CARDIFF UNIVERSITY
PRIFYSGOL CAERDYDD

ORCA – Online Research @ Cardiff

This is an Open Access document downloaded from ORCA, Cardiff University's institutional repository:https://orca.cardiff.ac.uk/id/eprint/145757/

This is the author's version of a work that was submitted to / accepted for publication.

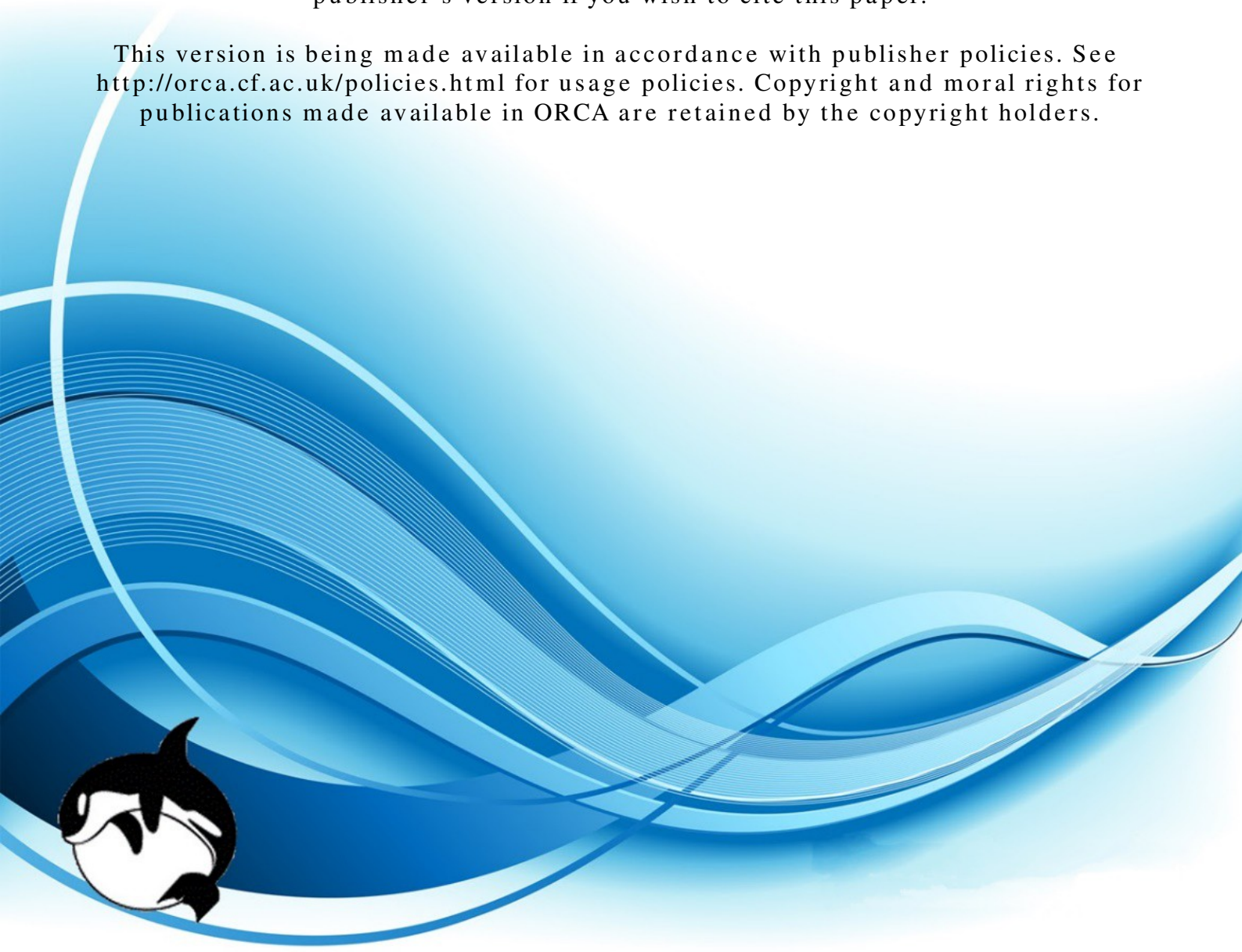
Citation for final published version:

Guo, Nan, Gu, Ke, Qiao, Junfei and Liu, Hantao 2022. Active vision for deep visual learning: a unified pooling framework. IEEE Transactions on Industrial Informatics 18 (10) , pp. 6610-6618. 10.1109/TII.2021.3129813

Publishers page: http://dx.doi.org/10.1109/TII.2021.3129813

Please note:
Changes made as a result of publishing processes such as copy-editing, formatting and page numbers may not be reflected in this version. For the definitive version of this publication, please refer to the published source. You are advised to consult the publisher's version if you wish to cite this paper.

This version is being made available in accordance with publisher policies. See http://orca.cf.ac.uk/policies.html for usage policies. Copyright and moral rights for publications made available in ORCA are retained by the copyright holders.



Active Vision for Deep Visual Learning: A Unified Pooling Framework

Nan Guo, Ke Gu, *Member, IEEE*, Junfei Qiao, *Senior Member, IEEE*, and Hantao Liu, *Senior Member, IEEE*

***Abstract*—Convolutional Neural Networks (CNNs) can be generally regarded as learning-based visual systems for computer vision tasks. By imitating the operating mechanism of the human visual system (HVS), CNNs can even achieve better results than human beings in some visual tasks. However, they are primary when compared to the HVS for the reason that the HVS has the ability of active vision to promptly analyze and adapt to specific tasks. In this study, a new unified pooling framework was proposed and a series of pooling methods were designed based on the framework to implement active vision to CNNs. In addition, an active selection pooling (ASP) was put forward to reorganize existing and newly proposed pooling methods. The CNN models with ASP tend to have a behavior of focus selection according to tasks during training process, which acts extrememly similar to the HVS.**

***Index Terms*—deep visual learning, pooling framework, active vision, deep convolutional neural networks, human visual system**

I. Introduction

GAINING and processing information like human vision system (HVS) is the final goal of computer vision [1]. Great achievements have been made in this area during the past few decades [2]-[11]. Among them, convolutional neural networks (CNNs) play an important role in dealing with plenty of visual tasks, such as image classification, recognition and segmentation. CNNs can be regarded as a learning-based visual system that basically consists of convolutional layers, pooling layers, normalization layers, nonlinear transformation layers and application layers. Pooling layers contribute a lot to CNNs in reducing computation load and information redundant. However, pooling layers also arouse controversy, because some valuable information may be lost when downsampling. Geoffery Hinton held the belief that pooling layers work so well in CNNs is a disaster that directly gives birth to the famous Capsule Networks [12]. In this paper, we make an intensive study to the pooling layers and offer an effective solution to alleviate the above-mentioned problems.

Pooling operation is an important part in CNNs, since it aggregates a local region data in one channel feature and transforms them into only one value. The question is that we have no idea whether the retained value after pooling operation is representative enough or not for the local region data. Therefore, we hold a view that the commonly used max-pooling or average-pooling are coarse grained and inflexible. To meet this challenge, we took steps in two aspects. First, a unified pooling framework was proposed to generate more pooling methods so as to offer more choices for pooling operation. Second, an active selection strategy was designed to decide which pooling method to be chosen from the new designed and existing pooling methods during training process of CNNs. We aim to allow CNNs to extract key information automatically like the human visual system (HVS) through a combination of pooling methods and proposed active selection strategy. To address the problems mentioned above, we made a profound investigation of existing pooling methods.

Hand-crafted pooling. Max pooling and average pooling are commonly used in hand-crafted pooling operation, which has some advantages, such as clear meaning, simplifying calculation and alleviating over-fitting. However, it is hard to make the training error converge to the global minimum value. What's more, the extracted features by max or average pooling are quite limited, and such phenomenon may weaken the representation ability of deep CNNs [14].

Learning-based pooling. Learning-based pooling operations are usually obtained by training end-to-end deep models. In [14], the authors tried to improve the performance of pooling operations by combining average pooling and max pooling operation, and named it "mixed average-max pooling", whose two hyper parameters were introduced to make a tradeoff between the max pooling and the average pooling. In [13], a learning-based end to end pooling operation called LEAP was put forward to overcome drawbacks of hand-craft pooling. The LEAP learned a shared linear combination of neurons for each feature map that can be generated by pooling operation as well as simplified convolutional layer. Although the learning-based pooling operation attempts to minimize the training error during the training stage, it's still difficult to keep a balance between searching range of the feature space and reducing computation cost, so the learning-based pooling remains to be further improved.

Probabilistic pooling. Probabilistic pooling is also an important field for pooling research. It has been proved to be effective that the stochastic pooling method proposed by Zeiler and Fergus adopted a size-based sampling method in each pooling region of feature maps [15], [16]. In [22], a rank-based stochastic pooling operation was embedded into a nine-layer convolutional neural network to replace the average or max pooling operation and achieve state-of-the-art accuracy in abnormal breast identification. Plenty of studies have shown

This work was supported in part by the National Science Foundation of China under Grant 62076013, Grant 62021003, and Grant 61890935. (Corresponding author: Junfei Qiao.)

Nan Guo, Ke Gu, and Junfei Qiao are with Faculty of Information Technology, Beijing University of Technology, Engineering Research Center of Intelligent Perception and Autonomous Control, Ministry of Education, Beijing Laboratory of Smart Environmental Protection, Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing Artificial Intelligence Institute, Beijing 100124, China (e-mails: guonan03@126.com; guke.doctor@gmail.com; junfeiq@bjut.edu.cn).

Hantao Liu is with the School of Computer Science and Informatics, Cardiff University, Cardiff, U.K.

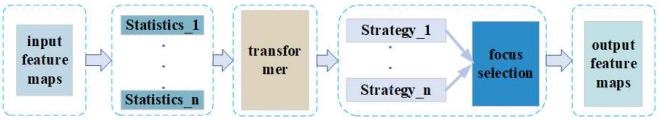


Fig. 1: The structure of proposed unified pooling framework.

that stochastic pooling performed better than average and max pooling.

Second-order and high-order pooling. In some special visual tasks, for example, the fine-grained classification, traditional hand-crafted pooling fails to tackle such difficult mission for the reason of information loss, and such problem has attracted a lot of researches' attention [17]-[19]. To address these problems, some researches further studied second-order or high-order pooling. In [17], the authors devised the bilinear pooling to solve the problem of fine-grained classification for the first time. Bilinear pooling reacts in feature fusion to gain second-order or high-order representation of images (or features). If two same features were fused, bilinear pooling turned to homogeneous bilinear pooling (or second-order pooling) [20], [21].

To conclude, we have systematically studied the pooling operations in CNNs and extracted some key points from the above four directions of pooling: 1) hand-crafted pooling has simple and clear mean, but it may be lacking in traversing the input data space and obtaining key information; 2) learning-based pooling has more chances to get better results through an end-to-end training process, however, it also brings more uncertainty to pooling layer and CNNs; 3) sampling methods can be applied to pooling layers to design frequency bias pooling operations; 4) second-order and high-order information is also needed in some visual tasks.

Based on these views, we conclude our work in this paper as follows.

1) We introduced a unified pooling framework which can cover the majority of existing pooling methods. The framework is general and flexible that can be utilized to design new pooling operations in order to remedy the disadvantages of common used hand-crafted pooling methods.

2) We devised several new pooling methods base on the proposed unified pooling framework, including first-order pooling and second-order pooling (see Table I), which provides more alternative choices of existing hand-crafted pooling in CNNs.

3) A novel pooling approach called active selection pooling (ASP) was devoted to implement active vision in CNNs. An active selection strategy was designed in ASP to automatically decide which pooling operation to be chosen during training process. CNNs with ASP are able to distill information in the way similar to human visual system.

4) Experimental results conducted on normal visual tasks and few-shot visual tasks have proved effectiveness of proposed unified pooling framework, and the proposed ASP pooling outperforms existing pooling operations by introducing active vision idea into CNNs.

II. Proposed Unified Pooling Framework

Recent years have witnessed the extensively use of CNNs in computer vision. The pooling operation plays an important role in a CNN, but it still has some flaws. In this paper, we view a CNN as a learning-based visual system which has the ability to extract and process information from input images. To relive the information loss problem in pooling operation, we devised a unified pooling framework to generate more pooling methods to strengthen pooling layers in CNNs. Besides, inspired by the active vision trait of human visual system, we presented the active selection pooling (ASP) approach to improve the traditional hand-crafted pooling and made the CNNs more close to human visual system (HVS). We arrange the whole Section III by first introducing the general structure of the proposed ASP. Then, we present details of each part in the framework. After that, we illustrate how our ASP operates in a CNN through a few examples. Finally, we embed the designed pooling into some existing networks for clear presentation.

A. General Structure of Proposed Unified Pooling Framework

Traditional pooling layers such as average pooling and max pooling benefit the convolutional neural networks mainly for their downsampling function, which greatly simplifies the calculation in CNNs. But pooling layers also bring some problems, information loss, for example, is long being criticized by researchers. In this paper, we argue that the biggest problem of existing pooling layers is that they have no choice to decide which part of information to retain. Only information that conforms to certain characteristic can be reserved after no matter max or average pooling. But we have no idea that whether the remained information is suitable for CNNs or not. It's counterintuitive to the HVS, because human eyes can timely change focus to seek important information. Based on such inspiration, we designed the ASP to make CNNs act more similar to the HVS.

Our designed pooling framework consists of four parts: 1) primary statistic part; 2) active vision strategies part; 3) transformer part; 4) and focus selection part. Details of each part will be introduced in rest of this section.

B. Statistics of Input Data in Pixel-level

Input images statistics. The statistics part is shown in Fig. 1. If the input are original images, statistics operation gives a description of the pixel distribution in the input image. We adopt the histogram method to gain the distribution of pixels. If the input image is an RGB image, the dims parameter is 3. Otherwise, the dims is 1. For simplicity, we do not re-divide the pixel level, so the bins parameter in histogram is set to 1. The range of the pixel is [0,255]. Then for any pixel x in the input image, its probability can be obtained through function (1).

$$p(x) = N(x)/(H \times W) \tag{1}$$

where *N(x)* is the times that *x* emerges in image; *H* and *W* are the height and width of the input.

Input feature maps statistics. In deep CNNs, feature maps are usually gained after normalization or sigmoid activation

TABLE I: Some implements of proposed pooling framework. Header of the table contains each part of proposed unified pooling paper. Pooling methods with the background color of light blue are local pooling operations. The ones with light brown background color are global pooling operations. Besides, pooling approaches with red foreground color are new designed according to proposed unified pooling framework.

	$x(H \times W)$	$p(x)$	$f\_math(x)$	$f\_statistic(x)$	$t(f(x))$	y
average pooling	x	$1/(H \times W)$	$sum(x)$	1	1	$1/(H \times W) \times sum(x)$
max pooling	x	1	$max(x)$	1	1	$max(x)$
median pooling	x	1	$get\_middle(x)$	1	1	$get\_middle(x)$
prob average pooling	x	$p(x)$	$sum(x)$	1	1	$sum(p(x) \times x)$
global average pooling	x	$1/(H \times W)$	$sum(x)$	1	1	$1/(H \times W) \times sum(x)$
global max pooling	x	1	$max(x)$	1	1	$max(x)$
variance pooling	x	1	1	$get\_var(x)$	$get\_normal()$	$I - get\_normal(get\_var(x))$
global median pooling	x	1	1	$get\_middle(x)$	1	$get\_middle(x)$
quartile pooling	x	1	1	$get\_quartile(x)$	1	$get\_quartile(x)$
entropy pooling	x	$p(x)$	1	$get\_entropy(x)$	$get\_normal()$	$get\_normal(get\_entropy(p(x)))$
prob average pooling	x	$p(x)$	$sum(x)$	1	1	$sum(p(x) \times x)$

which are nonlinear transformations of the input. Under this circumstance, elements in a feature map can not be statistic simply through method mentioned above. We should set bins for the histogram to divide the elements in feature map into different levels. Suppose that the feature map has been divided into l levels. $i \in [1, l]$ is an integer. $l_{low}(i)$ and $l_{up}(i)$ are the boarder lines of the $i-th$ level. $N(x_i)$ is total number of the elements that belong to $i-th$ level. So, $p(x_i)$ can be described as equation (2).

$$p(x_i) = \begin{cases} N(x_i)/(H \times W), & l_{up}(i) > x_i \geq l_{low}(i) \\ 0, & else. \end{cases} \tag{2}$$

C. Design of Active Vision Strategies

Original intention. The original idea comes from computer vision theory in 1990s. A camera pan can control the camera orientation in different tasks or orders, which is called active vision. It's a kind of visual system that imitates human visual system. However, it has no ability to learn and think. Deep CNNs can be perceived as visual system based on learning when they are applied to visual tasks. We design an pooling framework to be a bridge between active vision idea and deep CNNs.

Principles of designing. The design of strategies can be very flexible, but there are some principles that we should take into consideration. First, we should not change the down sample property of pooling layer for saving computing resource. Second, the logic of the active vision strategy should not be too complicated in order to fit deep neural models which are usually trained with data flow. Third, each strategy should reflect one special feature of the input data. The final goal of proposed active vision strategies is to explore the input data space and statistical space more sufficiently during training process of deep convolutional neural network models.

Active vision strategies. We design two categories of strategies: 1) pixel value level strategies; 2) pixel statistic level strategies. Pix value level strategies mean that the strategies are gained through simple math computation of the input feature map. Usually the value of the active vision feature gained after designed pooling layer is not beyond input feature value. Details of the pixel value level strategies can be depicted as function (3).

$$y = f_{math}(x) \times p(x) \tag{3}$$

where $p(x)$ is the calculation result of function (2). We set a restriction to function (3) that $y \in [min(x), max(x)]$. Where $min(x)$ is the minimum value of x, and $max(x)$ is the maximum value of x. This restriction is obtained through experiments. We find that if y is a value out of input feature x, CNNs tend to diverge in the training process. This category of strategies can be used to calculate a feature value which reflects a characteristic of the input. These strategies are alternative options for focus-selections part.

Different from pixel value level strategies, pixel statistic level (statistic for short) strategies provide more complex abstractions of input feature maps. The calculation performed by statistic strategies is shown as follows:

$$y = f_{statistic}(x) \times p(x) \tag{4}$$

$$y_{pool} = t(y) \tag{5}$$

where $f_{statistic}(x)$ is the statistics of input x. There are many optimal statistical approaches which can be used in our designed active vision pool. Note that the statistics results should be transformed before embedding into our pooling layer, as is shown in function(5). The detail of the transformation rules will be analysed in subsection F.

We present some implements in Table I to illustrate our ASP. Except for common used average pooling and max pooling, we also proposed some first-order active pooling method instructed by our proposed unified pooling framework. Median pooling, for an example, which is designed to gain the middle value of pooling region in the input features. Prob average pooling as can be see in Table I, is obtained by calculating the probabilistic average value as follows:

$$y = \sum p(x) \times x \tag{6}$$

where x is the collection of pixel value of the pooling region. The value range of y is the same as function (3). $p(x)$ is gained through histogram method as has been mentioned above. *Different from common hand-crafted pooling, prob average pooling takes the frequency spectrum of pooling area into consideration to further explore the feature data space*. We will show its performance in Section III.

Besides, we also provide some examples of statistic level active pooling strategies. ***We want to address that majority of this category of pooling methods perform well in attention mechanisms rather than common pooling scenarios***. **Potential cause of this phenomenon is that common hand-crafted pooling (max or average pooling) can adapt to back propagation algorithm of the CNNs in training process, while statistic level active pooling methods may be invalid to train CNN models**. This limitation disappeared when they are applied to attention mechanisms, for example, channel attention. We provided four global pooling which were constructed based on statistic of the input data as were listed in Table I. Taking variance pooling for instance, it is built on the variance of input data. The variance pooling results can not be directly inserted into a CNN to replace pooling layers for the reason that it will significantly decrease the accuracy of the inserted model. A transformation is utilized to the variance as below:

$$y = I - get\_normal(get\_var(x)) \tag{7}$$

where I is a ones-tensor which has the same size as the variance tensor. $Get\_var(x)$ is a function to get the variance of x, and get_normal is a normalization operation of the variance. As a matter of fact, the variance pooling operation becomes a second-order pooling which is effective for the CNNs after transformation. Details of the transformer part will be interpreted in next subsection and the rest of global active pooling operations can be seen in Section III.

$$y = get\_normal(get\_entropy(x)) \tag{8}$$

Another instance is entropy pooling. We call it entropy pooling because it is obtained by calculating the information entropy of input x. Expression of proposed entropy pooling is displayed as function (8). $get\_entropy(x)$ is the function to get entropy of x and get_normal is same as mentioned in function (7). Similar to variance pooling, entropy pooling is also a second-order pooling method. Specially, the calculation of information entropy depends on the input data. If the input data is images, that is to say $x_i \in [0, 255]$, where $i \in [0, H \times W - 1]$. H and W represent the height and width of input images. However, if the input data is normalized feature maps, we can go to histogram for help as has been mentioned above. Different levels will be divided for the input x and the number of pixel value will be calculated for each level which is necessary for calculating information entropy.

D. Transformer Part

As is shown in Fig. 1 and Table I, transformer part is a transparent function of statistic results. The function can be flexible to make the most use of the statistic features. Taking median pooling for example, which is a first-order pooling method that gained if we set the transformer function to 1 as is shown in Table I. Variance pooling presents a normalization-based transformer as well as entropy pooling. Note that we are able to escalate first-order pooling into second-order or high order pooling. In this paper, we proposed transformer part to provide a solution to explore the second-order or high-order representation of feature maps.

E. Focus Selection Part

We will illustrate the detail of focus selection part through Fig. 2. The pooling operations have been divided into two groups: first-order pooling strategies and second-order/high-order pooling strategies. To achieve focus selection, two stages have been adopted in this part. First, a primary selection operation is applied to decide which group of pooling to be chosen. Note that this action can be realized through learning and back-propagation [13], but our suggestion is to choose simply by expert experience for reducing computation cost. So far, we have partly achieved the goal of focus selection to some degree. Then, it comes to the further selection stage which is a decision making process to gain the final pooling method.

The design of further focus selection strategy is an open question, and here we offer a solution for first-order pooling by introducing the metric idea to the selection strategy. The solution consists of three procedures: 1) sample from pooling region of input data according to [15] to get x'; 2) compute probabilistic average of the sampled data to gain x'_prob_avg according to function (6); 3) compute distances between x'_prob_avg and results of other pooling methods, such as x_avg, x_max, x_middle, $x\_quartile$, and choose the nearest one as the final pooling method. The superiority of our focus selection solution is that we introduce randomness to the pooling layers which may be beneficial to reduce over-fitting problem. As for second-order and high-order pooling, we suggest to apply them depending on visual tasks. Besides, it is more reasonable to utilize them in a fine-tune process for reducing computation cost [23].

F. Operation of The Proposed Active Selection Pooling

Built on statistics of input feature maps and inspired by active vision idea, we establish the active selection pooling (ASP), as is shown in Fig. 1. The proposed method is computed by the following:

$$y = t(f_{math}(x)) \times f_{math}(x) \times p(x) \times x \tag{9}$$

$$y_{global} = t(f_{statistic}(x)) \times f_{statistic}(x) \times p(x) \times x \tag{10}$$

Following function (6), a pooling layer which is similar to average pool and max pool can be obtained. While the pooling layer is more alike to global average pooling if we follow calculation of function (7). Where $p(x)$ is probability of occurrence for x in the input, and it can be calculated by function (1) or function (2). $f\_math(x)$ represents for

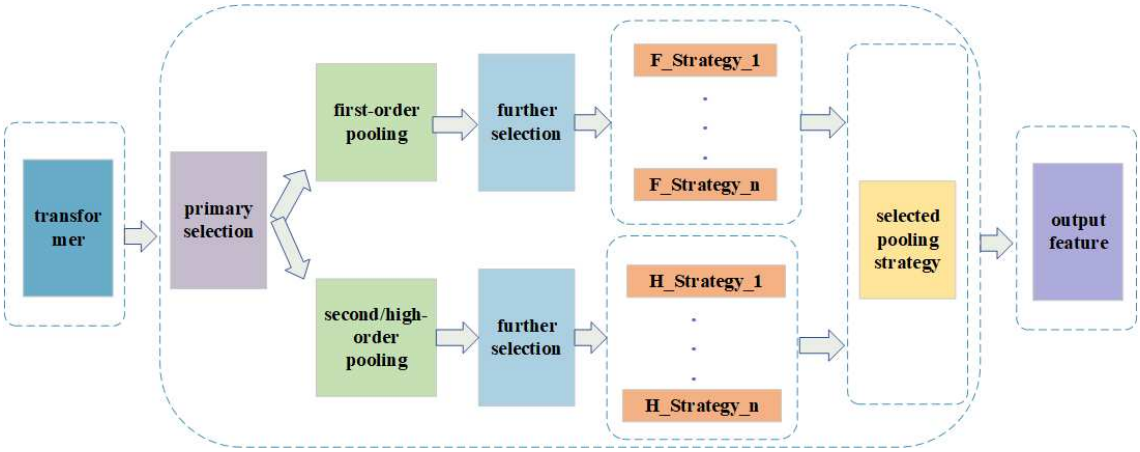


Fig. 2: The structure of proposed active selection pooling(ASP).

statistical feature gained through function (3). $f\_statistic(x)$ is computed as function (4).

What we want to address is that our work can be regarded as a general framework to design pooling layers. We introduce the active vision idea to the layer to initiative explore input data space as well as the statistical space. To illustrate our work, we provide some examples in Table I.

In the table, the parameters of the header have been defined in this section. $sum()$, and $max()$ are functions defined in any of deep learning framework. $get\_middle()$ is a function designed to calculate the middle value of input x. $get\_var()$ is the function to get variance of the feature map. $get\_normal()$ is a self defined function for normalization. $get\_entropy()$ is another method designed to calculate information entropy. Our new proposed active selection strategies have been colored in red. Output of the designed methods can be obtained through multiplying each part showed in the table.

We can clearly infer from Table I that average pooling and max pooling are special cases of our proposed active selection pooling. That is to say, our pooling layer have a baseline of existing pooling method. What's more, we provide more choices for information processing during training of deep neural models. Taking prob average pooling for example, the output feature is calculated by the following:

$$y = sum(p(x) \times x) \tag{11}$$

In this layer, we gain a category of statistic feature which is named "probability average feature". The value of $y \in [min(x), max(x)]$, which makes the feature capable likely to traversal the entire input data space. Besides, this kind of pooling pays more attention to pixels frequency spectrum of feature maps. It is a kind of focus that different from traditional average or max pooling layer.

Another pooling approach that we want to analysis is variance pooling. In this method, we adopt variance as our pooling strategy. But we found that classification error rate would increase significantly. So we transform the variance as is shown in Table I. Then a promising result has been obtained through the new designed pooling. More details can be found in Section III.

TABLE II: Structure of backbone networks for visual classification.

	ResNet34	SEResNet34
basic_conv_block× 1	conv2d,bn,relu	conv2d, bn,relu
basic_residual _block× 4	conv2d,bn,relu, conv2d,bn	conv2d,bn,relu, conv2d,bn,relu
shortcut× 4	conv2d,bn,relu	conv2d,bn,relu
	--	squeeze(avg_pool),
	--	excitation(linear,relu,linear,sigmoid)
pooling layer× 1	avg_pool	avg_pool
classifier × 1	linear classifier	linear classifier

III. Experiments

In this section, we will validate the performance of our proposed ASP to demonstrate its superiority compared with common state-of-the-art pooling methods. This section is composed of experimental settings, visual classification, few-shot visual classification and analysis of proposed ASP.

A. Experimental Settings

Backbone networks and settings. Residual convolutional networks have been widely used as baselines for visual classification tasks [24]-[28], [33] . In this paper we choose ResNet34 and SEResNet34 as our backbones to validate the effectiveness of proposed pooling framework. Original architecture of the baselines are listed in Table II according to [4] and [5]. It is clear that there is a global average pooling layer in both baselines. Besides, each shortcut connection has an global average pooling layer to calculate channel attention. We replace all the pooling layers with proposed ASP to validate our work. We set training epoches to 200. The initial learning rate is set to 0.1 with a decay of 0.2 every 60 steps.

Besides, we also conducted experiments on mini-ImageNet which is a few-shot classification task. The backbone architecture for this task is the same as MAML which is listed in Table IV. The network consists of 4 $basic\_conv2d\_block$ and a linear classifier. Each block is composed of a convolutional layer, a batch normalization layer, a rectified linear unit layer and a max pooling layer. We embedded our designed ASP

TABLE III: Structure of backbone networks for visual classification.

	Conv4 + maxpool	Conv4 + ASP
basic_conv2d_block × 4	conv2d	conv2d
	bn	bn
	relu	relu
	maxpool	ASP
classifier	linear classifier	linear classifier

into the backbone to replace the original pooling layer. The parameter settings followed paper [27].

B. New pooling methods designed according to active selection pooling framework

Median pooling. The core of median pooling is to obtain the middle value of pooling region which is inspired by the idea of median filtering. The calculational logic of median pooling is presented in Algorithm 1. Note that the Algorithm 1 merely presents the logic of calculating median pooling, pipeline features should be considered if we want to apply it to deep models.

Quartile pooling. Calculation of quartile pooling has similar logic to median pooling. Simply changing the value of K in Algorithm 1 to $int((3 \times H \times W)/4)$ will gain the quartile pooling result.

Prob average pooling. This pooling operation is obtained according to function (6) in Section II. In this process, it is the key point to obtain $p(x)$ which has been introduced in subsection B of Section II. The advantage of prob average pooling is that it takes frequency spectrum of an input feature map into consideration which may be beneficial to promote the final results.

Variance pooling. Variance pooling is a second-order pooling method which has been mentioned above. Our original intention to design variance pooling is that variance can reflect the dispersion of data. We want to figure it out that whether it can be applied to pooling operation. During our research, we found that performance of CNNs will drop if we directly replace the average or max pooling with variance value. Considering that variance has opposite meaning compared to average in some degree, we design our variance pooling as is defined in function (7). The proposed variance pooling has been proved to be effective in practice.

Entropy pooling. Calculation of entropy pooling has been elaborated in Section II. In this paper, for brevity, we divide the input feature map into two levels by taking the average value as the boundary.

Active selection pooling method. Different from proposed unified pooling framework mentioned above, here ASP is a pooling method which integrates proposed pooling methods and active selection strategy as is elaborated in Fig. 2. In this paper, we suggest to give a primary selection through human experience. On one side, it is easier for human beings to discriminate requirements of visual tasks. A simple configuration for the selection can tackle this situation in an economical way. On the other side, it is difficult to design a general selection strategy for both first-order pooling methods and second-order (high-order) pooling operations as they have different representation levels. Here, we raised an active focus selection strategy for first-order pooling method and designed an active selection pooling (ASP) method as is listed in the last line of Table IV.

The details of active selection strategy is illustrated in Fig. 3. The procedure of selection strategy for ASP is as followed: 1) sample from pooling region of input data in a put back way independently; 2) gain x' through computing the prob average value according to function (6); 3) calculate distance between x' and pooling results of the methods listed in Figure 3; 4) obtain the minimum value of the distances and deciding the selection.

Algorithm 1 Calculational logic of the proposed median pooling. Here $reshape()$ is a function which is used for emerging dimensions of a tensor. $range()$ performs as a sort function for a tensor. $get\_KthValue()$ is another function to get $K-th$ value in a tensor.

Require: input feature maps tensor $x(B, C, H, W)$, where B is batch size, C is input channels, H and W are height and width of the feature map.
Ensure: $x(B, C, 1) \longleftarrow reshape(x)$, each feature map in x has $H \times W$ pixel value(or normalized value)
for i in [1,B] **do**
  for j in [1,C] **do**
    $x(i, j, 1) \longleftarrow range(x)$
    $x(i, j, 1) \longleftarrow get\_KthValue(x(i, j, 1), K)$, where $K = int((H \times W)/2)$
  end for
end for
  output $x(B, C, 1)$

C. Visual Classification

Comparison pooling methods. As is listed in Table IV, we compare our proposed ASP methods with stochastic pooling which is representative of probabilistic pooling methods. The sampling frequency is set to 10 and 100 according to paper [15]. Besides, we also conducted experiments with LEAP which is a group of learning-based pooling operations [13]. We also proposed some first-order and second-order pooling methods according to our proposed framework as is presented in Table IV.

According to the performance of pooling methods listed in Table IV, we can conclude several expressions as followed.

1) All the pooling methods listed are effective for ResNet34 and SEResNet34, except for variance pooling and entropy pooling in ResNet34. Possible reason for this phenomenon is that it is different for first-order pooling and second-order pooling in feature statistical spaces.

2) Common used average pooling and max pooling have almost the same performance in ResNet34, but max pooling has poor performance in SEResNet34 which has pooling

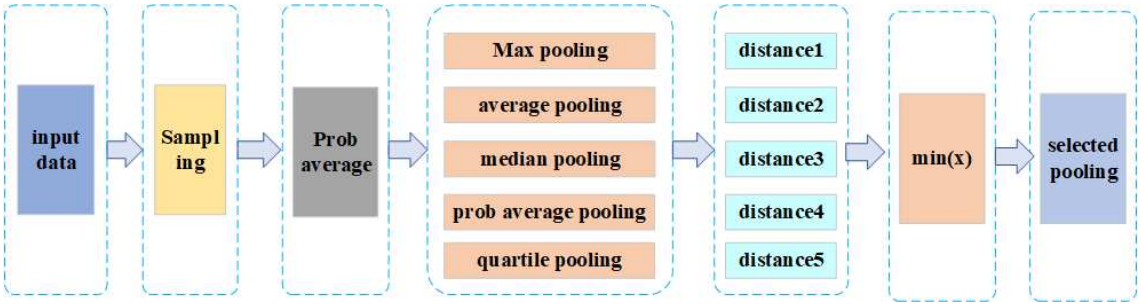


Fig. 3: The details of the proposed active selection strategy.

TABLE IV: Experimental results on CIFAR-10 and CIFAR-100. Experimental results of listed pooling methods are obtained by an average of 5 runs.

Pooling Methods \ Experiments		ResNet34		SEResNet34	
		CIFAR-10	CIFAR100	CIFAR-10	CIFAR-100
existing pooling	average pooling [04]	6.63	23.24	6.12	22.07
	max pooling	6.65	23.43	6.72	23.20
	stochastic pooling(10) [10]	6.55	23.10	5.66	22.00
	stochastic pooling(100) [10]	6.60	23.17	5.85	22.00
	mixed average-max pooling [09]	6.57	23.12	6.03	22.05
	LEAP [08]	6.59	23.10	5.90	22.02
new designed pooling	median pooling	6.42	23.08	5.16	21.53
	quartile pooling	6.77	24.05	6.01	22.13
	prob-average pooling	6.74	23.60	5.83	22.06
	variance pooling	--	--	6.32	23.14
	entropy pooling	--	--	5.55	23.77
ASP	active selection pooling	**6.29**	**22.80**	**4.97**	**21.46**

operation in attention mechanism as is colored in red in Table IV.

3) Stochastic pooling seems to have better performance compared to average and max pooling, and the performance drops as the sampling frequency increases. Stochastic pooling has a baseline of average pooling for the reason that enough sampling times make it equal to average pooling.

4) LEAP and mixed average-max pooling performs better than common used hand-crafted pooling.

5) Among all the pooling methods designed based on proposed ASP, median pooling outperforms the others. Other pooling methods also offer selections for pooling layers, especially when they are applied in channel attention mechanisms like SEResNet.

6) The listed ASP which is an ensemble of proposed pooling methods with active selection strategy outperforms the others, as has been bolded in last line of Table IV.

In fact, our proposed unified pooling framework is capable to incorporate all the pooling methods listed in Table IV. The proposed pooling framework is generalize and flexible for designing pooling layers.

D. Few-shot Visual Classification

To further evaluate the proposed pooling framework, we conducted experiments on MiniImagenet which is a few-shot visual classification task. Experimental results are presented in Table V. Our experiments followed the setting of MAML [27], which is one of the most representative methods in few-shot learning. As has been signed in the table, the max pooling [27] which is in color blue is a baseline for comparison. The line in red is our proposed quartile pooling which gained least competitive experimental results compared to other pooling methods. The proposed ASP achieved the best accuracy in 5-way 1-shot experiments.

We can generalize from Table V that common hand-crafted pooling methods(max or average pooling), learning-based pooling [13], and probabilistic pooling [15] performs well according to our test results. Among them, stochastic pooling with a sampling frequency of 10 tend to have better performance. As has been bolded in the Table V, stochastic pooling achieve best result among the 5-way 5-shot experiments. Possible reason is that stochastic pooling brings more randomness into the training process to explore the feature space [15]. Specially, max pooling outperforms the average pooling in this meta model [27]. In [14], the authors hold an view that max pooling can restrain over-fitting problem for its

TABLE V: Experimental results on MiniImagenet.

Pooling Methods \ Experiments		MiniImagenet	
		5-way 1-shot	5-way 5-shot
existing pooling	average pooling	48.55 ± 1.76	62.87 ± 0.95
	max pooling [22]	48.70 ± 1.84	63.11 ± 0.92
	stochastic pooling(10) [10]	48.93 ± 1.67	**63.35 ± 1.21**
	stochastic pooling(100) [10]	48.66 ± 1.79	63.07 ± 0.95
	mixed average-max pooling [09]	48.77 ± 1.66	63.05 ± 0.90
	LEAP [08]	48.72 ± 1.77	63.15 ± 0.93
new-designed pooling	median pooling	48.66 ± 1.70	63.10 ± 1.04
	quartile pooling	46.77 ± 1.55	63.00 ± 0.96
	prob-average pooling	48.80 ± 1.83	63.15 ± 0.95
	variance pooling	--	--
	entropy pooling	--	--
ASP	active selection pooling	**49.20 ± 1.94**	63.22 ± 1.07

powerful ability in activating neurons in neural networks which may explain this phenomenon. What's more, the pooling methods designed according to ASP operates well in MAML model except for variance pooling and entropy pooling. In a word, our framework offers more choices for pooling layers which may improve the deep models with a replacement operation.

IV. Discussion

Application of second-order and high-order pooling. In this paper, the two proposed second-order pooling methods turned out to be invalid in convolutional architecture of MAML model. The meta model would not convergence if we replace the max pooling layers of the backbone with proposed second-order pooling methods. Then the question is how to apply these pooling methods ? Here we have two suggestions. One suggestion is inspired by bilinear pooling [17], [18]. The second-order pooling can be used for feature fusion and then classification. The other advice is to embed them into attention mechanisms [29]-[32] to calculate the channel attention as is presented in Table IV. The later has been proved in practice.

How to take advantage of ASP. In this paper, we provide a sampling-based strategy to decide which pooling method to choose during training process. However, there are other ways to use these pooling methods rationally. One way is to combine proposed or existing pooling methods to form learning-based pooling which can refer to paper [13]. The advantage of this kind of pooling is that it becomes an end to end learning system which is similar to other parts of neural networks. We all know that end to end learning is one of the key reasons for the success of neural networks. The issue is we do not know what it will learn in the training process and it also brings more uncertainty to original networks. Another way to employ these pooling methods is to concatenate the pooling methods. For an example, a 1×1 convolutional layer can be applied to the networks after these pooling operations to conduct a new one in a fused way.

Design of focus selection strategy. It is an open problem to implement focus selection in deep models. The proposed active selection pooling is an effort to meet this challenge. In our opinion, embedding human knowledge to deep neural networks is a popular trend, because it is beneficial to understand why deep neural models work well.

V. Conclusion

In this paper, we treat deep convolutional neural networks as learning-based visual systems, and present a unified pooling framework to further explore the information extraction ability of CNNs. New pooling methods were designed based on proposed framework, such as median pooling, quartile pooling, prob average pooling, variance pooling and entropy pooling. In addition, a novel pooling approach called active selection pooling (ASP) was proposed to unify existing and new designed pooling by a sample-based active selection strategy. Experimental results on several popular datasets are able to derive three crucial conclusions: 1) the proposed unified pooling framework is powerful to guide designment of new pooling operations which provides more choices for CNNs except common used hand-crafted pooling; 2) new designed pooling methods based on devised framework perform favourably against widely used max or average pooling; 3) the presented active selection pooling outperforms other pooling approaches, and it is able to choose pooling operation automatically through a sample-based active selection strategy which makes CNNs operate more closer to human visual system.

References

[1] S. Thorpe, D. Fize, and C. Marlot, "Speed of processing in the human visual system", Nature, vol. 381, pp. 520-522, 1996.

[2] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition", Neural computation, 1989.

[3] G. Backer, B. Mertsching, and M. Bollmann, "Data and model driven gaze control for an active vision system", IEEE Trans. Pattern Anal. Mach. Intell., vol. 23, no. 12, pp. 1415-1429, 2001.

[4] K. M. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", CVPR, June, 2016.

[5] J. Hu, L. Shen, G. Sun, "Squeeze-and-excitation networks", arXiv preprint arXiv:1709.01507, 2017.

[6] K. Gu, Z. Xia, J. Qiao, and W. Lin, "Deep dual-channel neural network for image-based smoke detection," *IEEE Trans. Multimedia*, vol. 22, no. 2, pp. 311-323, Feb. 2020.

[7] K. Gu, Z. Xia, and J. Qiao, "Stacked selective ensemble for $PM_{2.5}$ forecast," *IEEE Trans. Instrumentation and Measurement*, vol. 69, no. 3, pp. 660-671, Mar. 2020.

[8] K. Gu, H. Liu, Z. Xia, J. Qiao, W. Lin, and D. Thalmann, "$PM_{2.5}$ monitoring: Use information abundance measurement and wide and deep learning," *IEEE Trans. Neural Networks and Learning Systems*, 2021.

[9] M. Alencastre-Miranda, R. M. Johnson, and H. I. Krebs, " Convolutional neural networks and transfer learning for quality inspection of different sugarcane varieties," *IEEE Trans. Ind. Inform.*, vol. 17, no. 2, pp. 787-794, Feb. 2021.

[10] Y. Djenouri, G. Srivastava, and J. Lin, "Fast and accurate convolution neural network for detecting manufacturing data," *IEEE Trans. Ind. Inform.*, vol. 17, no. 4, pp. 2947-2955, Apr. 2021.

[11] M. Putro, L. Kurnianggoro, and K. Jo "High performance and efficient real-time face detector on central processing unit based on convolutional neural network," *IEEE Trans. Ind. Inform.*, vol. 17, no. 7, pp. 4449-4457, Jul. 2021.

[12] C. Y. Jang, S. Kim, K. R. Cho, and Y. H. Kim, . "Performance analysis of structural similarity-based backlight dimming algorithm modulated by controlling allowable local distortion of output image", Displays, vol. 59, pp. 1-8, 2019.

[13] M. Sun, Z. Song, X. Jiang, J. Pan, and Y. Pang, "Learning pooling for convolutional neural network", Neurocomputing, S0925231216312905, 2016.

[14] C. Y. Lee, P. W. Gallagher, and Z. Tu, "Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree", In Artificial intelligence and statistics, pp. 464-472, 2016.

[15] M. D. Zeiler, and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks", arXiv preprint arXiv:1301.3557, 2013.

[16] X. T. Tang, J. Yao and H. F. Hu , "Visual Search Experiment on Text Characteristics of Vital Signs Monitor Interface", Displays, vol. 62:101944, 2020.

[17] Y. T. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnn models for fine-grained visual recognition", CVPR, PP. 1449-1457, 2015.

[18] Z. Yu, J. Yu, C. Xiang, J. Fan, and D. Tao, "Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering", IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 12, pp. 5947-5959, 2018.

[19] L. Liu, C. Shen and A. V. D. Hengel, "Cross Convolutional Layer Pooling for Image Recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 11, pp. 2305-2313, Nov. 2017.

[20] Y. Cui, F. Zhou, J. Wang, X. Liu, Y. Lin, and S. Belongie, "Kernel pooling for convolutional neural networks", CVPR, pp. 2921-2930, 2017.

[21] X. H. Lian, Y. W. Pang, J. G. Han, J. Pan, "Cascaded hierarchical atrous spatial pyramid pooling module for semantic segmentation", Pattern Recognition, vol. 110, 2021.

[22] Y. D. Zhang, C. Pan, X. Chen, and F. Wang, "Abnormal breast identification by nine-layer convolutional neural network with parametric rectified linear unit and rank-based stochastic pooling", Journal of computational science, vol. 27, pp. 57-68, 2018.

[23] M. Emoto, "Depth perception and induced accommodation responses while watching high spatial resolution two-dimensional TV images", Displays, vol. 60, pp. 24-29, 2019.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions", Proc. of Computer Vision and Pattern Recognition (CVPR), Jun. 2015.

[25] S. Woo, J. Park, J. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module", arXiv, 2018.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks",CoRR, abs/1603.05027, 2016.

[27] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks", ICML, pp. 1126-1135, 2017.

[28] Z. Ma, D. Chang, J. Xie, Y. Ding, S. Wen, X. Li, and J. Guo, "Fine-grained vehicle classification with channel max pooling modified CNNs", IEEE Trans. Veh. Technol., vol. 68, no. 4, pp. 3224-3233, 2019.

[29] K. Gu, Y. Zhang, and J. Qiao, "Ensemble meta-learning for few-shot soot density recognition," *IEEE Trans. Ind. Inform.*, vol. 17, no. 3, pp. 2261-2270, Mar. 2021.

[30] B. Su, H. Chen, P. Chen, G. Bian, K. Liu and W. Liu, "Deep Learning-Based Solar-Cell Manufacturing Defect Detection With Complementary Attention Network," IEEE Trans. Ind. Inform., vol. 17, no. 6, pp. 4084-4095, June 2021.

[31] N. Martinel, M. Dunnhofer, R. Pucci, G. L. Foresti and C. Micheloni, "Lord of the Rings: Hanoi Pooling and Self-Knowledge Distillation for Fast and Accurate Vehicle Re-Identification," IEEE Trans. Ind. Inform., 2021.

[32] J. Ye, Y. Zhang, Q. Yang and C. Liu, "Joint stroke classification and text line grouping in online handwritten documents with edge pooling attention networks", Pattern Regnition, vol. 112, Apr. 2021.

[33] S. Zagoruyko, N. Komodakis, "Wide residual networks". arXiv preprint arXiv:1605.07146, 2016.

[34] S. H. Wang, Y. D. Lv, Y. Sui, S. Liu, S. J. Wang, and Y. D. Zhang, "Alcoholism detection by data augmentation and convolutional neural network with stochastic pooling", Journal of medical systems, vol. 42, no. 1, 2018.

[35] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu, "Semantic segmentation with second-order pooling", ECCV, pp. 430-443, Springer, Berlin, Heidelberg, October, 2012.

[36] S. Sabour, N. Frosst, G. E. Hinton, "Dynamic Routing Between Capsules", Advances in Neural Information Processing Systems, vol. 30, 2017.