Applied Intelligence
https://doi.org/10.1007/s10489-021-02211-x

Mixed attention dense network for sketch classification

Ming Zhu [1,2] · Chun Chen [2] · Nian Wang [1,2] · Jun Tang [1,2] · Chen Zhao [2]

Accepted: 11 January 2021

Abstract
State-of-the-art convolutional neural networks (CNNs) on sketch classification cannot balance the expression ability of final feature vectors and the problems of gradient vanishing and network degradation. In order to improve the classification accuracy, we design a mixed attention dense network for sketch classification. According to the sparse characteristics of the sketch, this network uses overlapping pooling of a large size. In addition, dense blocks are added on the top of the middle convolutional layers to achieve feature reuse. Specifically, in order to extract more representative local, detail information, mixed attention is applied in the dense blocks. Finally, the center loss is combined with the softmax cross entropy loss to improve the classification accuracy. Through experiments, we compare our model with several state-of-the-art methods on the TU-Berlin dataset, and the experimental results demonstrate the effectiveness of our model.

Keywords Convolutional neural networks · Sketch classification · Dense network · Mixed attention

1 Introduction

With the massive growth of image data on the internet, image classification has become one of the research hotspots in the field of computer vision. Sketch classification [1–3], as an important branch of image classification, has also received wide attention from researchers. In recent years, with the popularization of various touch devices and the development of deep learning, the research in sketch classification has also made great progress.

Unlike images, sketches are composed only of sparse lines. Since different people have different painting habits, even the same object may be depicted in completely different styles. Some people's painting styles are relatively simple, so that they only draw the most representative parts, while other people are used to depicting all of the details clearly, which leads to different degrees of abstraction and the deformation of the sketch. In addition, the lack of texture information makes it difficult to distinguish two sketches. Thus, sketch classification is a great challenge.

Most of the existing methods based on CNNs simply use several convolutional layers and fully connected layers [4, 5]. Although good results have been achieved occasionally, due to the shallowness of the network, the fitting ability is often insufficient when the datasets are large, and the network does not make use of the rich features in the middle convolutional layers. This means that the expression ability of final feature vectors is not sufficient, and the sketch classification accuracy is usually not high. We hope to use a deeper network to extract more discriminative features and improve the sketch classification accuracy. However, if we just stack more convolutional layers to increase the depth of the network, problems such as gradient vanishing and network degradation may occur. There are some networks that can effectively alleviate these problems, such as the residual network [6] and the dense network [7]. The residual network connects the deep layer and the shallow layer through a shortcut connection, so that the gradient can flow to the shallow network, while the dense network connects the convolution layers of different layers in the channel direction. Compared with the residual network, the dense network integrates the characteristics of convolutional layers at various levels; it not only has a smaller number of parameters than the residual network, it also strengthens the reuse of features.

For the above reasons, we design an attention-dense-sketch network with a dense network and mixed attention to extract more discriminative features and improve the sketch

✉ Nian Wang
wn_xlb@ahu.edu.cn

[1] National Engineering Research Center for Agro-Ecological Big Data Analysis & Application, Anhui University, Hefei 230601, China

[2] School of Electronics and Information Engineering, Anhui University, Hefei, China

Published online: 04 March 2021

classification accuracy. The contributions of this paper are summarized as follows:

(1). By combining the features of the sketch with the dense network, the information from the middle convolution layer is utilized better.
(2). Mixed attention is embedded in a dense network to extract more discriminative features.
(3). The center loss is combined with the softmax cross entropy loss to increase the inter-class distance while reducing the intra-class distance to improve the classification accuracy.

The rest of this paper is organized as follows. We briefly review some related work in Section 2. Section 3 describes the sketch classification method we used in detail. In Section 4, we report and analyze the experimental results on the TU-Berlin dataset. The conclusions and discussion are given in Section 5.

2 Related work

Sketches have been widely studied by researchers due to their intuitiveness and simplicity. As early as the 1960s, a computer program called SketchPad [8] appeared to realize human-computer interaction. There are many sketch research directions, such as sketch classification [1–3], sketch retrieval [9–12] and sketch segmentation [13–16]. However, all these works face a common problem, which is the lack of datasets. Eitz et al. [1] collected the first large-scale sketch dataset, TU-Berlin. Each sketch was completed by a non-professional painter. The research on this dataset reported that the average recognition accuracy of humans is 73.1%. This dataset is also the benchmark dataset commonly used in this field.

From the perspective of feature extraction methods, sketch research can be divided into two categories: methods based on manual feature extraction and methods based on deep learning.

There are many manually extracted features, such as SIFT features, HOG features and shape context features. Although these features can improve the effective description of sketches from various angles, they cannot be used to achieve the recognition accuracy of humans. Meanwhile, manual feature extraction not only costs a lot of manpower and material resources, it also has higher experience requirements.

With the rapid development of deep learning, various deep neural networks (DNNs) have achieved remarkable results in the field of image recognition, especially CNNs, which are widely used to extract the features of sketches.

Inspired by the success of CNNs in computer vision, many methods have been proposed for sketch classification in recent years. Yu et al. [4] developed a network named sketch-a-net, which uses a larger convolution kernel and overlapping pooling size to divide a sketch into multiple parts to realize a multi-channel CNN. Experiments on the dataset TU-Berlin showed that the recognition accuracy of this method exceeded the recognition accuracy of humans. Yu et al. [5] proposed an improved sketch classification network, "sketch-a-net2", which designed several data augmentation methods suitable for sketches and applied integrated fusion and pre-training strategies to improve recognition performance. Sarvadevabhatla et al. [17] introduced a sketch recognition framework based on deep features by using two classic CNN structures: the ImageNet CNN and an improved LeNet CNN. In [18], a deep neural network model (sketch-DNN) for sketch classification was trained with a large convolution kernel. Sert et al. [19] proposed a freehand sketch recognition scheme based on the feature-level fusion of Convolutional Neural Networks (CNNs) in a transfer learning context.

Sketch images usually have large-scale visual variations caused by drawing styles or viewpoints, which makes it difficult to develop generalized representations using the fixed computational mode of the convolutional kernel. Thus, Zhang et al. [20] employed an architecture to dynamically discover object landmarks and learn discriminative structural representations to address the problem of the fixed computational mode in the feature extraction process without extra supervision. Shi et al. [21] presented an improved deformable convolutional neural network for recognizing sketches, such that the network can identify the deformation of a sketch to achieve higher accuracy on a sketch dataset. Zhang et al. [22] proposed a cousin network to transfer the knowledge of a network learned from natural images to a sketch network by extracting more relevant features. He et al. [23] introduced a Deep Visual-Sequential Fusion (DVSF) model to obtain the visual features and the sequential features of a sketch. Zhang et al. [24] exploited a Hybrid CNN network composed of A-Net and S-Net to describe appearance information and shape information. Kong et al. [25] proposed a lightweight neural network architecture with depth-wise separable convolution to reduce parameters and adjust the network effectively for the sparsity of a sketch.

The proposed network is essentially different from networks in previous papers. It is a deeper network, extracting the sketch characteristics with the dense network and mixed attention, and it achieves better results than previous networks with simple data augmentation without pre-training.

3 Methods

In this section, we introduce the proposed mixed attention dense network for sketch classification. The network structure is illustrated in Fig. 1.

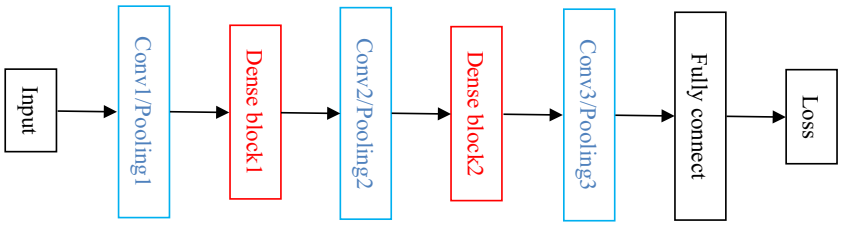


Fig. 1 Network structure

The proposed network is a single-branch network with two dense blocks. Three convolution layers and pooling layers are interlaced with the two dense blocks. Pooling layer 3 is followed by three fully connected layers. The overall loss function is the combination of the softmax cross entropy loss and the center loss. Since the sketches are composed of sparse lines, there is no information in most positions; therefore, a larger convolution kernel is used in the first convolution layer. According to the research in [26], we know that overlapping pool size can alleviate overfitting and improve prediction accuracy to a certain extent, so we apply a 7×7 pooling.

3.1 Dense blocks and mixed attention

Existing CNN-based sketch classification models mostly use several convolution layers and several fully connected layers. It is well known that a deeper network can be designed to extract more discriminative features. However, if only a few convolution layers are added, the effect may not be improved by changing the convolution parameters. Sometimes, the network's performance will worsen due to gradient vanishing. In order to facilitate the backpropagation of the gradient and make better use of the features of each layer in the training stage, a dense network [7] is used to realize feature reuse and improve efficiency by taking the output of each layer as the input of the following layer.

The structure of the dense block employed in the proposed network is illustrated in Fig. 2. The input of each layer is the concatenation of the output of all the previous layers. In order to ensure that the size of the feature map is consistent, only a convolution layer is used in the dense block, rather than a pooling layer. For simplicity, Fig. 2 shows only the structure of the three-layer dense block. However, in the experiment, the eight-layer dense block is used.

In order to extract more useful features, soft attention is applied to the dense blocks. From the perspective of the attention domain, soft attention can be divided into three types: spatial attention, channel attention and mixed attention. Spatial attention treats the features in each channel equally, ignoring the importance of different channels, while channel attention performs global average pooling or global maximum pooling on the feature map of each channel, ignoring the local information of each feature map. We use a mixed attention block, which is able to achieve the effects of both spatial attention and channel attention. For an input three-dimensional convolution feature map $F \in \Re^{H \times W \times C}$, a convolution layer with a convolution kernel of 1×1 is used for convolution, where the number of filters is the same as the number of input channels, which is C. Then, the sigmoid function for each position is utilized to compute the weight of each position:

$$M = \frac{1}{1 + \exp(-x_{i,c})} \tag{1}$$

where M is the learned mask, and $x_{i,c}$ represents the convolution feature at the spatial position i and the channel c.

The specific parameters are shown in Table 1. The filter size of the dense block is written as 3×3(×8), which means that the dense block is composed of eight convolution layers, and the convolution kernel size of each convolution layer is 3×3.

In summary, we will achieve feature reuse by connecting multiple convolutional layers and obtain more discriminative features with mixed attention.

3.2 Loss function

In general, the softmax cross entropy loss is often used for classification problems. The softmax cross entropy loss can be written as

$$L_s = -\sum_{k=1}^{N}\sum_{i=1}^{D} t_{ki}\log(softmax(y_{ki})) \tag{2}$$

where N is the total number of samples, D is the number of

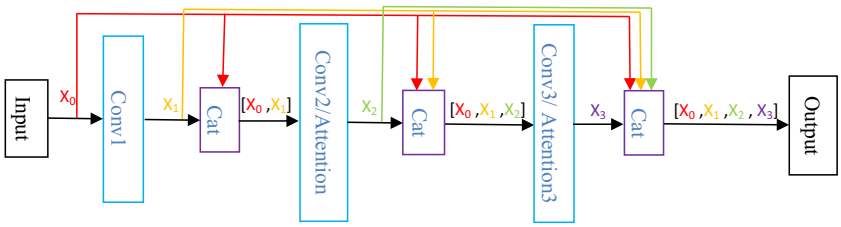
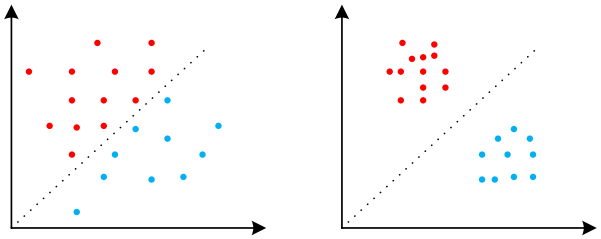


Fig. 2 Dense block



(a) cross entropy loss (b) cross entropy loss + center loss

Fig. 3 Feature distribution learned from different loss functions. **a** cross entropy loss **b** cross entropy loss + center loss

categories, t_{ki} represents the probability that the kth sample belongs to the ith category and y_{ki} represents the probability that the kth sample is predicted as the ith category.

The softmax cross entropy loss function can increase the distance between different categories, but the distribution of different samples in the same category is still scattered, and some hard samples are often close in the feature space [27], as illustrated in Fig. 3a.

The center loss proposed in [27] can reduce the distance between the same categories, as shown in Fig. 3b. The definition of the center loss is as follows:

$$L_C = \frac{1}{2}\sum_{i=1}^{m}\left\|x_1 - c_{yi}\right\|_2 \tag{3}$$

where m represents the total number of samples of a training batch, x_i represents the deep features of the output, y_i represents the category of the ith sample and c_{yi} represents the feature center of the category y_i. Since c_{yi} is computed by averaging the features of the corresponding label predicted (this may not be the true label) in each iteration, it should be updated separately, as described in [27]. Therefore, this equation is optimized to make the sum of squares distance between the feature and the feature center of each sample in the batch as small as possible. That is, the smaller the distance within the class, the better the results that can be produced.

The overall loss is the weighted combination of the two losses:

$$L = L_S + \lambda L_C \tag{4}$$

where λ is a parameter to control Lc.

4 Experiment

4.1 Dataset and experimental details

We conduct experiments on the TU-Berlin dataset, which contains 250 categories. Each category contains 80 sketches, and the total number of samples in the dataset is 20,000.

We implement the model using the TensorFlow framework. In all experiments, we set the learning rate to 0.001. We adopt the Adam optimizer in the training stage. The batch size is set to 128.

The sample number of each class is relatively small, and direct training on the original dataset may lead to overfitting, so the technique provided by [4] is used for data augmentation in section 4.3. Additionally, during the training stage, common augmentation methods such as horizontal flipping and random cropping are also employed.

4.2 Comparative experiments

In order to verify the effectiveness of the proposed network, we compare the proposed network with other thirteen different methods. For the first nine methods [1, 4, 5, 19, 28–31], we randomly select 2/3 of the dataset for training and 1/3 of the dataset for testing, and the experimental results are summarized in Table 2. For another four methods [20–22, 24], in order to do a fair comparison with them at other ratios of the

Table 1 Network structure

Layer	Type	Filter size	Filter num	Stride	Pad	Output size
	Input	–	–	–	–	225×225
L1	Conv1/relu	15×15	64	3	6	75×75
	Maxpool	7×7	–	2	0	35×35
L2	Dense block1	3×3(×8)	32	1	1	35×35
L3	Conv2/relu	3×3	128	1	1	35×35
	Maxpool	7×7	–	2	0	15×15
L4	Dense block2	3×3(×8)	64	1	1	15×15
L5	Conv3/relu	3×3	256	1	1	15×15
	Maxpool	7×7	–	2	0	5×5
L6	Conv(=FC) /relu	5×5	512	1	0	1×1
L7	Conv(=FC) /relu	1×1	512	1	0	1×1
L8	Conv(=FC)	1×1	250	1	0	1×1

Table 2 Results of different methods on the TU-Berlin dataset

Methods	Accuracy (%)
Humans	73.10
HOG-SVM	56.00
FV-SP	68.90
Sketch-a-Net1	74.90
DeepSketch1	75.42
DeepSketch2	77.69
Sketch-a-Net2	77.95
SketchPointNet	74.22
Transfer Learning	72.50
Ours (2:1)	**78.25**

training set and test set, we randomly select {1/2, 3/4, 4/5, 9/10} of the dataset for training. The corresponding remaining images are treated as the test sets, and the experimental results are summarized in Table 3.

Humans [1] represents the accuracy of human recognition on the TU-Berlin dataset. HOG-SVM [1] and FV-SP [28] are methods for manually extracting features. HOG-SVM is the method using BOF features and a support vector machine, and FV-SP uses a mixture of the SIFT model and the Fisher model. DeepSketch1 [29], DeepSketch2 [30], Sketch-a-Net1 [4], Sketch-a-Net2 [5], Transfer Learning [19], Dynamic Landmarks [20], Deformable-CNN [21], Hybrid-CNN [24] and CNG-SCN [22] are eight methods using CNNs. SketchPointNet [31] was a point based DNN method, which directly discretized the sketch into a series of points as the network input.

According to the results, we can see that the accuracy of the method using manual feature extraction does not exceed the accuracy of humans, as the methods of manual feature extraction are mostly designed for information-rich images and are not completely suitable for sketches. The DNN specifically designed for sketches is significantly more accurate than the manual method, and it also exceeds the accuracy of humans.

Table 3 Results of the other four methods on the TU-Berlin dataset

Methods	Accuracy (%)
Dynamic Landmarks (4:1)	82.95
Dynamic Landmarks (1:1)	76.11
CNG-SCN (4:1)	80.10
Ours (4:1)	**83.03**
Ours (1:1)	**75.91**
Deformable-CNN (3:1)	79.10
Ours (3:1)	79.88
Hybrid-CNN (9:1)	85.07
Ours (9:1)	**85.55**

Our method achieves the highest accuracy in this experiment, 78.25%.

When the ratios are 1:1, 3:1, 4:1 and 9:1, the performances of our method are 75.91%, 79.88%, 83.03% and 85.55%, respectively; our method outperforms the corresponding methods.

4.3 Impact of loss function and data augmentation

In this section, the impact of the center loss and data augmentation is evaluated. We first use the technique provided in [4] for data augmentation. In [4], they divided a sketch into three parts, as illustrated in Fig. 4. The first column is the original sketch, the sketches in the first line are the three parts of the original sketch and those in the second line are the augmented sketches obtained by removing the sketches of the first line from the original sketch. Thus, one sketch can be expanded to four.

Figure 5 shows the effect of center loss on our model. Figure 5 shows that when we do not apply data augmentation, the center loss improves the accuracy of our model by 1.06%, and when we use data augmentation, the center loss improves the accuracy of our model by 1.2%. It can be seen that combining the center loss and the cross-entropy loss is better than using the cross-entropy loss alone.

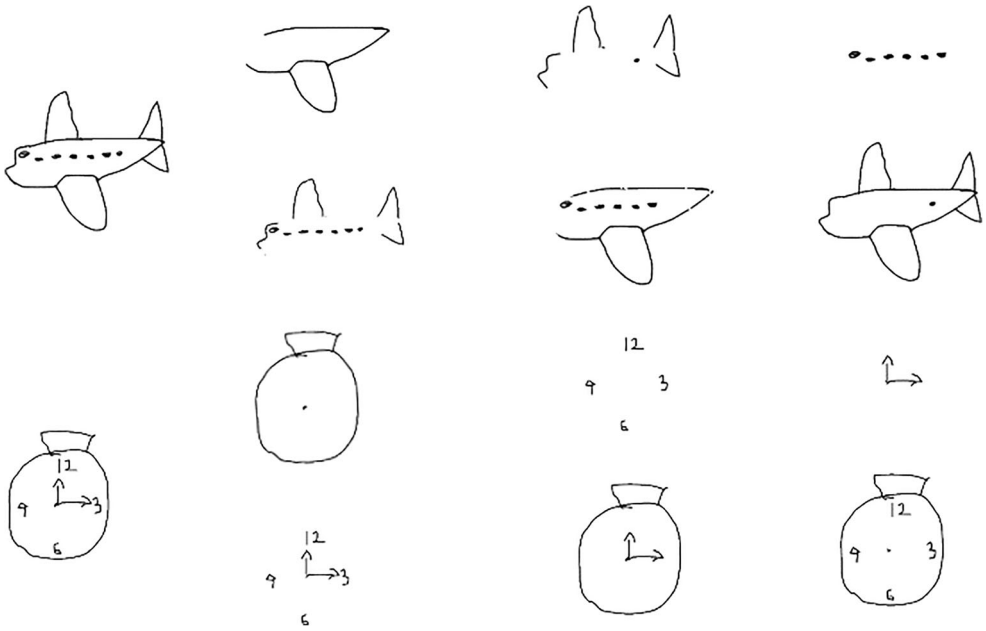
According to Fig. 5, we can find that for the model without the center loss, data augmentation improves the accuracy by 6.13%, and for our model with the center loss, data augmentation improves the accuracy by 6.27%. From the above experimental data, data augmentation has a significant impact on the effectiveness of the models, and narrowing the gap within the class with center loss is useful to improve performance.

4.4 Impact of different backbones

To verify the effectiveness of the dense network, four different backbones are compared. The first network structure is sketch-a-net, which is the network specifically designed to solve sketch problems. It consists of five convolutional layers and three fully connected layers. In the second network structure, we replace the dense blocks of our model with a residual network, which is composed of four convolutional layers. In the third network structure, we replace the dense blocks of our model with the eight-layer stacked network. The fourth network structure is the dense network. The specific experimental results are summarized in Table 4. Note that only the data augmentation method is used, and the attention mechanism and center loss are not used in the fourth network structure.

According to Table 4, we can see that the classification accuracy of the sketch-a-net network is 73.82%, and the residual network accuracy is 76.14%. This shows that increasing the number of network layers and making the gradient flow to the shallow layer can effectively enhance the

Fig. 4 Data augmentation



network's fitting ability. The dense network can achieve an accuracy of 76.82%, which is 0.68% and 2.6% higher than the accuracies of the residual network and the eight-layer stacked network, respectively. This is because the dense network connects multiple layers of the network, which not only makes the shallow network easy to train, but also makes better use of the features of the middle layers to realize feature reuse.

4.5 Impact of attention mechanism

In order to verify the effectiveness of using mixed attention, we compare the effects of three attention mechanisms, including spatial attention [32], channel attention [32] and the mixed attention used in our model. The spatial domain attention mechanism is composed of a 1×1 convolution layer with one channel and using the sigmoid function. The channel attention mechanism first performs global average pooling and global maximum pooling on the feature map; then, the two pooling results are passed through a weight-shared fully connected layer. Finally, the sum of the results is used to get the attention of each channel by using the sigmoid function. In order to avoid the loss of some useful information caused by the multiple multiplications of the convolution layer and the attention mask, we apply the original convolution layer and the multiplied convolution layer. The experimental results are summarized in Fig. 6.

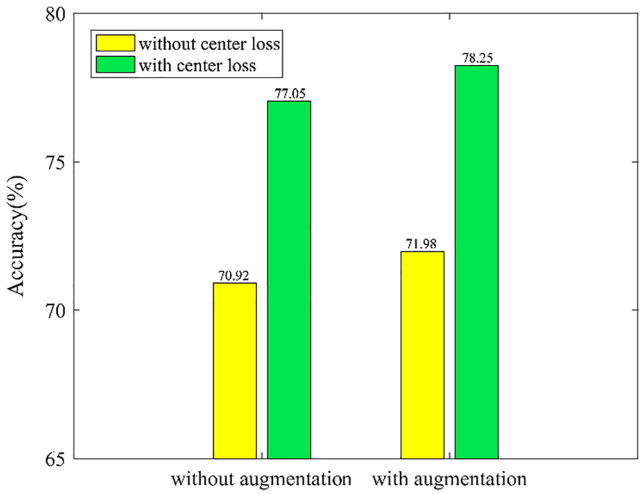


Fig. 5 Effects of loss function and data augmentation

By comparing the effects of three attention mechanisms with center loss, we can see that the accuracy of the spatial attention mechanism is 77.98%, which is 0.27% lower than that of the mixed attention mechanism, and the accuracy of the channel attention is 77.83%, which is 0.42% lower than that of the mixed attention mechanism. By comparing the effects of three attention mechanisms without center loss, we can see that the accuracy of the spatial attention mechanism is 76.74%, which is 0.29% lower than that of the mixed attention mechanism, and the accuracy of the channel attention mechanism is 76.93%, which is 0.12% lower than that of the mixed attention mechanism. The experimental results indicate that the mixed attention mechanism can obtain better results than

Table 4 Experiments about the influence of different backbones

Method	Accuracy (%)
sketch-a-net	73.82
residual network	76.14
eight-layer stacked network	74.22
dense network	76.82

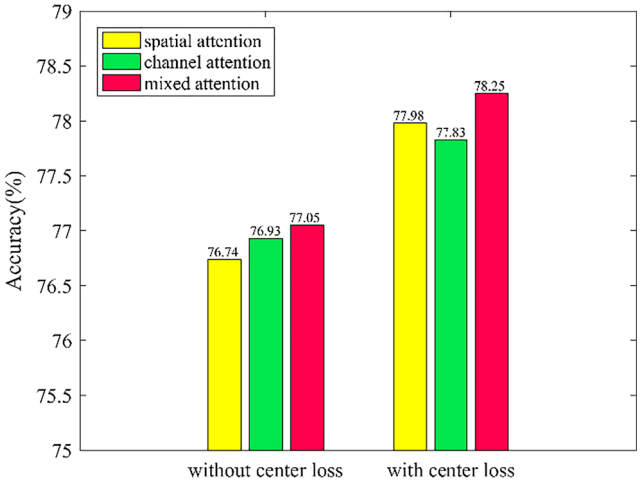


Fig. 6 Effects of different attention mechanisms

the other two attention mechanisms. This is probably because that the mixed attention mechanism achieve the effects of both spatial attention and channel attention, and the local information of each feature map from different channels is helpful to extract representative features.

4.6 Impact of the dense block layers and the pooling layers

To verify the effectiveness of different dense block layers, we compared the results of using a four-layer dense block, an eight-layer dense block and a sixteen-layer dense block. In addition, in order to verify the effectiveness of larger pooling sizes, we compared the effects of different pooling sizes. The experimental results are summarized in Table 5 and Table 6.

By comparing the effects of different numbers of dense block layers on the model, we can see that the accuracy of the four-layer dense block is 76.80%, which is 1.45% lower than that of the eight-layer dense block. The accuracy of the sixteen-layer dense block is 77.52%, which is 0.73% lower than that of the eight-layer dense block. Therefore, we chose the eight-layer dense block as our building block.

By comparing the effects of different pooling sizes on the model, we can see that the accuracy of the 3×3 pooling size is 75.05%, which is the lowest, and the accuracy of 7×7 pooling size is 78.25%, which is the highest. According to [26], we know that overlapping pooling can improve the accuracy. Additionally, [4] also verified the effectiveness of overlapping pooling on the sketch problem, using a 3×3 pooling size with stride 2. Considering the sparse characteristics of the sketch, we use a pooling size of 7×7 with stride 2 to improve the classification accuracy, while 9×9 or larger pooling cannot obtain sufficient fine-grained features.

Table 5 Effects of dense block layers

Method	Accuracy (%)
four-layer dense block	76.80
eight-layer dense block	78.25
sixteen-layer dense block	77.52

Table 6 Effects of different pooling layers

Method	Accuracy (%)
3×3 pooling	75.05
5×5 pooling	77.12
7×7 pooling	78.25
9×9 pooling	76.75

5 Conclusion

We have proposed a novel network for sketch classification by combining a dense network and the mixed attention mechanism. Using the dense block to synthesize features of different convolutional layers not only achieves feature reuse, but also eases problems such as gradient disappearance and network degradation. At the same time, mixed attention is applied in the dense blocks to obtain more representative features. In addition, the center loss and cross entropy loss are also combined to improve the classification accuracy. The effectiveness of our method has been proved through experiments. However, our model cannot distinguish some highly similar sketches, such as a banana and a crescent moon, well. In our future work, we will pay more attention to this problem.

Acknowledgments This study was supported by the Open Research Fund of the National Engineering Research Center for Agro-Ecological Big Data Analysis & Application, Anhui University (No. A201903), the National Natural Science Foundation of China (No. 61772032; 61672032) and the National Key R&D Project (SQ2018YFC080102).

References

1. Eitz M, Hays J, Alexa M (2012) How do humans sketch objects? ACM Trans Graph 31(4):44.1–44.10
2. Eyiokur Fİ, Yaman D, Ekenel HK (2018) Sketch classification with deep learning models. In: 26th Signal processing and communications applications conference, pp 1–4
3. Zhang JH, Chen YL, Li L et al (2018) Context-based sketch classification. In: Proceedings of the Joint Symposium on Computational Aesthetics and Sketch-Based Interfaces and Modeling and Non-Photorealistic Animation and Rendering, pp 1–10
4. Yu Q, Yang YX, Song YZ et al (2015) Sketch-a-net that beats humans. In: British machine vision conference
5. Yu Q, Yang YX, Liu F, Song YZ, Xiang T, Hospedales TM (2017) Sketch-a-net: A deep neural network that beats humans. Int J Comput Vis 122(3):411–425

6. He KM, Zhang XY, Ren SQ et al (2016) Deep residual learning for image recognition. In: 2016 IEEE conference on computer vision and pattern recognition, pp 770–778
7. Huang G, Liu Z, Maaten LVD (2017) Densely connected convolutional networks. In: 2017 IEEE conference on computer vision and pattern recognition, pp 2261–2269
8. Sutherland IE (1964) Sketch pad a man-machine graphical communication system. Proceedings of the SHARE Design Automation Workshop, In, pp 329–346
9. Saavedra JM, Barrios JM, Orand S (2015) Sketch based image retrieval using learned keyshapes. British Machine Vision Conference, pp 164:1–164.11
10. Li K, Pang KY, Song YZ et al (2016) Fine-grained sketch-based image retrieval: the role of part-aware attributes. In: 2016 IEEE winter conference on applications of computer vision, pp 1–9
11. Qi YG, Song YZ, Zhang HG et al (2016) Sketch-based image retrieval via Siamese convolutional neural network [A]. In: 2016 IEEE international conference on image processing, pp 2460–2464
12. Wang F, Kang L, Li Y (2015) Sketch-based 3D shape retrieval using convolutional neural networks. In: In: 2015 IEEE conference on computer vision and pattern recognition, pp 1875–1884
13. Huang Z, Fu HB, Lau RWH (2014) Data-driven segmentation and labeling of freehand sketches. ACM Trans Graph 33(6):175.1–175.10
14. Sun ZB, Wang CH, Zhang LQ et al (2012) Free hand-drawn sketch segmentation. In: European Conference on Computer Vision, pp 626–639
15. Li K, Pang KY, Song JF et al (2018) Universal sketch perceptual grouping. In: European Conference on Computer Vision, pp 593–609
16. Li L, Fu HB, Tai CL (2018) Fast sketch segmentation and labeling with deep learning. IEEE Comput Graph Appl 39(2):38–51
17. Sarvadevabhatla RK, Babu RV (2015) Freehand sketch recognition using deep features. arXiv preprint arXiv:1502.00254
18. Yang YX, Hospedales TM (2015) Deep neural networks for sketch recognition. arXiv preprint arXiv:1501.07873v1
19. Sert M, Boyac E (2019) Sketch recognition using transfer learning. Multimed Tools Appl 78:17095–17112
20. Zhang H, She P, Liu Y, Gan J, Cao X, Foroosh H (2019) Learning structural representations via dynamic object landmarks discovery for sketch recognition and retrieval. IEEE Trans Image Process 28(9):4486–4499
21. Shi Y, Wang K (2020) Sketch recognition based on deformable convolutional network. Computer Science Engineering 6(2): 2456–1843
22. Zhang K, Luo W, Ma L et al (2019) Cousin network guided sketch recognition via latent attribute warehouse. In: Proceedings of the AAAI conference on artificial intelligence, pp 9203–9210
23. He J-Y, Wu X, Jiang Y-G, Zhao B, Peng Q (2017) Sketch recognition with deep visual-sequential fusion model. In: Proceedings of the 25th ACM international conference on multimedia, pp 448–456
24. Zhang X, Huang Y, Zou Q, Pei Y, Zhang R, Wang S (2020) A hybrid convolutional neural network for sketch recognition. Pattern Recognition Letters 130:73–82
25. Kong N, Hou H, Bai Z et al (2019) Lightweight neural network for sketch recognition on mobile phones. In: 5th EAI international summit, pp 428–439
26. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Proces Syst 25:1–9
27. Wen YD, Zhang KP, Li ZF et al (2016) A discriminative feature learning approach for deep face recognition. In: European Conference on Computer Vision, pp 499–515
28. Schneider RG, Tuytelaars T (2014) Sketch classification and classification-driven analysis using fisher vectors. ACM Trans Graph 33(6):174.1–174.9
29. Seddati O, Dupont S, Mahmoudi S (2015) DeepSketch: deep convolutional neural networks for sketch recognition and similarity search. In: 13th international workshop on content-based multimedia indexing, pp 1–6
30. Seddati O, Dupont S, Mahmoudi S (2016) DeepSketch 2: deep convolutional neural networks for partial sketch recognition. In: 14th international workshop on content-based multimedia indexing, pp 1–6
31. Wang XX, Chen XJ, Zha ZJ (2018) Sketchpointnet: a compact network for robust sketch recognition. In: 2018 25th IEEE international conference on image processing, pp 2994–2998
32. Woo S, Park J, Lee JY et al (2018) CBAM: convolutional block attention module. In: European Conference on Computer Vision, pp 3–19

Ming Zhu is currently an associate professor with the School of Electronics and Information Engineering, Anhui University, Hefei, China. His research interests include computer vision and pattern recognition.