Received: 23 October 2022
Accepted: 15 November 2022
Published: 17 November 2022

1. Introduction

Image classification is a crucial task of computer vision, whose goal is to divide an image into feature categories according to its feature information. Based on statistical learning theory, support vector machine (SVM) [1] is a kernel-based machine learning model that has shown promising classification results in fields such as pattern recognition, pattern classification, and computer vision [2]. However, traditional SVM has difficulty processing large amounts of image data and meeting the requirements of classification accuracy, and is therefore unsuitable for large and complex image classification. Deep learning (DL) learns features autonomously from images through neural networks, extracts abstract, high-dimensional features, and can tightly connect features to classifiers [3–5].

The convolutional neural network (CNN) is a widely used structure in DL models [6], with good results in image classification and recognition, such as of handwritten digits, where it has reached the accuracy of human vision, which was recognized in 2012 for the best image classification effect in the ImageNet Large-scale Visual Recognition Challenge [7].

With increasing advantages of neural networks in processing image features, traditional classification methods are combined with DL. SVM and softmax are two standard classifiers in computer vision, and with the improvement of CNN algorithms, the use of the two classifiers is increasing. Classification of extracted data is an essential aspect of DL. SVM can classify pre-extracted data and take the scores of data as a basis for evaluation [8,9]. A linear SVM has replaced the softmax classifier in most DL models, and SVM has been used as the final classification unit of a CNN from feature extraction to classification, achieving better classification discrimination of L2-SVM over softmax on a DL image classification dataset [3]. Baldomero-Naranjo et al. [10], Thillaikkarasi [11], and Nguyen et al. [12] used a CNN to extract data features. They used SVM for feature extraction and classification, to somewhat better effect than a CNN. Nanglia et al. [13] and Sun et al. [14] combined deep neural networks with SVM in a hybrid algorithm to my in-depth features, which improved the classification or recognition accuracy of data. Khairandish et al. [15] and Gong et al. [16] fused an SVM classifier in a neural network with robust classification accuracy. SVMs and neural networks are used in the fields of image classification and recognition, such as human biology [17–19], medical imaging [11,20–23], and remote sensing [24,25], performing better at complex tasks than original neural network models.

The attention mechanism [26] is gaining importance in DL in image classification. Mnih et al. [27] integrated a recursive neural network into an attention mechanism to adaptively extract features in images, promoting large-scale image classification and recognition. Attention mechanisms have been widely used in computer vision and in areas such as finance, materials, meteorology, industry, and emotion detection [28]. In computer vision, these include the squeeze-and-excitation network (SENet) [29] and convolutional block attention module (CBAM) [30], but these cannot establish long-term dependencies on image classification or recognition tasks when capturing features. Although mechanisms such as GSoPNet [31], SCNet [32], and A2Net [33] use non-local mechanisms to overcome the defects of SENet and CBAM, they have significant computational overhead. Coordinate attention (CA) [34] aggregates features in two spatial directions to capture position information and channel relations with low computational overhead, and achieves the feature representation of an enhanced network.

SVM is an alternative to softmax classification, and fusing it with CNNs usually improves image classification accuracy, but features of lower levels do not adjust to the objectives of SVM. Inspired by hinge loss in SVM, this paper enables the backpropagation of the gradient in a CNN by the hinge loss, thus learning lower-level features. Inspired by CA, features can be enhanced by embedding location information in channel attention, without a large overhead, providing a method for efficient image classification. The method fuses a CNN embedded in a CA mechanism with an SVM for image classification, which we call CA-support vector machine-CNN (CASVM). A CNN based on an attention mechanism is used to extract features classified by linear SVM. An empirical analysis of commonly used image datasets verifies the accuracy of the model for image classification.

This study makes the following contributions:

1. An SVM classifier replaces the softmax classifier commonly used in CNNs to improve the robustness of image classification; the hinge loss function of SVM is used to backpropagate the CNN to improve the generalization performance of the classifier on images, which reduces overfitting and improves classification results.
2. The CA mechanism used by CASVM has two components, coordinate information embedding (CIE) and CA generation (CAG), to enable the network to acquire information over a larger area without significant overhead. CIE captures location information and establishes long-range dependencies in the spatial direction by decomposing channel attention. CAG transforms the generated features to locate the position of the object of interest.
3. CASVM combines a CNN with an SVM that introduces a CA mechanism to improve classification accuracy. CASVM can improve feature representation performance in different network models, which improves image classification accuracy in different device environments.

2. Related Work

A CNN is a feedforward neural network with convolutional computation that has excelled in image algorithms due to its exceptional learning power. From the traditional LeNet [35] and AlexNet [7] to the more modern MobileNet [36], ResNet [37], Nasnet [38], Efficient [39], Polynet [40], etc, they have all demonstrated excellent performance. Even so, the network model's classification layer in image classification tasks continues to use the softmax function and cross-entropy loss for back-propagation, which can cause the model to become over-fitted and degrade the model's performance. In contrast, when SVM is used as the classifier, it learns the lower-level parameters by back-propagating the gradient of the last layer, which improves the model's generalization ability [3,16,22,41].

In recent years, several excellent works combining SVM and CNN have been proposed. These techniques typically fall into two broad categories: image data is fed directly into a CNN-Softmax [16,42] model to complete CNN model training, and then the CNN is used as a feature extractor to extract feature vectors to feed into the SVM to achieve image classification. The other option is to replace the Softmax function with SVM to achieve end-to-end CNN-SVM training with improved operability and generality [3,4,22,43]. Furthermore, the method does not require any manual feature extraction or raw data processing, and it does not rely on expert experience or prior knowledge. Both of these approaches are intended to enhance the algorithm's generalizability and avoid over-fitting the model. However, using such techniques can result in drawbacks like the inability to do multi-classification or issues like low multi-classification accuracy [4]. Researchers have discovered one-to-all or one-to-one [44–47] SVM classification approaches to address the issue of limiting multi-classification and have concentrated their research on feature extraction models paired with SVM to increase multi-classification accuracy [48].

As the Vision Transformer (VIT) [49] model has achieved excellent performance in downstream tasks such as image classification and target detection [50–52], it has attracted the attention of many researchers. The key reason for the success of VIT is the adaptive, remote dependence through self-attentive operations to extract a global understanding of the visual scene. VIT expands the perceptual field of an image by building a global relational model, which in turn obtains more information about image features. However, VIT has a complex computational structure and large data requirements, and even the lightweight VIT still has a huge computational cost and is not applicable to edge devices [53]. In contrast, CNN with attention mechanism has received more and more attention due to its efficiency and effectiveness. Attention mechanisms widely used in image classification tasks are channel attention and spatial attention. In channel attention, different channels in different feature maps usually represent different objects [54]. The channel attention mechanism can adaptively recalibrate the weights of each channel, viewing this process as a focused object selection process, thus eliminating the need for larger computations to determine what image features to pay attention to [29,31,55–57]. Spatial attention, on the other hand, can be seen as where to pay attention [27,58–60]. Channel attention combined with spatial attention enables the adaptive selection of objects and regions worthy of attention [30,54,61]. However, such an attention mechanism, while emphasizing the importance of space and channels, also brings a higher computational cost and limited perceptual field. In contrast, the coordinate attention mechanism combines the advantages of SENet [29], and CBAM [30] to capture location information and channel relationships with very low computational overhead.

3. Methods

3.1. Overview

Figure 1 shows the model architecture of this paper, in which we combine a CNN with an embedded CA mechanism and SVM for feature learning. The network embedded with an attention mechanism can extract highly representative image features and learn the essential laws of data. The last layer of the CNN outputs the boundary probability of the classification calculated by the linear combination of the output weights and bias terms of the previous hidden layer, for which the CNN hidden layer output has no special significance. Therefore, after learning the features, the output values must be input to a classifier effective in image classification, and SVMs are among the best. Therefore, we combine these methods in CASVM, which outperforms softmax at image classification.

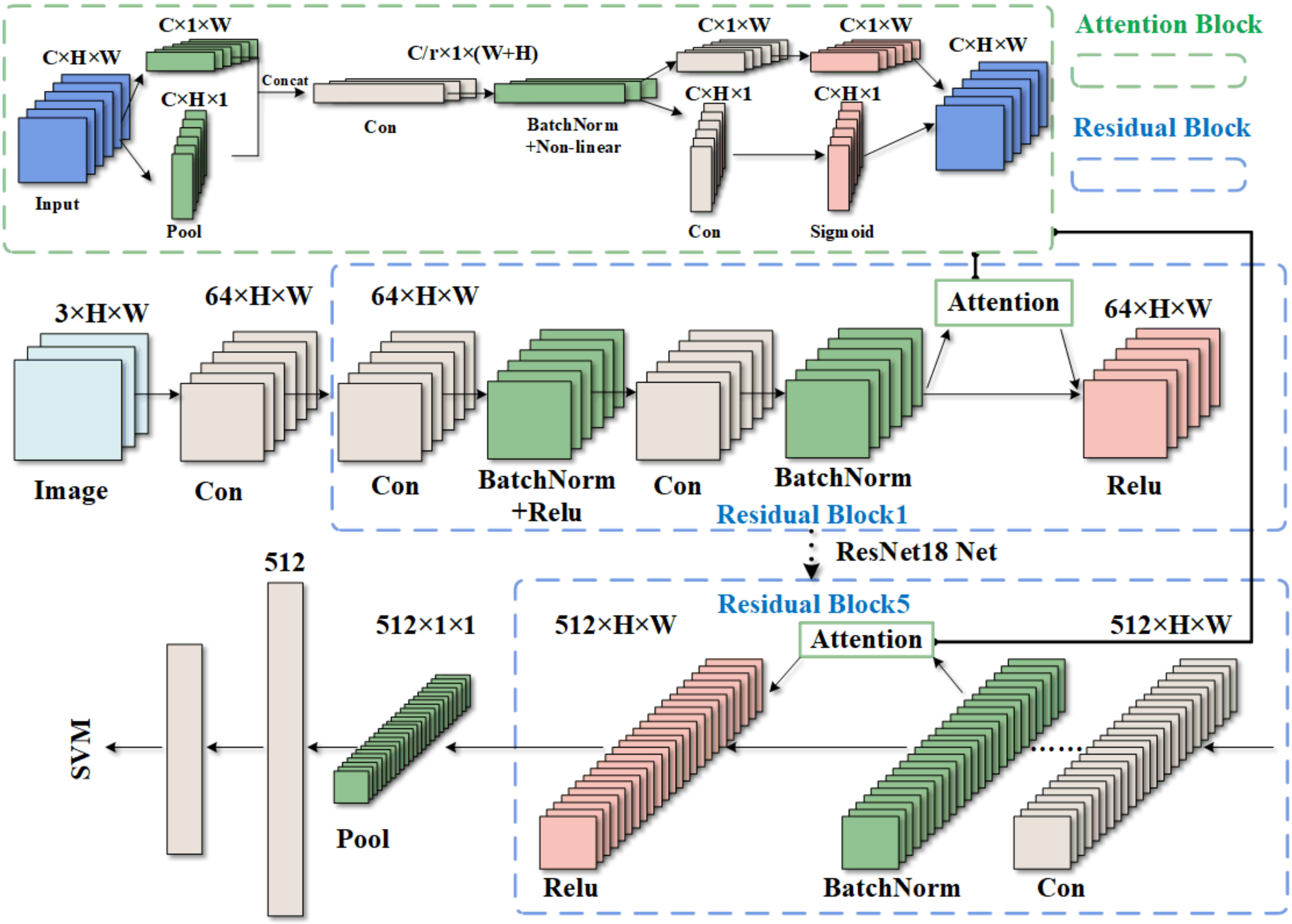


Figure 1. CASVM using ResNet18. Image goes through five residual blocks, with convolutional layer, attention block, and ReLU activation function. Attention block averages pooled data along two directions, reduces number of channels by a factor of r, and uses operations such as 1 × 1 convolution to obtain a number of channels equal to the input. Data go into SVM after final residuals for classification.

CASVM is a combination of CNN embedded with CA and SVM; Figure 1 shows the CNN using ResNet18 as an example. We enter an image of size C × H × W, where C is the number of channels on the feature graph, H is the height, and W is the width. Before the image is sent to the residual (blue) block, convolution is carried out to preserve the feature information as much as possible. The last data normalization in the residual block is entered in the attention block (green box). The feature maps are average pooled along the horizontal and vertical directions, and a factor of r reduces the number of channels of the feature maps. The channels are compressed by concatenation of spatial dimensions and 1 × 1 convolution, spatial information in both directions is segmented into feature maps by batch normalization and nonlinear encoding, and the same number of channels as the input is obtained using 1 × 1 convolution. A sigmoid function normalizes weight. Subsequent residual blocks are designed to nest the attention block between data normalization and activation, as shown in the first residual block, effectively amplifying the differences between image features to differentiate image classes better. The weight is normalized by a sigmoid function. Subsequent residual blocks embed the attention block between data normalization and activation, effectively magnifying the differences between image features to distinguish image categories better. After residual computation, two fully connected layers represent classes that divide the images, and a multiclassification SVM loss function learns the network parameters.

The CASVM model combines the advantages of CNN and SVM with embedded attention mechanisms to outperform a single classifier for image classification. The CNN is an empirical risk minimization model, which can fall into local minima when backpropagation finds the classification hyperplane. SVM is a structural risk minimization model, which can avoid local optima by solving a quadratic programming problem. Hence, its generalization ability is better than that of CNN, and replacing the output layer of CNN with SVM can further improve image classification accuracy.

A CNN can perform downsampling and convolution on images to extract representative image features, with an attention mechanism to enhance effective information. Attention is focused on the image region with a more significant impact on classification extract more robust image features. Manually designed feature extraction methods may be complex and require more skill, and the extracted features may not be universally valid. DL methods can avoid much manual involvement and extract more significant features than traditional methods, which is the unique advantage of combining DL with SVMs.

3.2. SVM Loss Function

Hinge loss is often used as the objective function of SVM. Standard hinge losses are focused on dichotomous problems. For example, given a training dataset $x_i \in R^D$, $i \in \{1, 2, \cdots, N\}$, $y_i \in \{-1, +1\}$, suppose there are N samples, each with dimension D, and xw is the predicted value of a linearly separable SVM, Y is the category of correct classification, and w is a parameter that a classifier can learn from the samples. Then its hinge loss can be expressed as

$$L_i = max(0, 1 - x_i w^T y_i) \tag{1}$$

for the objective function of a binary classification problem, the loss value is 0 when $x_i w^T y_i$ is greater than 1, i.e., the loss function prediction is accurate.

Since SVM can filter samples at long distances from the hyperplane by choosing a threshold for the distance from the sample to the hyperplane, these samples are often the ones that can be easily and correctly classified, thus extending the loss function to a multi-classification SVM. Given $y_i \in \{1 \cdots K\}$, the multi-classification hinge loss for a single sample can be expressed as

$$L_i = \sum_{j \neq y_i} max(0, 1 + w_j^T x_i - w_{y_i}^T x_i) \tag{2}$$

where w_j is the j-th component in parameter vector w. The multi-classification SVM is to set the correctly classified prediction score at least one threshold higher than the other incorrectly classified prediction scores. If the other classification prediction scores are below the threshold, the loss value is calculated. Due to the uniqueness of w, a regularization penalty is introduced to remove ambiguity. Most commonly used is the L2 paradigm. Hence, the multi-classification SVM loss function contains the regularization penalty, and the loss function of k-classification linear SVM can be expressed as

$$\underset{w}{min} \sum_{i=1}^{N} \sum_{j \neq y_i} max(0, 1 + w_j^T x_i - w_{y_i}^T x_i) + \lambda \sum_k \sum_n w_{k,n}^2 \tag{3}$$

Equation (3) can be regarded as the loss function of multi-classification SVM by data loss plus regularization loss, where hinge loss is data loss, and the L2 normal form is regularization loss. Since the penalty in L2 regularization is biased toward smaller, relatively dispersed weight vectors, the L2 paradigm can improve the generalization performance of the classifier and reduce overfitting.

3.3. Attention Mechanism

The channel attention mechanism represented by SENet, as shown in Figure 2a, establishes the interrelationship between channels by simply compressing the 2D feature map, which significant affects on model performance, However, SENet is weak at processing location information. CA retains location information by embedding it in channel attention based on SENet, which establishes long-distance dependencies. Channel attention converts the input to a single feature vector by 2D global pooling, as shown in Figure 2b. CA splits channel attention into two one-dimensional feature encodings in different directions, and encodes these to produce two feature maps with enhanced position sensitivity and orientation perception, to enhance the target of interest. The attention block includes compression and excitation to, respectively, collect global information and capture the importance of each channel. The compression operation can be expressed as

$$z_c = \frac{1}{H \times W} = \sum_{i=1}^{H} \sum_{j=1}^{W} x_c(i,j) \tag{4}$$

The excitation operation of the capture channel can be expressed as

$$\hat{X} = X \cdot Sigmoid(T_2(Relu(T_1(z)))) \tag{5}$$

where T_1 and T_2 denote the learnable linear transforms used to capture the importance of each channel.

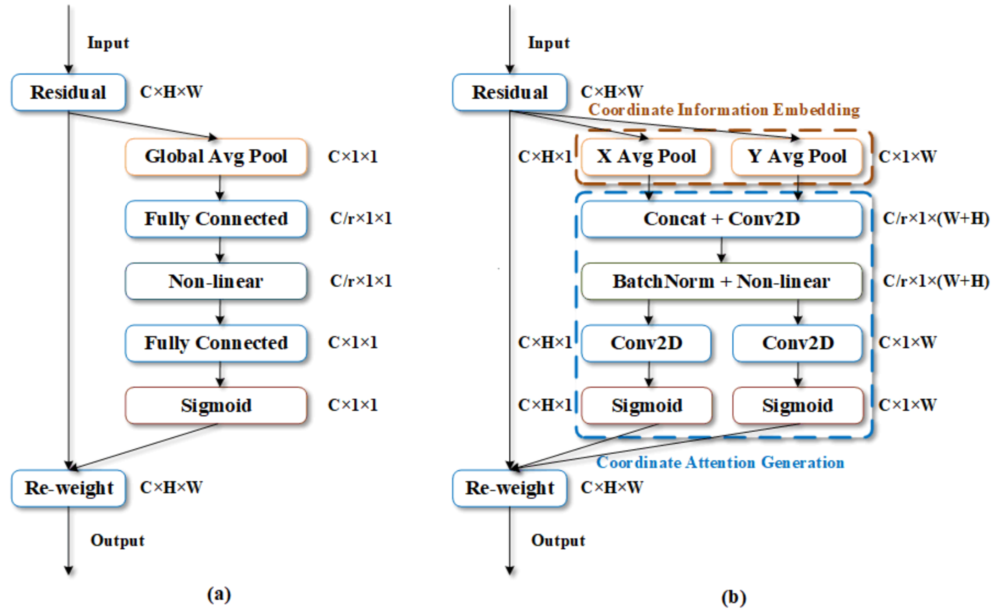


Figure 2. Structure of attention mechanism: (**a**) SENet; (**b**) CA.

Similar to SENet, the CA attention block can be considered a computational with two steps: CIE and CAG. CIE is the critical operation in the attention module to capture precise location information and establish long-term dependencies on spatial orientation. A given X is encoded for each channel along horizontal and vertical coordinates using pooling kernels of dimensions $(H, 1)$ and $(1, W)$, respectively, i.e., the global pooling is transformed to a pair of one-dimensional feature encoding operations. The output of the C-th channel at the height and width of H and W, respectively, must satisfy the following equation:

$$\begin{cases} z_c^h(h) = \frac{1}{W} \sum\limits_{0 \le i < W} x_c(h,i) \\ z_c^w(w) = \frac{1}{H} \sum\limits_{0 \le j < H} x_c(j,w) \end{cases} \tag{6}$$

Equation (6) aggregates feature along two spatial directions through two transformations to generate a pair of directional perception attentional diagrams, which is entirely different from the compression operation in the SE module. One captures long-term dependencies along the spatial direction, and the other direction retains precise positional information, which helps to de-localize objects of interest. CAG uses the location information of the global receptive field generated by CIE to generate an attentional map. CAG must comply with the following requirements:

1. The conversion should be as simple and efficient as possible.
2. The captured position information can be used to precisely locate the region of interest.
3. Relationships between channels can be efficiently captured.

To this end, the first operation of CAG is to cascade the two feature maps generated by the CIE module and transform them using 1 × 1 shared convolution, i.e., F_1 generating $f \in R^{C/r\times(H+W)}$ as an intermediate feature map of spatial information in the horizontal and vertical directions, where r is the downsampling ratio,

$$f = \delta(F_1([z^h, z^w])) \tag{7}$$

where δ is the nonlinear activation function. The second step of CAG divides f into tensors $f^h \in R^{C/r\times H}$ and $f^W \in R^{C/r\times W}$, which are transformed by 1 × 1 convolutions F_h and F_w, respectively, into the same number of channels as the input X, to obtain

$$\begin{cases} g^h(h) = Sigmoid(F_h(f^h)) \\ g^w(w) = Sigmoid(F_w(f^w)) \end{cases} \tag{8}$$

The final CA output is expressed as

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j) \tag{9}$$

4. Experimental Results and Analysis

The effectiveness of image classification is affected by the learning ability of the network and the different regularized loss functions. The learning ability of the network is affected by the complexity of the training dataset, the learning rate, the optimizer, and the activation function, and the loss function of SVM has some impact on image classification.

4.1. Parameter Impact and Selection

4.1.1. Impact and Selection of Datasets and Models

We used the Fashion-MNIST, Cifar10, Cifar100, and Animal10 datasets as benchmarks for the model, and Animal10 and Fish9 [38] as performance tests. Fashion-MNIST is an alternative to MNIST, with 60,000 grayscale images of 28 × 28 resolution, and is widely used to benchmark DL models; Cifar10 is a pervasive object recognition dataset with the same number of 32 × 32 RGB color images. Cifar100 is a 100-class dataset for pervasive object recognition. Animal10 has 26,000 images of 10 species of animals, with a size of approximately 300 × 200; its labels are reviewed manually, and some wrong labels are included to simulate real situations. Fish9 has 9000 images of nine fish species, of size 2832 × 2128. Animal10 and Fish9 images both have size 224 × 224 in this work, to facilitate model training and testing. Table 1 displays information on the processed image dataset.

Table 1. Image dataset parameters.

Dataset	Image Size (Input)	Number
Fashion-MNIST	28 × 28	60,000
Cifar10	32 × 32	60,000
Cifar100	32 × 32	60,000
Animal10	224 × 224	26,000
Fish9	224 × 224	9000

We conducted classification comparison experiments on image datasets using four models: CNN + SVM, CASVM, CNN + softmax, and CNN + CA + softmax. All models were trained 100 times, and the training sample size was N/128.

Since there are two sizes of the dataset used in this paper, 32 × 32 and 224 × 224, two CNN structures were improved for the small-size dataset. In the improved LeNet-based model CNN1, the number of convolutional kernels is increased, the original 5 × 5 kernels are changed to 3 × 3, dropout is added, and CA is embedded in the final convolutional layer. Based on AlexNet, the improved model CNN2 reduces the number of convolutional kernels, embeds CA in the last convolutional layer, and reduces the parameters of the fully connected layer. CNN structures used for our experiments include AlexNet, ResNet18, ResNet152, and VGG19. Table 2 shows the number of parameters (millions float), amount of computation (billions float), and model size (MB) of each CNN model.

Table 2. Number of model parameters and calculation volume.

Model	Params (M)	Flops (G)	Model Size (MB)
CASVM (CNN1)	4.23	0.01	16.12
CNN1	4.23	0.01	16.12
CASVM (CNN2)	7.54	0.21	28.77
CNN2	7.54	0.21	28.75
CASVM (AlexNet)	38.38	0.32	146.41
AlexNet	38.37	0.31	146.38
CASVM (ResNet18)	11.19	54.60	42.70
ResNet18	11.18	54.50	42.66
CASVM (ResNet152)	57.99	365.61	221.23
ResNet152	57.93	365.45	220.99
CASVM (VGG19)	139.67	39.30	532.81
VGG19	139.67	39.30	532.78

4.1.2. Impact and Selection of Loss Function

The loss function affects the learning ability of a network. In this study, four loss functions are used by CNNs for softmax classifiers, of which three are SVM loss, and the other is categorical cross-entropy loss. Among the data loss functions of SVM are hinge loss, squared hinge loss, and categorical hinge loss. Multiple SVM models are formed using one-versus-rest to solve multiple classification problems using hinge loss and squared hinge loss by CNNs. Figure 3 shows the impact of the four loss functions on network training and the results of AlexNet trained on Cifar10.

The experiment shows that SVM is unsuitable for cross-entropy loss, so the accuracy of training results is low, and the loss is high. However, the accuracy of classification using SVM or CASVM is greater than that of softmax, regardless of whether the attention mechanism is embedded, and whether softmax uses cross-entropy loss or hinge loss. Figure 3, shows that the hinge loss has a lower error than cross-entropy loss, and there is some difference in performance between SVM and softmax.

The shallow network CNN1 and the deep network ResNet18 are chosen to compare the performance of the two classifiers without using the embedded attention mechanism. Figure 4 shows the accuracy and loss values obtained by the CNN1 and ResNet18network models, using Animal10, Cifar100, Cifar10, and Fashion-MNIST.

The images of all datasets were compressed to 32 × 32 when training CNN1, and the network model using softmax used cross-entropy loss. From Figure 4a, it is clear that the network model combining CNN1 with SVM had less accuracy improvement on small datasets. Figure 4b, shows that ResNet18 combined with SVM showed some improvement on each dataset, with more obvious improvement on image datasets with 100 classifications like Cifar100 and on the Animal10 large dataset. It can be concluded that the loss function of the SVM classifier shows some improvement in network classification performance.

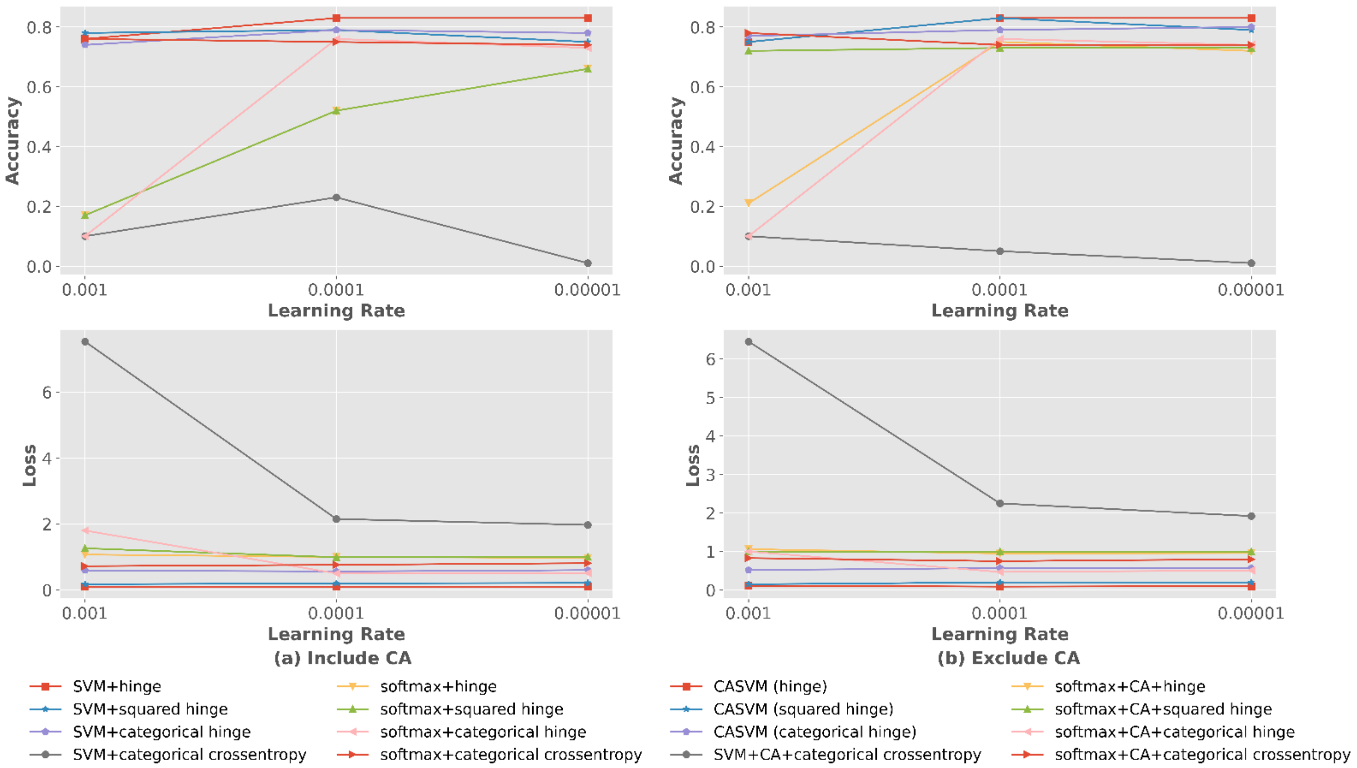


Figure 3. Impact of attention mechanism and different loss functions on classification accuracy.

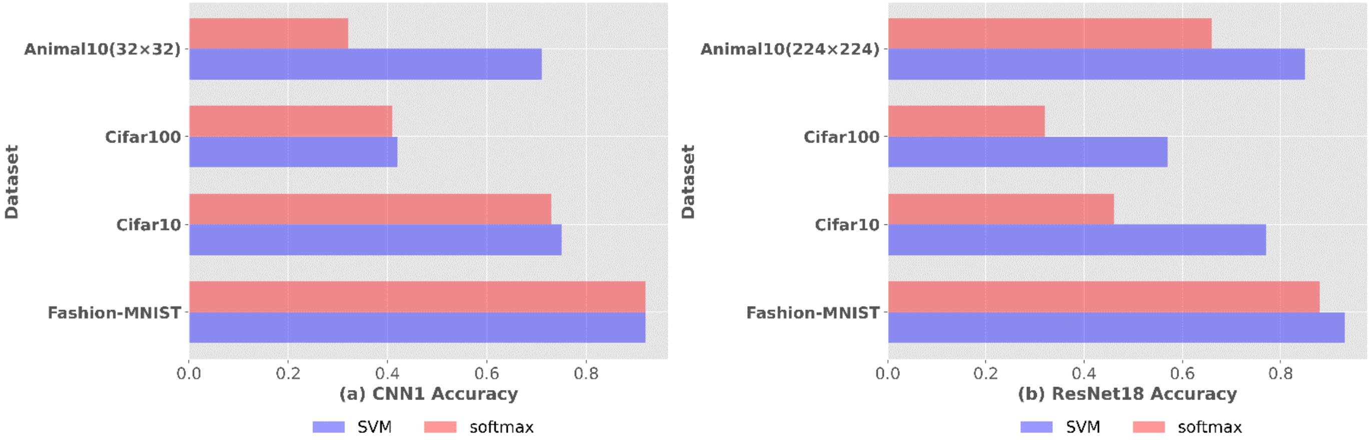


Figure 4. Classification accuracy of two network models on four datasets.

4.1.3. Impact of Learning Rate and Selection

The learning rate significantly impacts the speed and stability of network training. A low learning rate may cause slow convergence, while too high a rate may fail to converge.A higher learning rate may improve the network training speed, but a low learning rate may alleviate training difficulty. We used learning rates of 0.01, 0.001, 0.0001, 0.0001, and 0.00001 in comparison experiments, as shown in Figure 5.

The network model in Figure 5 is based on the CNN1 structure under the CASVM (squared hinge) framework, using Fashion-MNIST for training and testing, to derive the impact on network training at learning rates of 0.01, 0.001, and 0.0001. It is known through the experiment that the test set accuracy is unstable when the learning rate is 0.01, which eventually leads to a significant error. Therefore, we used learning rates of 0.001, 0.0001, and 0.00001 as model comparison parameters.

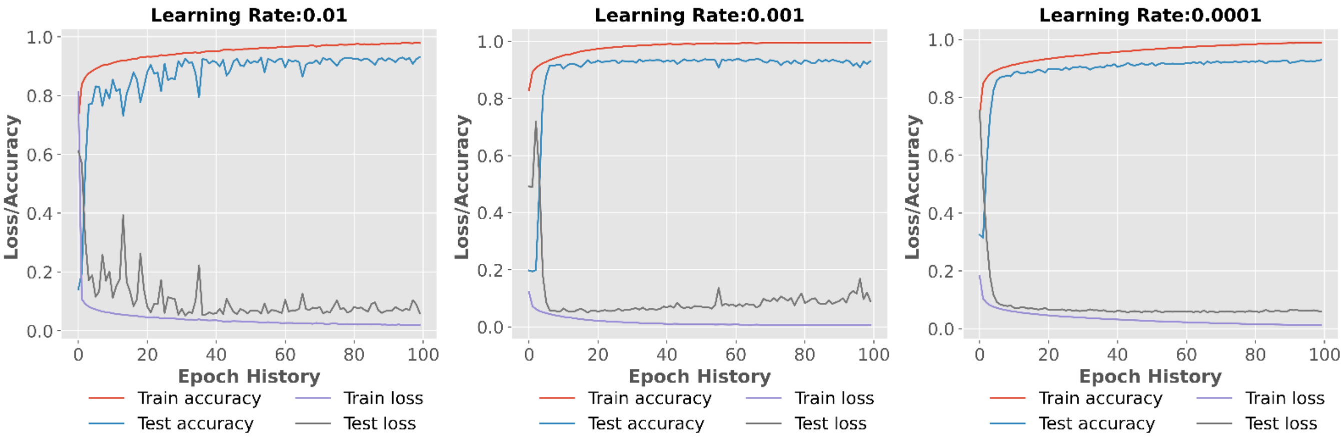


Figure 5. Classification accuracy of test set with training set learning rates of 0.01, 0.001, and 0.0001.

4.1.4. Optimizer Impact and Selection

The optimizer updates and computes network parameters for model training and output. Commonly used optimizers are Adam, SGD, and RMSProp. Adam combines the advantages of Momentum and RMSProp by dynamically adjusting the learning rate of each parameter using first- and second-order moment estimation of the gradient. Compared to SGD and RMSProp, Adam can better constrain each learning rate to a definite range, which makes the parameters smoother.

The impact of different optimizers on training is shown in Table 3, using Animal10 for training, and applying Adam, SGD, and RMSProp optimizers to compare the accuracy and loss values in the CASVM (squared hinge)-based framework. Optimizers were compared using CNN1, AlexNet, CNN2, and ResNet18. Adam outperformed the others in terms of both accuracy and loss, and was chosen as the leading optimizer for CASVM.

Table 3. Effect of different optimizers on training.

Optimizer	Model	Learning Rate (Accuracy)			Learning Rate (Loss)		
		0.001	0.0001	0.00001	0.001	0.0001	0.00001
SGDOptimize	CNN1	0.26	0.30	0.32	0.35	0.34	0.33
	AlexNet	0.38	0.42	0.43	0.32	0.30	0.29
	CNN2	0.51	0.54	0.55	0.27	0.25	0.25
	ResNet18 (32 × 32)	0.34	0.46	0.50	0.32	0.28	0.28
	ResNet18 (224 × 224)	0.36	0.39	0.41	0.32	0.31	0.30
RMSPropOptimize	CNN1	0.68	0.74	0.76	0.20	0.17	0.17
	AlexNet	0.50	0.60	0.59	0.26	0.22	0.24
	CNN2	0.63	0.70	0.76	0.22	0.18	0.21
	ResNet18 (32 × 32)	0.72	0.78	0.81	0.17	0.18	0.20
	ResNet18 (224 × 224)	0.61	0.81	0.81	0.22	0.13	0.14
AdamOptimizer	CNN1	0.74	0.77	0.80	0.17	0.15	0.15
	AlexNet	0.66	0.69	0.73	0.19	0.18	0.16
	CNN2	0.64	0.73	0.77	0.20	0.17	0.15
	ResNet18 (32 × 32)	0.80	0.86	0.86	0.15	0.13	0.15
	ResNet18 (224 × 224)	0.62	0.83	0.89	0.21	0.12	0.09

4.1.5. Impact and Selection of Attentional Mechanisms

The attention mechanism functions to improve image feature extraction. Under the same structure conditions, a network with an attention mechanism can better extract image features. Figure 6 shows the effect of the attention mechanism on training in a network model with different loss functions and two classifiers.

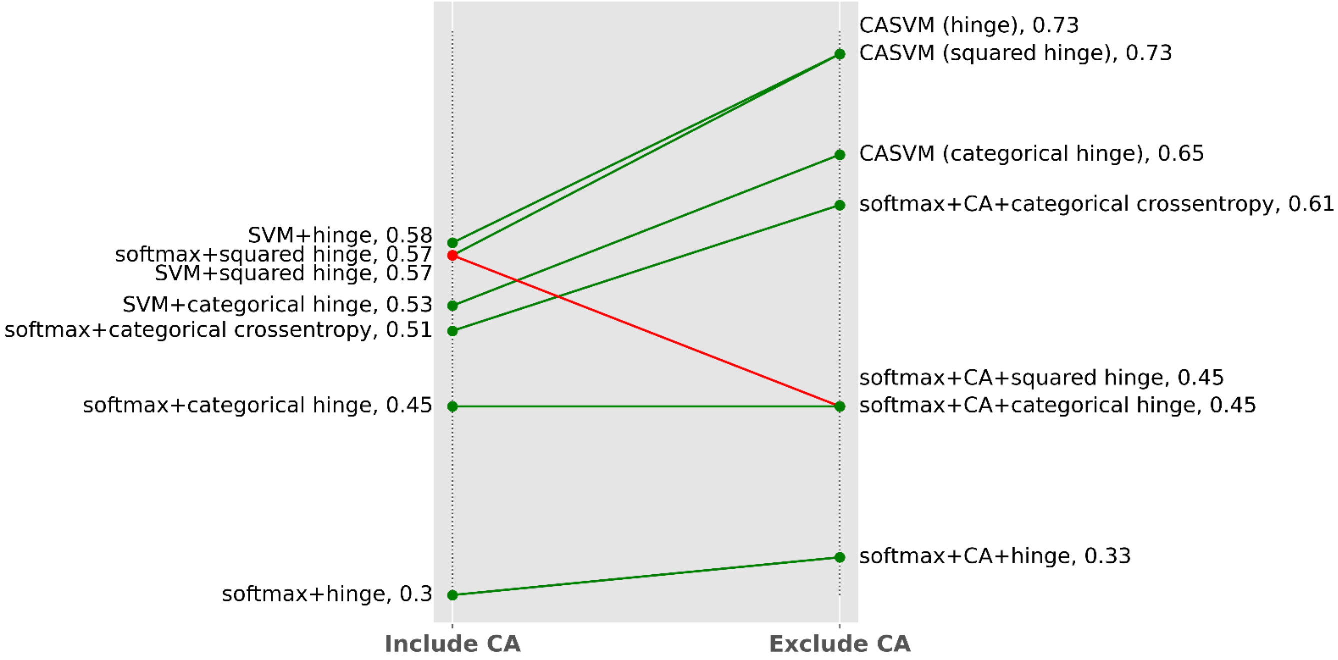


Figure 6. Performance improvement of attention mechanism in network model with different loss functions and two classifiers.

In Figure 6, Animal10 is the dataset for training and testing, and AlexNet is the network model. The network uses hinge and cross-entropy loss functions. As can be seen from the figure, the accuracy of the model improves with the addition of CA, which is denoted by the green line; the reverse is denoted by the red line. With consistent loss functions and CA added, CASVM has higher classification accuracy than softmax. Whether using SVM or softmax classification, accuracy improves with the addition of CA. Hence, attention usually greatly improves the accuracy of image classification.

4.1.6. Impact and Selection of Activation Functions

The activation function affects the learning ability and training speed of a network. Standard activation functions are sigmoid, Tanh, and ReLU. The sigmoid activation function is used in the CA module to compress the values propagated from the upper layers to between [0, 1], which makes the CA module optimally stable and helps in the representation of attentional features. Compared with sigmoid and Tanh, the popular ReLU activation function can alleviate gradient disappearance and accelerate network convergence. Therefore, ReLU is chosen as the primary activation function in the network model in this paper.

4.1.7. Impact and Selection of Other Parameters

The other parameters contain the regularization penalty parameters in the regularized loss in the SVM multiclassification loss function, as shown in Equation (3). The regularization parameter has an important impact on the similar weighting of the data, which can eliminate the ambiguity of the weights and smooth the weight curves. Generally, a smooth weight curve reflects the actual situation; hence, setting the appropriate regularization parameter helps with model training. When the penalty parameter is set low, it will lead to underfitting of the network training, and conversely, it is easy to overfit. Based on comparison experiments, we set the penalty parameter to 0.0001.

4.2. Analysis of Experimental Results

In Section 4.1, the results of the four datasets Anima10, Cifar10, Cifar100, and Fashion-MNIST are compared with the four network models CNN1, AlexNet, CNN2, and ResNet18 combined crosswise between different parameters. The final classification accuracies of the SVM and softmax classifiers are used as benchmarks to compare their effects on the four datasets with different parameter choices. From the results, it is seen that the loss function of the SVM classifier shows some improvement in network classification performance; a lower learning rate of 0.00001 can make the model accuracy smoother; and the Adam optimizer can further improve model accuracy. Therefore, we used the Adam optimizer and a learning rate of 0.00001 as the main parameters of CASVM. Figure 7 compares the image features in CA layers at different locations.

Figure 7 shows the feature output plots of CASVM combined with CNN1. Figure 7a–d show the feature plots obtained from one image randomly selected from Animal10, Fashion-MNIST, Cifar10, and Cifar100, respectively, after training. Both the convolutional and CA layers of the model contain 64 channels, and the figure shows the first convolutional layer. By comparing the output image features of the CA layer at different positions, it is found that the characteristics of channels differ. There are great differences in the roles of different convolution kernels in CNN structures.

Figure 7a shows that the CA layer is located after the first convolutional layer of the network, and it can be observed that the feature difference is not obvious. From this, it can be concluded that when the CA layer is located after the first convolutional layer in a shallow network like CNN1, the output features are still more detailed, and they contain detailed information about the underlying visual features. The third and fourth rows of the feature map in Figure 7a contain some uncorrelated features, which are not critical to image classification, so the CA layer should not be located after the first convolutional layer. As shown in Figure 7b,c, locating the CA layer in the middle of the network enhances the representation of edge contour, global, and texture information. As shown in Figure 7d, a CA layer close to the top layer can weaken invalid features and strengthen effective features, which can improve classification accuracy.

Based on the CNN1 model, CA was introduced to three positions (bottom, middle, and top), and the Animal10 dataset obtained classification accuracies of 67%, 78%, and 80%, respectively. Also, ResNet18 was used to validate it, and the best result of 92% was obtained by embedding CA in the top layer of the residual block. For this purpose, the CA introduction position is set close to the top layer of the network model, except for the network model, which is very shallow.

The most intuitive measure of a model's image classification accuracy is its classification accuracy versus loss value. We performed eight sets of comparison experiments to verify the advantages of CASVM in image classification, with results as shown in Tables 4 and 5, with four basic frameworks: model structure not embedded in CA and based on SVM classification, model structure not embedded in CA and based on softmax classification, model structure embedded in CA and based on SVM classification (CASVM), and model structure embedded in CA and based on softmax classification. The above four frameworks are compared to verify the classification accuracy of CASVM in four different CNN models based on CNN1, AlexNet, CNN2, and ResNet18. To verify the generalization ability of CASVM, the Fashion-MNIST, Cifar10, Anima10, and Cifar100 datasets were selected for training and testing, and images of the Animal10 dataset were preprocessed. Tables 4 and 5 show the results. Softmax is the network model for classification, with cross-entropy loss functions.

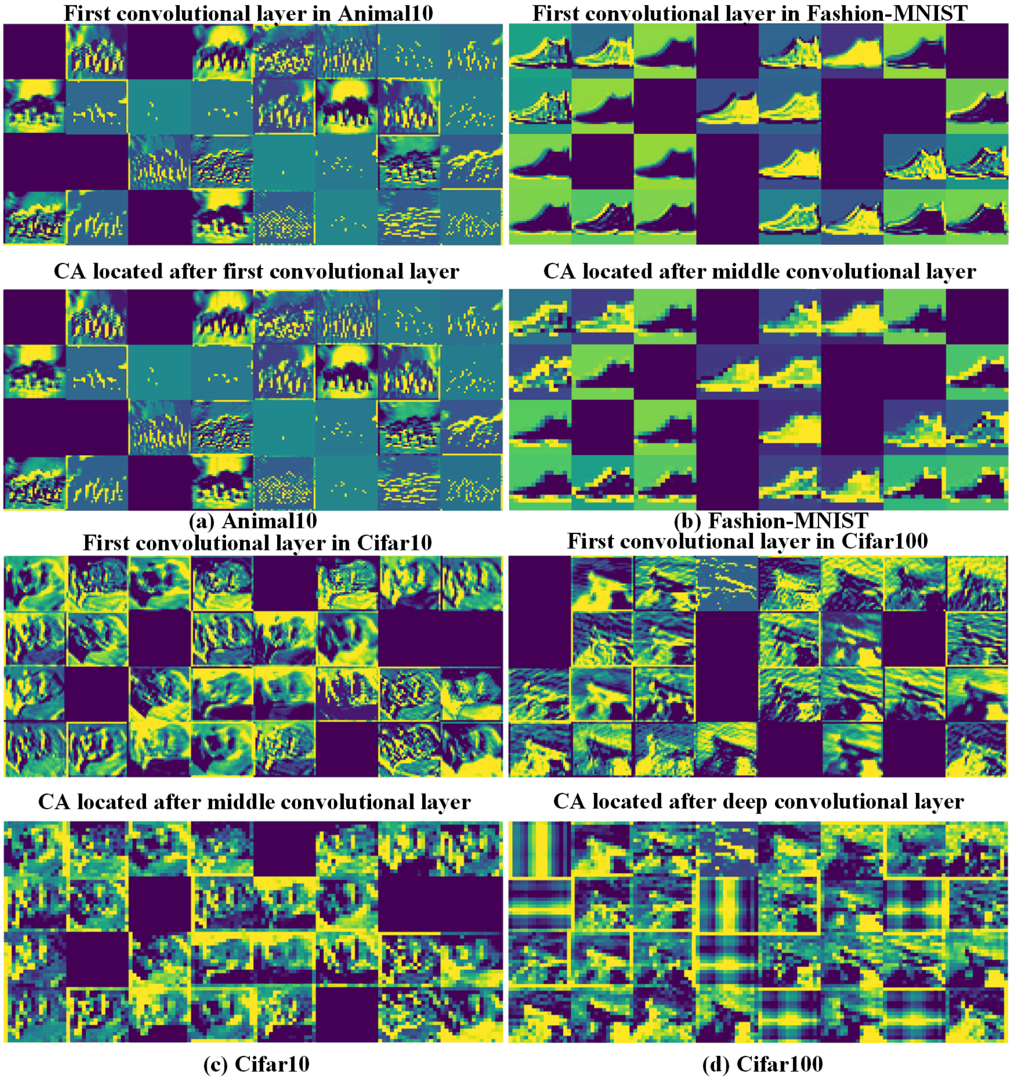

First convolutional layer in Animal10
First convolutional layer in Fashion-MNIST
CA located after first convolutional layer
CA located after middle convolutional layer
(a) Animal10
(b) Fashion-MNIST
First convolutional layer in Cifar10
First convolutional layer in Cifar100
CA located after middle convolutional layer
CA located after deep convolutional layer
(c) Cifar10
(d) Cifar100

Figure 7. Output of feature maps by CA layers in different positions.

As can be seen in Tables 4 and 5, CNN + SVM without CA embedding has five items with the best classification accuracy among the eight sets of image classification metrics. One was obtained when the Cifar10 dataset was selected, i.e., the network model based on ResNet18, with an accuracy of 0.77 when the learning rate was 0.0001. Two were obtained when the Cifar100 dataset was selected, i.e., the network model based on ResNet18, with learning rates of 0.0001 and 0.00001, both with 0.58 accuracy. Two were obtained when the Animal10 dataset was selected, i.e., the network model based on CNN2 with learning rates of 0.0001 and 0.00001 and accuracies of 0.74 and 0.79. The CNN + SVM structure without embedded CA had 12 items with the same accuracy as other structures in the experiment. Among them, four were obtained with the Fashion-MNIST dataset. Values of 0.93 and 0.94 were obtained with learning rates of 0.001 and 0.00001, respectively, with the Resnet18 model; two optimal accuracies, both 0.94, were obtained with a learning rate of 0.0001. Four optimal identical accuracies were obtained when the Cifar10 dataset was selected. When the AlexNet-based network model was selected, accuracies of 0.78, 0.83, and 0.83 were obtained at learning rates of 0.001, 0.0001, and 0.00001, respectively; when the ResNet18-based network model was selected, an accuracy of 0.74 was obtained at the learning rate of 0.001. When the Cifar100 dataset was selected, two optimal identical accuracies of 0.58 were obtained with the ResNet18-based network model. When the Animal10 dataset was used, i.e., the CNN2-based network model, the accuracy was 0.79 at a learning rate of 0.00001.

Table 4. Image classification accuracy and loss values for small sizes.

Dataset	Model	Classifier + Loss Function	Learning Rate (Accuracy)			Learning Rate (Loss)		
			0.001	**0.0001**	**0.00001**	**0.001**	**0.0001**	**0.00001**
Fashion-MNIST	ResNet18	CASVM (hinge)	0.93	0.94	0.94	0.04	0.03	0.03
		CASVM (squared hinge)	0.93	0.94	0.94	0.05	0.06	0.06
		CASVM (categorical hinge)	0.92	0.92	0.93	0.23	0.23	0.24
		SVM + hinge	0.92	0.94	0.94	0.04	0.03	0.03
		SVM + squared hinge	0.92	0.94	0.93	0.06	0.07	0.07
		SVM + categorical hinge	0.93	0.93	0.92	0.18	0.21	0.22
		softmax + CA	0.91	0.91	0.90	0.23	0.28	0.29
		softmax	0.92	0.90	0.88	0.22	0.35	0.35
Cifar10	AlexNet	CASVM (hinge)	0.75	0.83	0.83	0.11	0.09	0.10
		CASVM (squared hinge)	0.75	0.83	0.79	0.15	0.20	0.20
		CASVM (categorical hinge)	0.77	0.79	0.80	0.52	0.57	0.57
		SVM + hinge	0.76	0.83	0.83	0.10	0.10	0.10
		SVM + squared hinge	0.78	0.79	0.75	0.17	0.20	0.21
		SVM + categorical hinge	0.74	0.79	0.78	0.59	0.56	0.60
		softmax + CA	0.78	0.74	0.74	0.83	0.75	0.80
		softmax	0.76	0.75	0.74	0.72	0.76	0.81
	ResNet18	CASVM (hinge)	0.72	0.76	0.79	0.11	0.10	0.09
		CASVM (squared hinge)	0.74	0.76	0.76	0.16	0.17	0.17
		CASVM (categorical hinge)	0.74	0.74	0.74	0.53	0.53	0.54
		SVM + hinge	0.70	0.76	0.77	0.11	0.09	0.09
		SVM + squared hinge	0.74	0.77	0.77	0.15	0.14	0.15
		SVM + categorical hinge	0.71	0.72	0.73	0.52	0.56	0.51
		softmax + CA	0.70	0.62	0.59	0.95	1.10	1.17
		softmax	0.19	0.62	0.48	0.98	0.67	0.94
Cifar100	ResNet18	CASVM (hinge)	0.06	0.40	0.55	0.02	0.02	0.02
		CASVM (squared hinge)	0.52	0.52	0.56	0.03	0.03	0.03
		CASVM (categorical hinge)	0.22	0.50	0.50	0.88	1.41	1.45
		SVM + hinge	0.03	0.57	0.58	0.02	0.02	0.02
		SVM + squared hinge	0.47	0.58	0.57	0.03	0.03	0.03
		SVM + categorical hinge	0.49	0.51	0.53	0.85	0.94	1.17
		softmax + CA	0.43	0.42	0.41	2.21	2.27	2.32
		softmax	0.50	0.35	0.32	1.92	2.59	2.80

The CNN + softmax structure had the best classification accuracy in only one of the eight sets of image classification metrics. The highest classification accuracy of 0.74 was obtained with the Animal10 dataset and the CNN2-based network model with a learning rate of 0.0001. CNN + softmax obtained the same accuracy as other experimental structures in the two items. Among them, one item was obtained by choosing the Cifar10 dataset, i.e., a network model based on AlexNet, with an accuracy of 0.78 at a learning rate of 0.001. One item was obtained with the Animal10 dataset, using a network model based on CNN2, with an accuracy of 0.74 at a learning rate of 0.0001.

It can be seen from Tables 4 and 5 that the CASVM structure has 14 items, in order to obtain the best accuracy in the experiment. With the Cifar10 dataset, an accuracy of 0.79 was obtained at a learning rate of 0.00001 using the ResNet18-based network model. With the Cifar100 dataset, the ResNet18-based network model was chosen to obtain an accuracy of 0.52 at a learning rate of 0.001. The Animal10 dataset was chosen to obtain 12 items. With the CNN1-based network model, accuracies of 0.74 and 0.8 were obtained at learning rates of 0.001 and 0.00001, respectively, and both had an accuracy of 0.77 at a learning rate of 0.0001. With the network model based on AlexNet, accuracies of 0.66 and 0.7 were obtained at learning rates of 0.001 and 0.0001, respectively, and both had an accuracy of 0.73 at a learning rate of 0.00001. With the network model based on CNN2, an accuracy of 0.65 was obtained at a learning rate of 0.001. With the network model based on ResNet18, accuracies of 0.62, 0.84, and 0.92 were obtained at learning rates of 0.001, 0.0001, and 0.00001, respectively. The CASVM structure obtained the same accuracy as other structures in the experiment with 10 items, five on the Fashion-MNIST dataset, with an accuracy of 0.93 at a learning rate of 0.001, and an accuracy of 0.94 was obtained at learning rates of 0.0001 and 0.00001. With the Cifar10 dataset, the network model based on AlexNet was chosen to obtain an optimal accuracy of 0.83 at learning rates of 0.0001 and 0.00001. The network model based on Resnet18 was chosen to obtain an optimal accuracy of 0.74 at learning rates of 0.0001 and 0.001. From the above results, it is clear that CASVM obtained the most optimal classification accuracy for the entire experiment in the Animal10 dataset.

Table 5. Image classification accuracy and loss values for large sizes.

Dataset	Model	Classifier + Loss Function	Learning Rate (Accuracy)			Learning Rate (Loss)		
			0.001	**0.0001**	**0.00001**	**0.001**	**0.0001**	**0.00001**
Animal10	CNN1	CASVM (hinge)	0.63	0.73	0.79	0.16	0.13	0.11
		CASVM (squared hinge)	0.74	0.77	0.80	0.17	0.15	0.15
		CASVM (categorical hinge)	0.69	0.77	0.79	0.66	0.52	0.52
		SVM + hinge	0.44	0.62	0.67	0.20	0.16	0.14
		SVM + squared hinge	0.61	0.65	0.71	0.22	0.20	0.18
		SVM + categorical hinge	0.59	0.64	0.73	0.76	0.71	0.56
		Softmax + CA	0.18	0.55	0.29	2.25	1.31	1.89
		Softmax	0.53	0.41	0.32	1.38	1.70	1.88
	AlexNet	CASVM (hinge)	0.34	0.70	0.73	0.19	0.12	0.11
		CASVM (squared hinge)	0.66	0.69	0.73	0.19	0.18	0.16
		CASVM (categorical hinge)	0.60	0.66	0.65	0.72	0.62	0.67
		SVM + hinge	0.28	0.33	0.58	0.19	0.19	0.15
		SVM + squared hinge	0.37	0.50	0.57	0.31	0.26	0.23
		SVM + categorical hinge	0.19	0.46	0.53	0.97	0.83	0.77
		Softmax + CA	0.62	0.65	0.61	1.11	1.03	1.15
		Softmax	0.64	0.57	0.51	1.11	1.23	1.45
	CNN2	CASVM (hinge)	0.45	0.71	0.77	0.18	0.11	0.10
		CASVM (squared hinge)	0.64	0.73	0.77	0.20	0.17	0.15
		CASVM (categorical hinge)	0.65	0.70	0.76	0.58	0.53	0.50
		SVM + hinge	0.47	0.70	0.75	0.15	0.11	0.10
		SVM + squared hinge	0.62	0.72	0.78	0.20	0.17	0.17
		SVM + categorical hinge	0.63	0.65	0.79	0.61	0.62	0.41
		Softmax + CA	0.62	0.12	0.49	1.10	2.32	1.51
		Softmax	0.67	0.74	0.46	0.99	0.81	1.60
	ResNet18	CASVM (hinge)	0.33	0.81	0.92	0.21	0.10	0.05
		CASVM (squared hinge)	0.62	0.83	0.89	0.21	0.12	0.09
		CASVM (categorical hinge)	0.62	0.84	0.87	0.76	0.39	0.33
		SVM + hinge	0.21	0.22	0.79	0.20	0.21	0.13
		SVM + squared hinge	0.31	0.63	0.85	0.35	0.25	0.11
		SVM + categorical hinge	0.12	0.61	0.85	1.00	0.82	0.37
		Softmax + CA	0.31	0.63	0.58	2.03	1.11	1.22
		Softmax	0.45	0.56	0.66	1.79	1.32	1.03

The lowest loss values of CASVM in Tables 4 and 5 are less than 0.2, and the loss values are lower than 0.05 in the small Fashion-MNIST dataset. CASVM occupies 7 of the 12 lowest loss values for large Animal10 datasets. The loss values of CNN + softmax structure in Animal10 are in the range of 0.99–2.25. Therefore, it can be concluded that CASVM can reduce the loss and better fit the CNN model based on the CASVM structure. Combining the accuracy results in Table 4 with the selection of each parameter in Section 4.1, the main training parameters for the given CASVM are shown in Table 6.

Table 6. Main training parameters of CASVM.

Model	Epoch	Bachsize	Loss	Learning Rate	Optimizer	Activation Function	L2 Penalty Factor
CASVM(CNN)	100	128	Squared Hinge	0.00001	Adam	ReLU	0.0001
CNN	100	128	Categorical Crossentropy	0.00001	Adam	ReLU	\

As shown in Table 7, three CNN models with deeper layers were selected to verify the classification advantages of CASVM on large image datasets. It can be seen that CASVM performed well in terms of accuracy, reaching 99% and 93% on Fish9 and Animal10, respectively. In particular, CASVM (ResNet18) was 26% more accurate than the Resnet18 model. Combined with Table 2, it can be seen that CASVM outperformed the benchmark model in terms of accuracy and loss value, its computational and parametric quantities were close to those of the benchmark model, and the difference in running time was slight, which demonstrates its better robustness and avoidance of significant computational overhead. As the average of accuracy and recall coordination, F1-scores of up to 98% demonstrate the good classification performance of CASVM on large images.

Table 7. Image classification accuracy results.

Dataset	Model	Accuracy	Time	Loss	F1-Score
Fish9	CASVM (ResNet152)	0.99	6157.4	0.04	0.98 ± 0.01
	ResNet152	0.93	5851.3	0.17	0.91 ± 0.02
	CASVM (ResNet18)	0.96	3167.8	0.05	0.93 ± 0.01
	ResNet18	0.91	2697.9	0.16	0.80 ± 0.05
	CASVM (VGG19)	0.96	2789.2	0.14	0.95 ± 0.01
	VGG19	0.91	2057.1	0.32	0.87 ± 0.03
Animal10	CASVM (ResNet152)	0.93	6815.0	0.34	0.90 ± 0.02
	ResNet152	0.76	6101.3	1.07	0.71 ± 0.02
	CASVM (ResNet18)	0.91	3277.3	0.30	0.87 ± 0.02
	ResNet18	0.65	2865.5	1.41	0.45 ± 0.05
	CASVM (VGG19)	0.65	2017.8	0.34	0.62 ± 0.02
	VGG19	0.52	1782.5	1.05	0.49 ± 0.04

To sum up, among the eight groups of image classification indexes, five items of CNN + SVM were best, and 12 items had the same accuracy as other indexes. CNN + softmax had one and two with the same accuracy as other structures. CASVM had 14 optimal, 10 with the same classification accuracy as other structures, and 16 with the lowest loss value. Tables 5 and 6, show that CASVM, using either shallow or deep networks, can be applied to large-size image data to obtain higher classification accuracy. Table 4, shows that CASVM has 10 optimal same precision indexes in small-size datasets, and CASVM in small-size datasets can still improve the model classification accuracy. When using the same CNN model, the classification accuracy and training loss of the combined softmax classification are inferior to those of SVM classification and those of CASVM, which again verifies that CASVM performs better and is more robust than softmax classification in general environments.

5. Discussions

To validate the effectiveness of the model structure proposed in this paper, we compared previous research on combined SVM and CNN models. Table 8 summarizes the performance of these works in 10 training and testing sessions. Following the combination of CNN and SVM for image classification in the literature [3,5], Peng et al. [62] and Chaganti et al. [63] conducted experiments for specific image classification with improved results. Deepak et al. [22] and Zhang et al. [43] have compared the classification results for both SVM and Softmax classifiers and obtained more significant results. Unlike previous works, the CASVM proposed in this paper uses a CA attention mechanism in the feature extraction module and employs a specific loss function. The CASVM framework can be easily ported into CNN-based models. According to our experiments, the highest accuracy is achieved using CASVM for image classification.

Table 8. Experiments with different models.

Model	FLOPs (G)	Accuracy	Training Time (s)	Testing Time (s)
Agarap et al. [5]	1.949	0.77	15.69	0.55
CASVM (Agarap et al.)	1.951	0.83	15.87	0.57
Chaganti et al. [63]	0.339	0.75	14.79	0.48
CASVM (Chaganti et al.)	0.342	0.77	14.92	0.48
Peng et al. [62]	0.928	0.66	15.27	0.54
CASVM (Peng et al.)	0.929	0.67	15.29	0.55
Deepak et al. [22]	0.423	0.83	15.06	0.46
CASVM (Deepak et al.)	0.423	0.84	15.11	0.49
Zhang et al. [43]	0.198	0.65	14.84	0.51
CASVM (Zhang et al.)	0.201	0.79	14.96	0.51

Image classification accuracy is this paper's most critical performance metric, and Figure 8 presents the image classification accuracy obtained in 10 experiments on the Animal10 dataset. 500 samples will be accumulated for each image read in each of the 10 experiments, and 5000 samples will be read for each model in the last experiment. We used the feature extraction network from the previous work as the feature extraction module of CASVM to compare it with the model from the previous work. The relevant experimental parameters are set as in the previous work, and the other parameters are set according to Table 6. Among the models given by Peng et al. [62], we choose the one that achieves the best results. Since the network of Zhang et al. [43] is shallow, the convolutional kernel size and the number of channels were adjusted appropriately to fit the input data. The parameters of the above-tuned models are synchronized to CASVM to ensure the validity of the comparison experiments.

The results are shown in Figure 8 and Table 8. The accuracy and inference time for a single image in Table 8 are the average values after 10 tests, and the training time is the average length of training for one epoch. It can be seen from Figure 8 that the proposed model can maintain higher accuracy compared with other models, and CASVM has higher accuracy than other models in all 10 tests. Combined with Table 8, CASVM does not require much computation, except for a slightly slower training time, and has a higher classification accuracy than the other models. Among them, the highest accuracy improvement is 14%, while the number of parameters is only 0.003G higher, which again verifies that CASVM can obtain better performance with less computation. From the experimental time of each model, CASVM is slower than the other models in terms of training time, with the slowest time being 0.18 s, while the fastest is the same as this model. The inference time for a single image is even more similar. Taken together, the extra training time of CASVM is shorter, while the shorter training time can produce higher image classification accuracy.

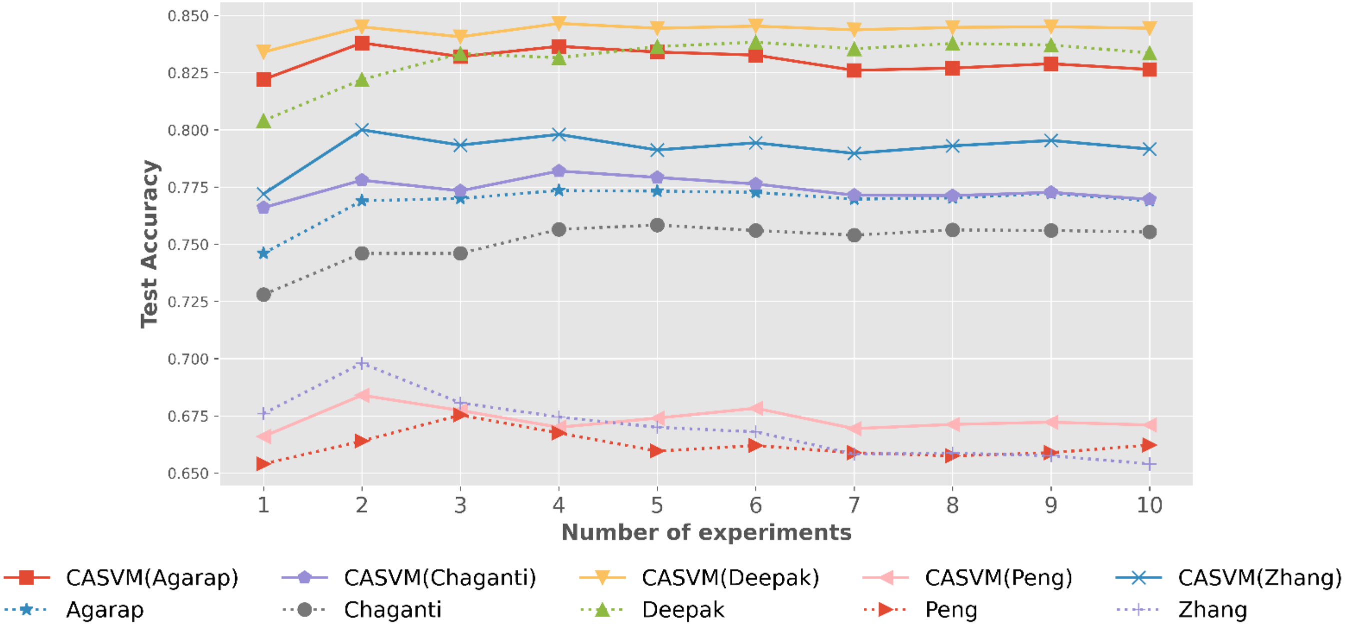
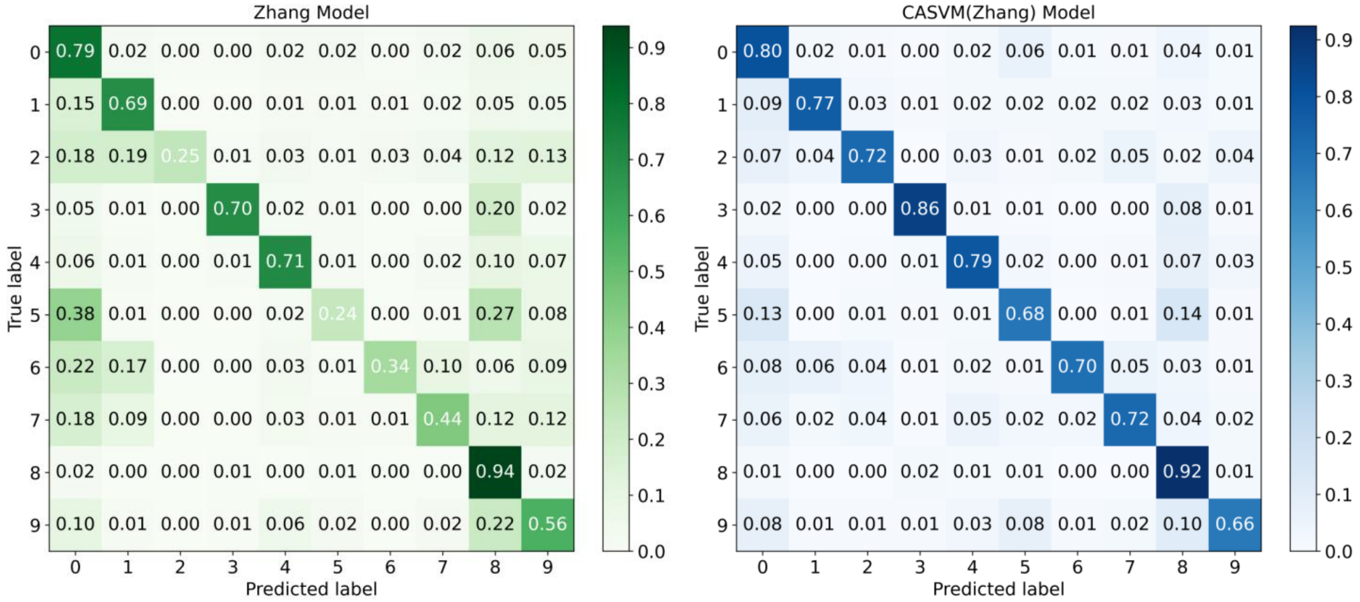


Figure 8. Test accuracy of different models.

Since the Animal10 dataset is unbalanced in terms of distribution, the confusion matrix to evaluate the model with the highest CASVM classification accuracy improvement is given in Figure 9, i.e., the performance results of the model based on Zhang et al. [43] on the dataset. The numbers 0–9 on the axes in the figure represent the ten categories in the dataset. The matrix's diagonal can show the model's recognition accuracy for each category, and both models have better classification results due to the high distribution of data in category 0 and category 8. However, from Figure 9a, it can be seen that the misclassification rate of images is significant.

In the data of classes 2 and 5, where the number of samples is small, the accuracy of the original model shows a significant decrease, and CASVM can maintain the stability of accuracy. For the class 5 image data with the smallest number of samples, the misclassification rate of CASVM is up to 14%, while the original model reaches 38%. In addition, we removed the CA module from CASVM and retrained it according to the configuration in Table 6, and the misclassification rate reached up to 17%. It shows that even without introducing CA to emphasize the features, the misclassification rate is much lower than that of the previous work. It also proves again that CASVM can improve image classification accuracy more effectively.

In Table 8, it can be found that the model proposed by Deepak [22] et al. obtained the highest accuracy rate, comparing it with ResNet18. Table 9 specifies the performance evaluation metrics between the models, including accuracy, recall, and F1 score (F1). As seen from the table, CASVM can achieve better performance than CNN-Softmax when image data is limited, because the hinge loss diverges more slowly than the cross-entropy loss, and because it relies only on support vectors when performing image classification. In addition, CASVM outperforms previous work in three metrics, particularly the F1 score and precision, achieving even 100% precision in each category. We attribute this to using the CA attention mechanism in conjunction with SVM, i.e., emphasizing low-frequency features helps the model to classify more effectively.



	0	1	2	3	4	5	6	7	8	9
0	0.79	0.02	0.00	0.00	0.02	0.02	0.00	0.02	0.06	0.05
1	0.15	0.69	0.00	0.00	0.01	0.01	0.01	0.02	0.05	0.05
2	0.18	0.19	0.25	0.01	0.03	0.01	0.03	0.04	0.12	0.13
3	0.05	0.01	0.00	0.70	0.02	0.01	0.00	0.00	0.20	0.02
4	0.06	0.01	0.00	0.01	0.71	0.01	0.00	0.02	0.10	0.07
5	0.38	0.01	0.00	0.00	0.02	0.24	0.00	0.01	0.27	0.08
6	0.22	0.17	0.00	0.00	0.03	0.01	0.34	0.10	0.06	0.09
7	0.18	0.09	0.00	0.00	0.03	0.01	0.01	0.44	0.12	0.12
8	0.02	0.00	0.00	0.01	0.00	0.01	0.00	0.00	0.94	0.02
9	0.10	0.01	0.00	0.01	0.06	0.02	0.00	0.02	0.22	0.56

(a)

	0	1	2	3	4	5	6	7	8	9
0	0.80	0.02	0.01	0.00	0.02	0.06	0.01	0.01	0.04	0.01
1	0.09	0.77	0.03	0.01	0.02	0.02	0.02	0.02	0.03	0.01
2	0.07	0.04	0.72	0.00	0.03	0.01	0.02	0.05	0.02	0.04
3	0.02	0.00	0.00	0.86	0.01	0.01	0.00	0.00	0.08	0.01
4	0.05	0.00	0.00	0.01	0.79	0.02	0.00	0.01	0.07	0.03
5	0.13	0.00	0.01	0.01	0.01	0.68	0.00	0.01	0.14	0.01
6	0.08	0.06	0.04	0.01	0.02	0.01	0.70	0.05	0.03	0.01
7	0.06	0.02	0.04	0.01	0.05	0.02	0.02	0.72	0.04	0.02
8	0.01	0.00	0.00	0.02	0.01	0.01	0.00	0.00	0.92	0.01
9	0.08	0.01	0.01	0.01	0.03	0.08	0.01	0.02	0.10	0.66

(b)

Figure 9. Confusion matrix: (**a**) The model of Zhang et al. [43]; (**b**) The model of CASVM (Zhang et al.).

Table 9. Performance metrics of different models on the Animal10 dataset.

Relation	**Precision**			**Recall**			**F1**		
	CASVM (Deepak)	**ResNet18**	**CASVM (ResNet18)**	**CASVM (Deepak)**	**ResNet18**	**CASVM (ResNet18)**	**CASVM (Deepak)**	**ResNet18**	**CASVM (ResNet18)**
Cane	0.75	0.75	0.85	0.87	0.71	1.00	0.81	0.73	0.92
Cavallo	0.80	0.71	1.00	0.86	0.77	0.95	0.83	0.74	0.97
Elefante	0.86	0.33	0.99	0.79	0.84	1.00	0.83	0.50	0.99
Farfalla	0.92	0.85	0.93	0.87	0.59	1.00	0.89	0.70	0.94
Gallina	0.87	0.84	1.00	0.84	0.67	0.94	0.86	0.75	0.97
Gatto	0.74	0.77	1.00	0.71	0.74	0.99	0.72	0.76	1.00
Mucca	0.81	0.82	1.00	0.73	0.64	0.96	0.77	0.72	0.98
Pecora	0.86	0.73	0.74	0.70	0.65	0.90	0.77	0.69	0.85
Ragno	0.95	0.75	1.00	0.91	0.83	0.68	0.93	0.79	0.81
Scoiattolo	0.75	0.76	1.00	0.81	0.63	0.80	0.78	0.69	0.89

Table 9 and Figure 9 show that each model has different performance levels for the class 0 Cane, class 7 Pecora, and class 5 Gatto images. From Table 9, for CASVM (ResNet18), cane and pecora have the lowest precision. For this reason, misclassification is further observed using the receiver operating characteristic (ROC), as shown in Figure 10, and compared using Gatto. The area under the curve (AUC) represents the classification ability of the model, and the higher the value, the higher the accuracy. The AUC values of Cane, Pecora, and Gatto are 0.9745, 0.9769, and 0.9933, respectively. The AUC values of cane and pecora are significantly lower than those of gatto. We focused on the image data and excluded the loss of classification accuracy brought on by remarkably similar features. As a result, the unbalanced distribution of the dataset and the lack of samples from this class in the training dataset are to blame for Cane with Pecora's low classification accuracy. Figure 8 and Table 9 show that CASVM can still maintain some classification stability in the presence of subpar image data, despite this.

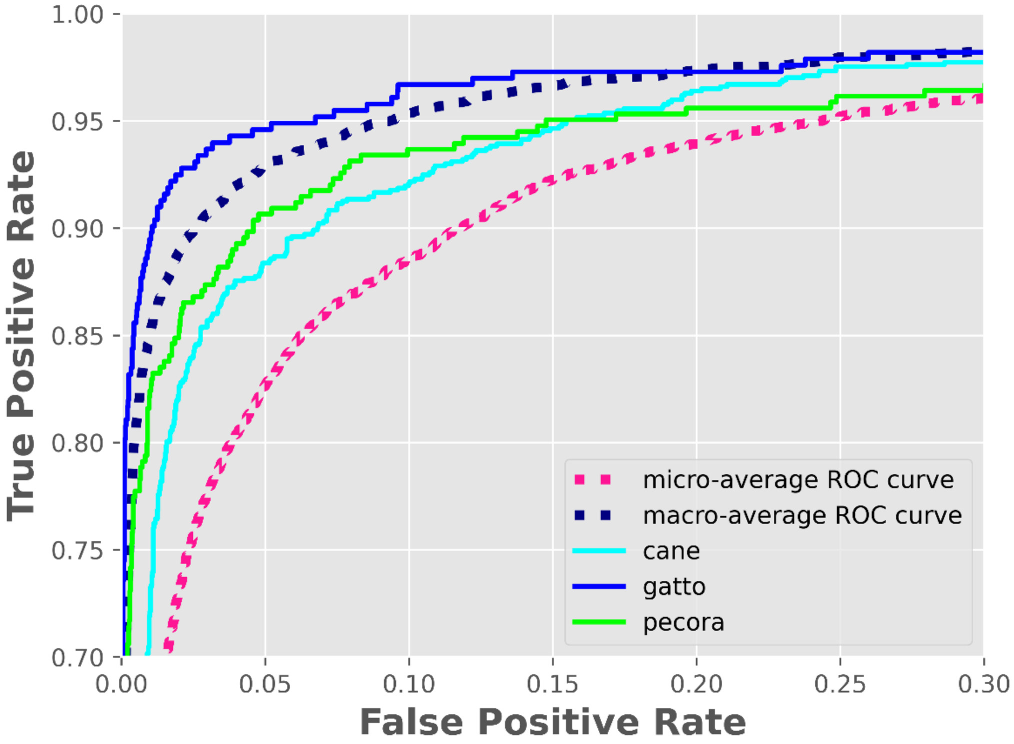


Figure 10. ROC curve of CASVM (ResNet18).

Despite the low computational overhead of CASVM and the more significant image classification results, the CA introduction position in the feature extraction module is determined according to Figure 7 in an experiment-oriented manner and, therefore, still lacks interpretation. In addition, this paper can also be tested using datasets with more than a hundred image categories, like Cifar100, but applying CASVM also requires modules with more robust feature capability extraction.

6. Conclusions

The proposed image classification method based on the fusion of DL and SVM has three parts:

1. We implement a combination of SVM and CNN, specifically replacing the softmax classifiers commonly used in CNNs with SVMs and training the network model using hinge loss function backpropagation to improve the generalization ability of image classification. As shown in Tables 4 and 6, CASVM has a higher accuracy and better classification performance in a general setting compared to softmax classification.
2. Based on this model, a CA mechanism is introduced to enhance network feature representation by aggregating features for two spatial directions to capture location information and channel relationships with low computational overhead. As shown in Table 7, the training time of CASVM is small compared with the benchmark model, and the classification accuracy is high. As shown in Tables 8 and 9, the model performance of CASVM is stable and robust compared to previous work.
3. The CASVM model is proposed to implement image classification, which uses the advantages of CNNs and SVMs and compensates for their disadvantages. CNNs using the backpropagation algorithm are prone to fall into local minima, and SVM can effectively avoid falling into local optima. However, if SVM alone is used for classification, it requires more complicated steps. CNNs can extract representative image features by convolving and downsampling the image. Therefore, the accuracy of image classification can be further improved by using CNNs to extract image features, and replacing the CNN classification layer with SVM, with excellent generalization ability.

Experiments showed that CASVM can effectively improve image classification accuracy. The higher F1-score of CASVM in different models not only improves the feature representation performance but also provides network models in different device environments to provide smaller computational overhead. With small overhead and good performance, CASVM provides a model reference for researchers and provides new options for research work on image classification, recognition, and segmentation with limited equipment. Although CASVM has high classification accuracy, many parameters require manual definition, the amount of training in this paper is fixed, and the numbers of deep network models and large-size datasets are insufficient. In addition, the interpretation of the specific position of CA in the network introduced in CASVM is not clear enough, and the appropriate position is derived by embedding CA in different positions in experiments. Our future work will focus on parameter adaption and a deeper network to optimize the algorithm, as well as improving the interpretation of CA position embedding to further improve classification accuracy.

Author Contributions: Conceptualization, J.P. and S.T.; methodology, J.P.; software, J.P.; validation, S.T., Y.L. and J.Z.; formal analysis, J.P.; investigation, J.P.; resources, S.T.; data curation, J.P.; writing—original draft preparation, J.P.; writing—review and editing, S.T.; visualization, J.P.; supervision, S.T.; funding acquisition, S.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Science and Technology Research Project (Youth) of Chongqing Municipal Education Commission (Project No. KJQN202101149).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
2. Kumar, B.; Vyas, O.P.; Vyas, R. A comprehensive review on the variants of support vector machines. *Mod. Phys. Lett. B* **2019**, *33*, 1950303. [CrossRef]
3. Tang, Y. Deep Learning using Linear Support Vector Machines. *arXiv* **2013**, arXiv:1306.0239. [CrossRef]
4. Agarap, A.F.M. A Neural Network Architecture Combining Gated Recurrent Unit (GRU) and Support Vector Machine (SVM) for Intrusion Detection in Network Traffic Data. In Proceedings of the 2018 10th International Conference on Machine Learning and Computing, Macau, China, 26–28 February 2018.
5. Agarap, A.F. An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification. *arXiv* **2017**, arXiv:1712.03541. [CrossRef]
6. Deng, L.; Yu, D. Deep learning: Methods and applications. *Found. Trends®Signal Process.* **2014**, *7*, 197–387. [CrossRef]
7. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 84–90. [CrossRef]
8. Qi, X.; Wang, T.; Liu, J. Comparison of support vector machine and softmax classifiers in computer vision. In Proceedings of the 2017 Second International Conference on Mechanical, Control and Computer Engineering (ICMCCE), Harbin, China, 8–10 December 2017; pp. 151–155.
9. Chandra, M.A.; Bedi, S.S. Survey on SVM and their application in image classification. *Int. J. Inf. Technol.* **2021**, *13*, 1–11. [CrossRef]
10. Baldomero-Naranjo, M.; Martínez-Merino, L.I.; Rodríguez-Chía, A.M. A robust SVM-based approach with feature selection and outliers detection for classification problems. *Expert Syst. Appl.* **2021**, *178*, 115017. [CrossRef]
11. Thillaikkarasi, R.; Saravanan, S. An Enhancement of Deep Learning Algorithm for Brain Tumor Segmentation Using Kernel Based CNN with M-SVM. *J. Med. Syst.* **2019**, *43*, 84. [CrossRef]
12. Nguyen, Q.H.; Nguyen, B.P.; Nguyen, T.B.; Do, T.T.T.; Mbinta, J.F.; Simpson, C.R. Stacking segment-based CNN with SVM for recognition of atrial fibrillation from single-lead ECG recordings. *Biomed. Signal Process. Control.* **2021**, *68*, 102672. [CrossRef]
13. Nanglia, P.; Kumar, S.; Mahajan, A.N.; Singh, P.; Rathee, D. A hybrid algorithm for lung cancer classification using SVM and Neural Networks. *ICT Express* **2021**, *7*, 335–341. [CrossRef]
14. Sun, L.; Zou, B.; Fu, S.; Chen, J.; Wang, F. Speech emotion recognition based on DNN-decision tree SVM model. *Speech Commun.* **2019**, *115*, 29–37. [CrossRef]

15. Khairandish, M.O.; Sharma, M.; Jain, V.; Chatterjee, J.M.; Jhanjhi, N.Z. A Hybrid CNN-SVM Threshold Segmentation Approach for Tumor Detection and Classification of MRI Brain Images. *IRBM* **2021**, *43*, 290–299. [CrossRef]
16. Gong, W.; Chen, H.; Zhang, Z.; Zhang, M.; Wang, R.; Guan, C.; Wang, Q. A Novel Deep Learning Method for Intelligent Fault Diagnosis of Rotating Machinery Based on Improved CNN-SVM and Multichannel Data Fusion. *Sensors* **2019**, *19*, 1693. [CrossRef]
17. Chlaoua, R.; Meraoumia, A.; Aiadi, K.E.; Korichi, M. Deep learning for finger-knuckle-print identification system based on PCANet and SVM classifier. *Evol. Syst.* **2019**, *10*, 261–272. [CrossRef]
18. Barua, P.D.; Baygin, N.; Dogan, S.; Baygin, M.; Arunkumar, N.; Fujita, H.; Tuncer, T.; Tan, R.-S.; Palmer, E.; Azizan, M.M.B.; et al. Automated detection of pain levels using deep feature extraction from shutter blinds-based dynamic-sized horizontal patches with facial images. *Sci. Rep.* **2022**, *12*, 17297. [CrossRef] [PubMed]
19. Guo, S.; Chen, S.; Li, Y. Face recognition based on convolutional neural network and support vector machine. In Proceedings of the 2016 IEEE International conference on Information and Automation (ICIA), Ningbo, China, 1–3 August 2016; pp. 1787–1792.
20. Baygin, M.; Yaman, O.; Barua, P.D.; Dogan, S.; Tuncer, T.; Acharya, U.R. Exemplar Darknet19 feature generation technique for automated kidney stone detection with coronal CT images. *Artif. Intell. Med.* **2022**, *127*, 102274. [CrossRef] [PubMed]
21. Kaur, P.; Singh, G.; Kaur, P. Intellectual detection and validation of automated mammogram breast cancer images by multi-class SVM using deep learning classification. *Inform. Med. Unlocked* **2019**, *16*, 100151. [CrossRef]
22. Deepak, S.; Ameer, P.M. Automated Categorization of Brain Tumor from MRI Using CNN features and SVM. *J. Ambient Intell. Humaniz. Comput.* **2021**, *12*, 8357–8369. [CrossRef]
23. Kaplan, E.; Chan, W.Y.; Dogan, S.; Barua, P.D.; Bulut, H.T.; Tuncer, T.; Cizik, M.; Tan, R.-S.; Acharya, U.R. Automated BI-RADS classification of lesions using pyramid triple deep feature generator technique on breast ultrasound images. *Med. Eng. Phys.* **2022**, *108*, 103895. [CrossRef]
24. Li, Y.; Li, J.; Pan, J.-S. Hyperspectral image recognition using SVM combined deep learning. *J. Internet Technol.* **2019**, *20*, 851–859.
25. Okwuashi, O.; Ndehedehe, C.E. Deep support vector machine for hyperspectral image classification. *Pattern Recognit.* **2020**, *103*, 107298. [CrossRef]
26. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2014; Volume 27.
27. Mnih, V.; Heess, N.; Graves, A.; Kavukcuoglu, K. Recurrent Models of Visual Attention. In *Advances in Neural Information Processing Systems*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2014; Volume 27.
28. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [CrossRef]
29. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
30. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
31. Gao, Z.; Xie, J.; Wang, Q.; Li, P. Global second-order pooling convolutional networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3024–3033.
32. Liu, J.-J.; Hou, Q.; Cheng, M.-M.; Wang, C.; Feng, J. Improving convolutional networks with self-calibrated convolutions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10096–10105.
33. Chen, Y.; Kalantidis, Y.; Li, J.; Yan, S.; Feng, J. A^2-Nets: Double Attention Networks. In *Advances in Neural Information Processing Systems*; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2018; Volume 31.
34. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
35. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
36. Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. *arXiv* **2019**, arXiv:1905.02244. [CrossRef]
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
38. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning Transferable Architectures for Scalable Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 8697–8710.
39. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv* **2019**, arXiv:1905.11946. [CrossRef]
40. Yavartanoo, M.; Hung, S.-H.; Neshatavar, R.; Zhang, Y.; Lee, K.M. PolyNet: Polynomial Neural Network for 3D Shape Recognition with PolyShape Representation. In Proceedings of the 2021 International Conference on 3D Vision (3DV), London, UK, 1–3 December 2021; pp. 1014–1023.

41. Passricha, V.; Aggarwal, R.K. Convolutional support vector machines for speech recognition. *Int. J. Speech Technol.* **2019**, *22*, 601–609. [CrossRef]
42. Fan, J.; Lee, J.; Lee, Y. A Transfer Learning Architecture Based on a Support Vector Machine for Histopathology Image Classification. *Appl. Sci.* **2021**, *11*, 6380. [CrossRef]
43. Zhang, X.; Zhang, M.; Xiang, Z.; Mo, J. Research on diagnosis algorithm of mechanical equipment brake friction fault based on MCNN-SVM. *Measurement* **2021**, *186*, 110065. [CrossRef]
44. Franc, V.; Hlavac, V. Multi-class support vector machine. In Proceedings of the 2002 International Conference on Pattern Recognition, Quebec City, QC, Canada, 11–15 August 2002; Volume 2, pp. 236–239.
45. Dogan, S.; Datta Barua, P.; Kutlu, H.; Baygin, M.; Fujita, H.; Tuncer, T.; Acharya, U.R. Automated accurate fire detection system using ensemble pretrained residual network. *Expert Syst. Appl.* **2022**, *203*, 117407. [CrossRef]
46. Duan, Y.; Zou, B.; Xu, J.; Chen, F.; Wei, J.; Tang, Y.Y. OAA-SVM-MS: A fast and efficient multi-class classification algorithm. *Neurocomputing* **2021**, *454*, 448–460. [CrossRef]
47. Gao, Z.; Fang, S.-C.; Gao, X.; Luo, J.; Medhin, N. A novel kernel-free least squares twin support vector machine for fast and accurate multi-class classification. *Knowl.-Based Syst.* **2021**, *226*, 107123. [CrossRef]
48. Deng, Y.; Deng, Y. A Method of SAR Image Automatic Target Recognition Based on Convolution Auto-Encode and Support Vector Machine. *Remote Sens.* **2022**, *14*, 5559. [CrossRef]
49. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *arXiv* **2020**, arXiv:2010.11929. [CrossRef]
50. Ryoo, M.S.; Piergiovanni, A.; Arnab, A.; Dehghani, M.; Angelova, A. TokenLearner: What Can 8 Learned Tokens Do for Images and Videos? *arXiv* **2021**, arXiv:2106.11297. [CrossRef]
51. Wortsman, M.; Ilharco, G.; Gadre, S.Y.; Roelofs, R.; Gontijo-Lopes, R.; Morcos, A.S.; Namkoong, H.; Farhadi, A.; Carmon, Y.; Kornblith, S. Model soups: Averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In Proceedings of the International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; pp. 23965–23998.
52. Zhang, Q.; Xu, Y.; Zhang, J.; Tao, D. ViTAEv2: Vision Transformer Advanced by Exploring Inductive Bias for Image Recognition and Beyond. *arXiv* **2022**, arXiv:2202.10108. [CrossRef]
53. Huang, T.; Huang, L.; You, S.; Wang, F.; Qian, C.; Xu, C. LightViT: Towards Light-Weight Convolution-Free Vision Transformers. *arXiv* **2022**, arXiv:2207.05557. [CrossRef]
54. Chen, L.; Zhang, H.; Xiao, J.; Nie, L.; Shao, J.; Liu, W.; Chua, T.-S. SCA-CNN: Spatial and Channel-Wise Attention in Convolutional Networks for Image Captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
55. Lee, H.; Kim, H.-E.; Nam, H. SRM: A Style-Based Recalibration Module for Convolutional Neural Networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
56. Qin, Z.; Zhang, P.; Wu, F.; Li, X. FcaNet: Frequency Channel Attention Networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 10–17 October 2021; pp. 783–792.
57. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. *arXiv* **2020**, arXiv:1910.03151. [CrossRef]
58. Jaderberg, M.; Simonyan, K.; Zisserman, A.; Kavukcuoglu, K. Spatial Transformer Networks. In *Advances in Neural Information Processing Systems*; Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2015; Volume 28.
59. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Vedaldi, A. Gather-Excite: Exploiting Feature Context in Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2018; Volume 31.
60. Zhao, H.; Zhang, Y.; Liu, S.; Shi, J.; Loy, C.C.; Lin, D.; Jia, J. PSANet: Point-wise Spatial Attention Network for Scene Parsing. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
61. Zhang, Z.; Lan, C.; Zeng, W.; Jin, X.; Chen, Z. Relation-Aware Global Attention for Person Re-Identification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
62. Peng, Y.; Liao, M.; Deng, H.; Ao, L.; Song, Y.; Huang, W.; Hua, J. CNN–SVM: A classification method for fruit fly image with the complex background. *IET Cyber-Phys. Syst. Theory Appl.* **2020**, *5*, 181–185. [CrossRef]
63. Chaganti, S.Y.; Nanda, I.; Pandi, K.R.; Prudhvith, T.G.N.R.S.N.; Kumar, N. Image Classification using SVM and CNN. In Proceedings of the 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA), Gunupur, India, 13–14 March 2020; pp. 1–5.