Deep learning for visual understanding

Guo, Y.; Guo Y.

Chapter 4

Bag of Surrogate Parts Feature for Visual Recognition

Convolutional Neural Networks (CNNs) have attracted significant attention in visual recognition. Several recent studies have shown that, in addition to the fully-connected layers, the features derived from the convolutional layers of CNNs can also achieve promising performance in image classification tasks. In this chapter, we propose a new feature from the convolutional layers, called Bag of Surrogate Parts (BoSP), and its spatial variant, Spatial-BoSP (S-BoSP). The main idea is, we assume the feature maps in the convolutional layers as surrogate parts, and densely sample and assign image regions to these surrogate parts by observing the activation values. Together with BoSP/S-BoSP, we further propose another two schemes to enhance the performance: scale pooling and global-part prediction. Scale pooling aims to handle the objects with different scales and deformations, and global-part prediction combines the predictions of global and part features. By conducting extensive experiments on generic object, fine-grained object and scene datasets, we find the proposed scheme can not only achieve superior performance to the fully-connected feature, but also produce competitive, or in some cases remarkably better performance than the state-of-the-art.

4.1 Introduction

Recently, convolutional neural networks (CNNs) have been widely used in visual recognition evaluations and achieved top tier performance on international benchmark datasets [226]. There have emerged some well-known CNN models, such as AlexNet [14], VGG [24], GoogLeNet [25] and ResNet [227]. It has been proved that these models, pretrained on ImageNet [149], can be employed as universal models and transferred to other visual recognition tasks [45, 228, 229].

In general, the CNN architecture consists of alternatively stacked convolutional layers and pooling layers, followed by several fully-connected layers. Initially, when utilizing the off-the-shelf CNN models, researchers tend to extract the image representation from the fully-connected layers, as they are reported to produce better results than the convolutional layers [22, 204]. However, compared with the fully-connected layers, there are several inherent advantages of the convolutional layers [230]. First, the activations of the convolutional layers contain more spatial information, because each spatial unit on the convolutional feature maps corresponds to one receptive field on the input image. Second, the convolutional features can be extracted from an image of any size and aspect ratio. Third, it has been demonstrated that the convolutional layers contain rich semantic information [231]. Owing to these promising advantages, many recent studies [230–235] have shifted to fully exploit the benefits of the convolutional layers.

A typical usage of the convolutional layers is to encode the convolutional features with the Bag-of-Words (BoW) variants, such as VLAD [5] and Fisher Vector [6]. This pipeline can not only preserve the high discrimination of the CNN activations, but also utilize the 'bag' conception to improve the invariance property to scale changes, location changes and occlusions. In this work, we also intend to generate features within the BoW framework, and accordingly propose a new feature, called Bag of Surrogate Parts (BoSP). The essential idea is: we assume the feature maps in the convolutional layers as surrogate parts, and define the activation values on the feature maps as assignment strengths for these surrogate parts. As each spatial unit on the feature maps corresponds to one image local region, the one-by-one processing of these spatial units acts like densely sampling

and assigning image regions. The final feature is generated by summing up the assignment strengths of different regions on the surrogate parts.

In comparison with prior research [70, 170, 230, 233, 235, 236] which also attempted to incorporate BoW and CNN, BoSP has several differences: First, BoSP does not need to generate the visual codebook, since the surrogate parts have already been inherently determined by the structure, i.e. the feature maps. This eliminates the time-consuming and sensitive process of visual dictionary learning. Second, in contrast to the features encoded by the variants of BoW [70, 230], BoSP is relatively in low dimension, making it advantageous in processing large scale datasets. Third, the surrogate parts are more semantically meaningful than the statistically clustered visual words. In Figure 4.1, we choose two images from SUN397 [237] and Indoor67 [221], and overlay some feature maps on the original images for visualization. As can be seen, the activated regions of the sampled feature maps indicate some semantically meaningful regions. For example, the activated region in top-left corner and top-right corner correspond to the 'table' and 'bed', respectively. A similar finding has also been presented in [231].

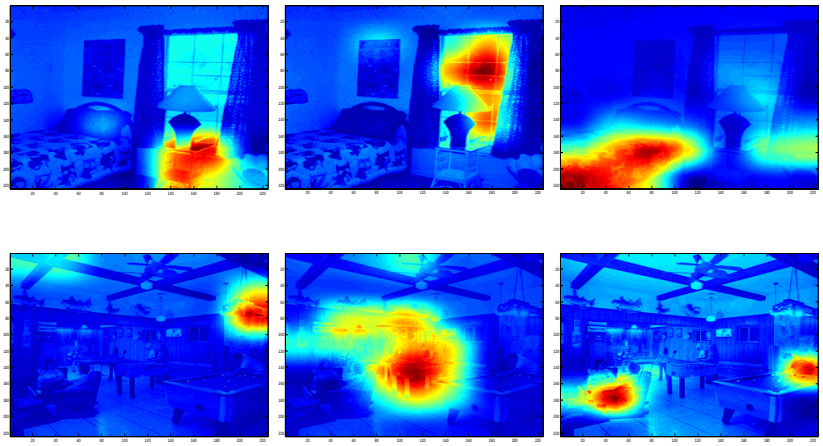

Figure 4.1: The visualization of the feature maps extracted from the last pooling layer of VGG [24].

Along with BoSP, there are three other contributions: (1) To incorporate more spatial information, we propose Spatial-BoSP (S-BoSP), by dividing the input image into several regions and concatenating the BoSP inside each region. (2)

To deal with the objects of different sizes and deformations, we develop a scale pooling scheme for the assignment of the spatial units, which could improve the performance considerably without enlarging the feature dimension and does not introduce much extra computational cost. (3) To make a more comprehensive prediction, we propose to multiply the global and part predictions, which constantly demonstrates better results than the individuals.

4.2 Related Work

Part-based representation. Parts can be used as mid-level visual elements to promote the object recognition process, and part-based approaches for object recognition have recently received much interest. Generally, the part-based methods can be viewed as a two-stage problem [238]. First, discover a collection of informative parts, and then train classifiers to detect the response of these parts. For example, Singh et al. [239] proposed an unsupervised method to find mid-level parts, by iteratively clustering HoG features and training classifiers. Juneja et al. [240] utilized the image-level labels to find the most discriminative parts based on entropy-rank curves, and employed BoW-based model to encode the features. Along with the promising achievements, the two-stage approaches also suffer from one possible drawback: the learned parts are not guarantied to be optimal for the classification task. As a consequence, several works [238, 241] suggested to jointly learn the parts and category models. On the other hand, Liu et al. [231] developed a scheme that did not explicitly define and detect the parts, but took the feature maps as the indicator maps of the parts, and concatenated the local features of parts as the image-level feature. This approach delivered encouraging results with a fraction of computational cost. Our method can be viewed as the combination of [231] and [240], as we do not explicitly define the parts either, and utilize the Bag of Parts model in [240] to represent the surrogate parts in [231].

BoW-based schemes with deep CNN feature. BoW-based methods have been widely used in previous researches, and achieved state-of-the-art perfor-

mance in various computer vision systems. In recent years, several studies [70, 170] attempted to introduce the deep CNN feature into the well-known BoW framework, especially its variants, such as VLAD [5] and Fisher Vector (FV) [6]. For instance, Gong et al. [170] extracted multiple fully-connected activations from three scales, and encoded them with VLAD scheme. Similarly, Yoo et al. [70] also extracted multi-scale top layer activations, but encoded them using the Fisher kernel. Additionally, Wu et al. [242] argued that the performances of FV and VLAD might fluctuate when different instant vectors were used, and proposed a more robust D3 (discriminative distribution distance) method. Although the aforementioned approaches achieved encouraging performance on various datasets, they all evolved a sensitive and computationally expensive process, i.e. learning the feature codebook. In comparison, the BoSP/S-BoSP are inherent features of the architecture which can be generated without manual tuning. This avoids the sensitive and time-consuming process of dictionary learning.

Typical usage of convolutional features. The convolutional features can be generally leveraged in two approaches. In the first approach, researchers encode the convolutional features with the variants of BoW scheme, such as VLAD and FV. For instance, Ng et al. [235] employed VLAD to encode the convolutional features and demonstrated that the intermediate layers could deliver better results for the image retrieval task than the top layers. In contrast, Cimpoi et al. [233] and Wei et al. [230] utilized FV to encode the descriptors from the intermediate layers, and also achieved promising performance. In the second approach, researchers take advantage of the convolutional activations in a more straightforward way, by aggregating and compressing them into the final representation. For example, Babenko et al. [232] aggregated the convolutional features in a simple sum-pooling way, and achieved a substantial boost in the performance. Liu et al. [231], on the other hand, took the feature maps as the indicator maps of parts, and aggregated the local features of each surrogate part as the image-level representation. Our method can be seen as the combination of these two approaches, in which we regard the feature maps as surrogate parts, similar to [231], but we do not explicitly concatenate the features of these parts. Instead, we only aggregate their statistical strengths, which makes the feature dimension much lower.

4.3 Bag of Surrogate Parts Feature

In this section, we first describe our proposed BoSP feature, and then give the interpretation of the surrogate part.

4.3.1 Bag of Surrogate Parts Feature

Generally, the CNN structure consists of multiple layers. Given an image, it will pass through several convolutional and pooling layers and generate various feature maps. We name the activation values on the feature maps as spatial units. As shown in Figure 4.2, one image region corresponds to multiple spatial units located in the same position on different feature maps, and the spatial units in higher feature maps have larger receptive fields than the lower ones.

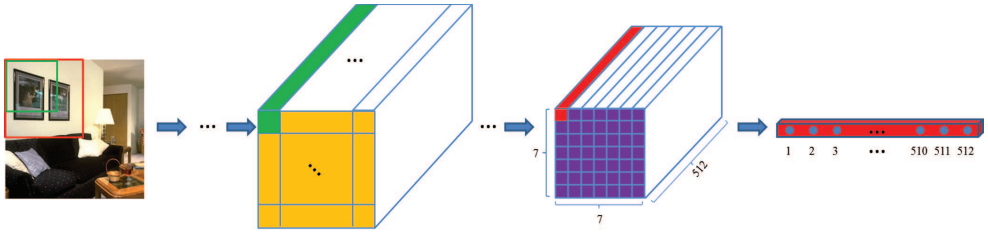

Figure 4.2: The illustration of the spatial unit. The local activations in green&red color are the responses of the green&red box surrounded image regions. For the VGG pool_5 layer, each image local region corresponds to 512 spatial units.

Intuitively, larger receptive field contains more semantic information. Therefore, we extract the BoSP feature from relatively higher layers (i.e. the 4th and the 5th pooling layer of VGG [24]. We simplified them as pool_4 and pool_5), as demonstrated in Figure 4.3. For the sake of clarity, we explain our method based on the pool_5 layer (For pool_4 layer, we first make an average pooling using 2×2 kernel with the stride 2, and then utilize the same operation with the pool_5 layer).

The specific procedure to extract BoSP is: we regard the feature maps as surrogate parts and assume that the activation values on the feature maps represent

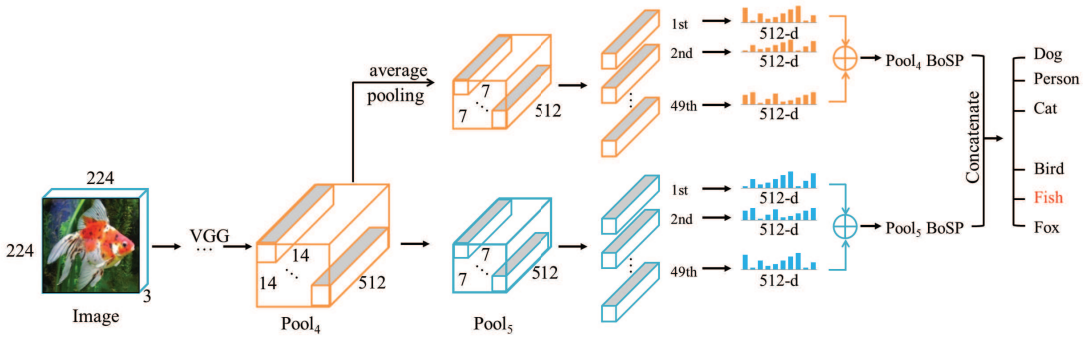

Figure 4.3: The framework of utilizing BoSP for image classification. We extract BoSP from the pool_4 and pool_5 layers of VGG. We first make an average pooling of pool_4 layer using 2×2 kernel with the stride 2, and then calculate the BoSP features of pool_4 and pool_5 layers. The final feature is the concatenation of pool_4 BoSP and pool_5 BoSP. $\bigoplus$ means element-wise addition of the vectors.

the assignment strengths for the surrogate parts. Therefore, given the architecture, the number of the surrogate parts is inherently determined, which equals the number of feature maps. For the spatial units on the feature maps, we can calculate their assignment strengths for the surrogate parts by observing the activation values. The one-by-one processing of these spatial units can be viewed as densely sampling and assigning spatial regions of the input image. Finally, we sum the assignment strengths for the surrogate parts and form a vector accordingly, i.e. BoSP, whose length equals the number of the feature maps.

More in detail, suppose there are M feature maps and each feature map contains n spatial units, then we have M surrogate parts and can densely sample n regions from the input image (for the pool_5 layer of VGG, $M = 512, n = 49$). The BoSP for this image can be writen as Eq 4.1:

$$BoSP = \sum_{i=1}^{n} [P_1^i, P_2^i, \cdots, P_j^i, \cdots, P_M^i] \tag{4.1}$$

P_j^i represents the assignment strength of i^{th} region on j^{th} surrogate part.

To explicitly restrict the membership of the surrogate parts to [0,1], we normalize the local activations by diving the largest component of the vector, and take the

normalized activations as the assignment strengths, i.e.

$$P_j^i = \frac{A_j^i}{\max_j(A_j^i)} \tag{4.2}$$

A_j^i means the j^{th} element of the activation A^i (It would always be non-negative since the activations are extracted after the ReLU layer).

To avoid using unreliable likelihood, we further adopt the idea of localized soft assignment [243] and only keep the assignment strengths with large values, and modify Eq 4.2 as Eq 4.3:

$$P_j^i = \begin{cases} 0 & \text{if } A_j^i < \text{mean}_j(A_j^i) \\ \frac{A_j^i}{\max_j(A_j^i)} & \text{if } A_j^i \geq \text{mean}_j(A_j^i) \end{cases} \tag{4.3}$$

The proposed BoSP has the following advantages: (1) It is efficient to be extracted. The feature is derived from the convolutional layers, which contain fewer parameters and need fewer computations. In practical, we only need to add up the larger normalized activation values of the feature maps. (2) It is less-specialized to be extracted. The dimension and the assignment strengths for the surrogate parts can be directly generated from the activation values, and there is no parameter for us to tune, which enhances its generality. (3) It is relatively low-dimensional. There are 512 feature maps in the pool_5 layer of VGG, so the dimension of BoSP from this layer is 512, much smaller than the schemes which need to concatenate convolutional features [230, 231, 234].

4.3.2 Interpretation of the Surrogate Part

In Figure 4.1, we have visualized that the feature maps can be viewed as surrogate parts. In this subsection, we try to give more insights into these surrogate parts by analyzing their influences on the categorization.

Similar to [244], we utilize regularized logistic regression method to make the prediction because it is faster and can help to evaluate the importance of the surrogate parts explicitly.

Logistic Regression assumes the probability for a binary classification satisfies:

$$\log \frac{p(y=1|x;\beta,w)}{p(y=-1|x;\beta,w)} = \beta + \sum_{j=1}^{d} w_j x_j \tag{4.4}$$

Where $p(y=1|x;\beta,w) + p(y=-1|x;\beta,w) = 1$, x indicates the extracted BoSP feature vector, and β, w are the parameters to learn.

From Eq 4.4, we can deduce that

$$p(y=1|x;\beta,w) = \frac{1}{1+\exp(-[\beta + \sum_{j=1}^{d} w_j x_j])} \tag{4.5}$$

By considering $\beta = w_0$ and $x_0 = 1$, Eq 4.5 can be rewritten as:

$$p(y=1|x;\beta,w) = \frac{1}{1+\exp(-w^T x)} \tag{4.6}$$

The optimal parameter w is obtained by optimizing the conditional log-likelihood function[245]:

$$\widehat{w} = \underset{w}{\operatorname{argmax}} \log \prod_i p(y_i|x_i;w) = \underset{w}{\operatorname{argmin}} \sum_i \log(1+\exp(-y_i w^T x_i)) \tag{4.7}$$

In this work, we use a L2-regularization term to restrict large values, as written in Eq 4.8:

$$\widehat{w} = \underset{w}{\operatorname{argmin}} \sum_i \log(1+\exp(-y_i w^T x_i)) + \lambda w^T w \tag{4.8}$$

Where $\lambda > 0$ is the regularization parameter.

Figure 4.4 demonstrates the learned weights for two categories (*Faces_ easy* and *beaver*) in the Caltech101 dataset, which correspond to the best-performing category (the accuracy of *Faces_ easy* is 100%) and worst-performing category (the accuracy of *beaver* is 62.5%) for the global prediction of BoSP. We can observe that, for different categories, the weights of the surrogate parts are different. A high positive value means the related surrogate part contributes a lot to the positive class, while a high negative value means the corresponding surrogate part

contributes a lot to the negative class. The surrogate classes with very small weights do not distinguish well between the positive and negative classes. For clarity, we denote the surrogate part which has the largest weight as *pos_part*, while the surrogate part with the smallest weight as *neg_part*.

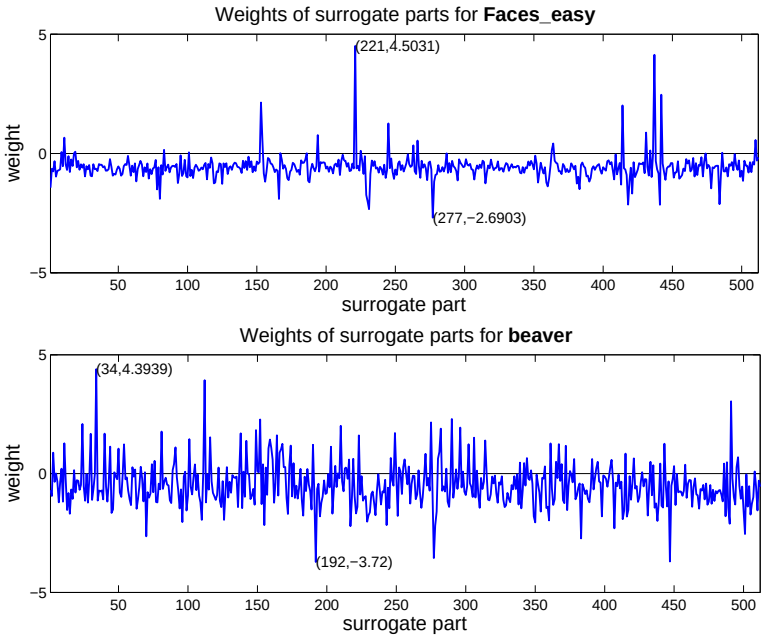

Figure 4.4: Learned weights for *Faces_easy* and *beaver* in Caltech101.

Ideally, given an input image, it should have large response in the *pos_part* of its groundtruth category, while have few response in the corresponding *neg_part*. To verify this, we select one positive image (i.e. correctly classified) from *Faces_easy* and one negative image (i.e. wrongly classified) from *beaver*. For these two images, we first draw the feature maps corresponding to the *pos_part* and *neg_part*, and then overlay the feature maps to original images for better visualization, as shown in Figure 4.5. The lightless on the feature maps represents the image's response on this surrogate part.

We can notice that, for the image of category *Faces_easy*, quite a lot of regions are assigned to its correct *pos_part*, and few regions are assigned to its *neg_part*, this contributes to its correct classification. In contrast, the wrongly classified image from category *beaver* contains quite a lot of regions assigned to its *neg_part*,

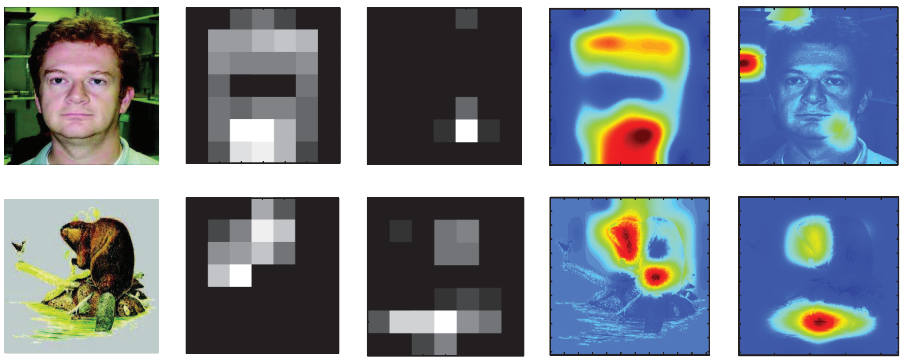

Figure 4.5: Visualization of the input images and the feature maps for the pos_part and neg_part. Pos_map and neg_map correspond to the feature maps which have the largest and smallest weight. Pos_overlay and neg_overlay demonstrate the activated regions when we overlay the corresponding feature maps to original images.

which leads to its mis-classification. Furthermore, when we overlay the feature maps to the original images, we found that these surrogate parts are semantically meaningful, and we could observe the image regions that affect the most for the classification. For example, the *pos_part* for the *beaver* image is located around the head area, while the *neg_part* corresponds to the 'floats', and the 'floats' contributes the most to its bad performance.

4.4 Enhancement schemes

In this section, we describe a spatial variant of BoSP, and propose two schemes to enhance the performance: scale pooling and global-part prediction.

4.4.1 Spatial BoSP

Motivated by the well-known spatial pyramid matching (SPM) method [135], we raise a spatial variant of BoSP, called S-BoSP. Specifically, we partition the image equally into multiple sub-regions (9 regions in 3 rows and 3 columns in

this chapter), calculate the BoSP inside each region and concatenate them into a single feature vector. Therefore, the dimension of S-BoSP is 9 times of BoSP. For simplicity, we conduct the partitioning process on the feature maps, rather than directly on the input images. As the size of the feature maps for the pool_5 layer of VGG is 7×7 , we need to divide the feature maps in an overlapping motion, i.e. some of the spatial units would exist in multiple sub-regions. The difference between BoSP and S-BoSP is illustrated in Figure 4.6.

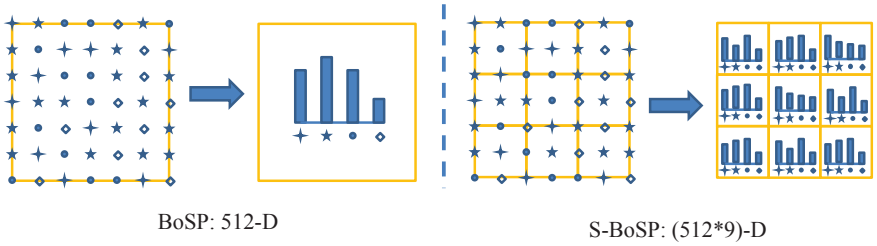

Figure 4.6: The illustration of BoSP feature and S-BoSP feature from the pool_5 layer of VGG (Different symbols represent different surrogate parts, and the histogram represents the assignment strength to these surrogate parts).

4.4.2 Scale Pooling

The BoSP/S-BoSP aforementioned only concern the spatial units at the finest level, and process them in a disjoint way, which means to sample and assign regions in input images with fixed size and position. However, the objects may appear in different positions, shapes and scales. The independent processing of the spatial units may capture different parts of the same object and have difficulties in classifying the objects of different scales. For example in Figure 4.7, the object 'water_lilly' from the two images appear in different scales. In this case, the fixed receptive region may capture different parts of the object. A 2×2 grid can capture most of the 'water_lilly' for the left image, while it can only capture some petals for the right image.

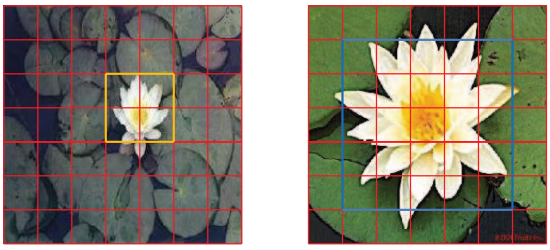

Figure 4.7: The demonstration of objects in different scale. For the left image, a 2×2 grid can capture most of the object, while the right image needs 6×6 grid to cover most of the object.

To address this problem, we proposed a scale pooling scheme. It can improve the assignment of objects with different scales and deformations by handling regions of different sizes and positions, together with the max pooling operation inside each region. The procedure of scale pooling is illustrated in Figure 4.8.

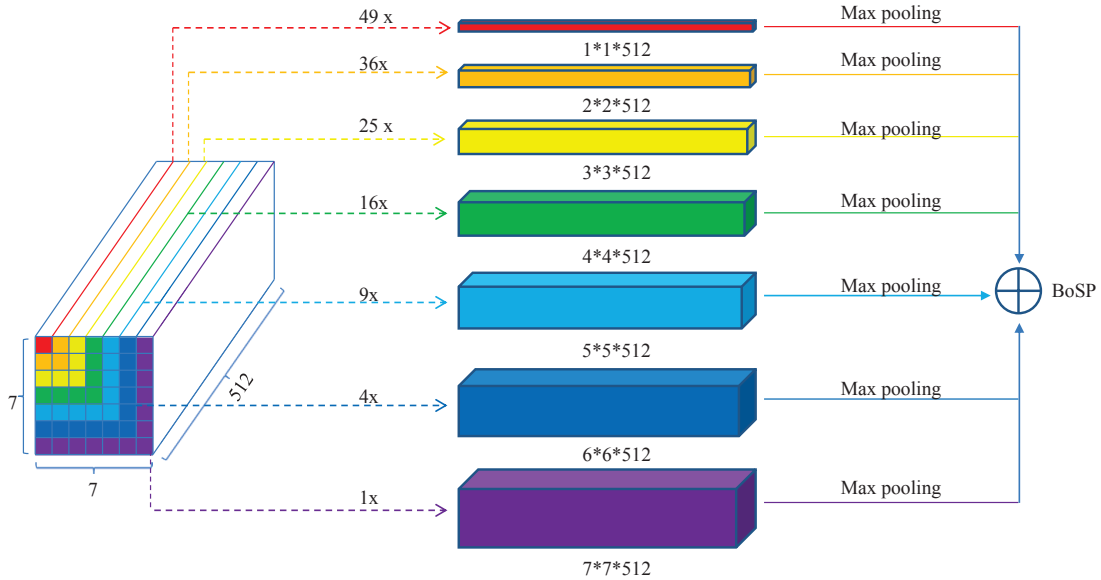

Figure 4.8: Pipeline of the scale pooling technique for BoSP. We can extract different number of features from 7 scales. For example, there are 49 red strips for the smallest scale, and only 1 purple strip for the largest scale. Next, we apply max pooling on the features inside each scale and calculate the BoSP individually, then add them up to form the final feature.

Specifically, we partition the activations from the pool_5 layer into 7 scales for BoSP. Under different scales, we derive spatial units of different numbers and different sizes. For clarity, we define the derived spatial units as coarse spatial units, and the coarse spatial units in scale 1 correspond to the original spatial units. Under scale i ($i \in [1, 7]$), we can derive $(8-i)^2$ coarse spatial units, and each coarse unit contains i^2 original spatial units. Therefore, the total number of the coarse spatial units is $\sum_{i=1}^{7}(8-i)^2 = 140$. Next, we pool these coarse spatial units and compute their assignment strengths for the surrogate parts by employing Eq 4.3. In this work, we utilize the max pooling operation inside each coarse spatial unit since it has been proved to be superior for capturing invariance in image-like data [40]. Finally, we sum the assignment strengths of the coarse spatial units under different scales together to form the refined assignment strengths for the image. For S-BoSP, we utilize the scale pooling scheme inside each sub-region, and then concatenate the resulting features together.

To demonstrate the effectiveness of the scale pooling, we visualize the feature without/with scale pooling in Figure 4.9 using t-SNE technique [246], and we can see that the feature with scale pooling is more distinguishable.

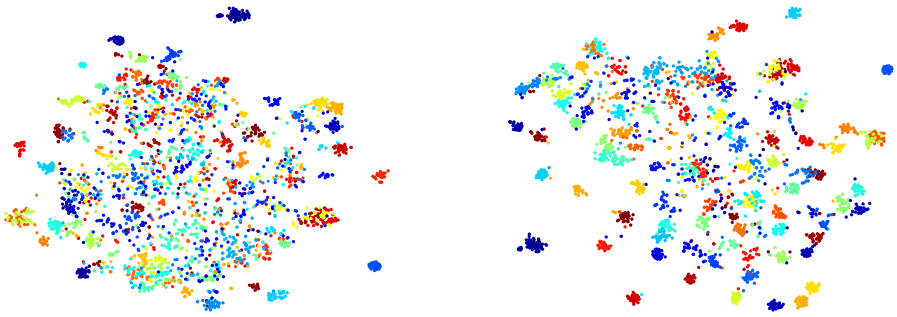

Figure 4.9: The visualization of feature without(left)/with(right) scale pooling for the Caltch101 dataset. Different symbol colors represent different categories.

Overall, the scale pooling scheme is proposed to make the assignment of receptive fields more comprehensive, by handling the image regions across different scales and positions. Benefiting from the max pooling operation, the scheme is robust

to object deformation inside each coarse spatial unit. In addition, scale pooling does not increase the feature dimension, nor does it have a significant effect on the computational efficiency.

4.4.3 Global-Part Prediction

For a given image, we first resize it to 224×224, and then extract its global feature by utilizing VGG. However, in many cases, extracting only one global feature from the input image is not discriminative enough, and many recent works [70, 170, 233, 235] proposed to extract multiple features from one single image, and generate a more comprehensive feature by integrating these features in a certain way. Without extra data, one common approach is to generate numerous sub-images from the input image, and average the sub-image features as augmented image feature. Although the augmented feature contains more information, it only considers individual parts of the input image, and fails to handle the input image entirely. To make a more comprehensive prediction, we propose to combine the predictions of the global feature and the augmented feature, as shown in Figure 4.10.

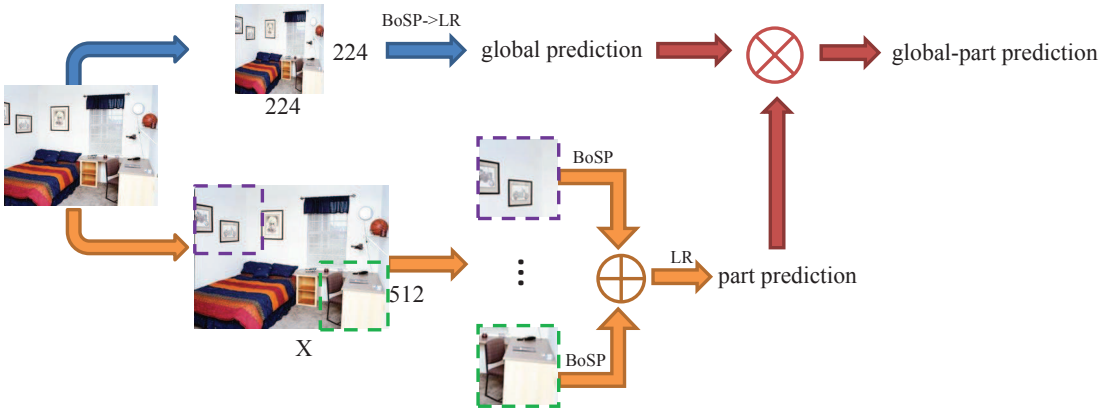

Figure 4.10: The illustration of global-part prediction. The global prediction is achieved by utilizing the global feature. The part prediction is achieved by averaging the parts' features. The global-part prediction is the product of the global prediction and the part prediction.

The specific procedure is: given an image, we first resize it to 224×224, and extract its global feature. This feature focuses more on the whole image, and we can compute the global prediction based on it, denoted as Pre_{global}; Next, we resize the image to make the smallest side equal to S while keeping its ratio, and crop regions of 224×224 with the stride of 32 pixels. Thereby, we formulate several sub-images from the original image, and each sub-image may only contain part of the original object. The image feature is the average of the sub-images' features, as is the same with general approach. This feature focuses more about parts of the image, and we can make the part prediction based on it, denoted as Pre_{part}. The global-part prediction is the multiplication of the global prediction and the part prediction:

$$Pre_{global-part} = Pre_{global} \times Pre_{part} \tag{4.9}$$

As our feature is obtained from the convolutional layers, the input image could be of any size, and we do not need to explicitly crop sub-images. In practice, we only need to input the resized image once to extract the part features.

4.5 Experiments

To evaluate the performance of our method, we conduct a series of experiments on four datasets, Caltech101 [220], Oxford 102 Flowers (referred to as Oxford102) [247], MIT Indoor67 (referred to as Indoor67) [221] and SUN397 [237], which cover several popular topics in image classification, i.e. generic object classification, fine-grained object classification, and scene classification. Some of example images are shown in Figure 4.11. The details of the datasets are described below:

Caltech101 consists of 9144 images in 102 object categories (101 object classes and a background class). The image number per category ranges from 31 to 800. For each category, we randomly select 30 images for training and test on up to 50 images. There are 44 'overlap' images of the Caltech101 dataset and ImageNet training data. We exclude these images from the test set.

Figure 4.11: The example images for the four datasets.

Oxford102 has 102 flower categories and a total of 8189 images. Each category contains 40 to 258 images. The flowers appear under various scales, pose and illuminations. For each class we use 20 images for training and the rest for testing.

Indoor67 contains 15620 images in 67 indoor categories. We use the standard train/test split provided in [221], which consists of 80 training and 20 test images per category.

SUN397 is a large scale scene dataset from a collaboration between MIT and Brown University. It contains more than 100K images for 397 categories and is generally considered to be at a high difficulty level and very challenging. Each category has at least 100 images. The standard training/test splits are available from [237], and each split contains 50 training and 50 test images per category. We average the results of the 10 public splits as the final classification accuracy.

For all the experiments, we employ VGG Net-D [24] as the pre-trained CNN model to extract features. The model is implemented by the Caffe [218] package. For simplicity, pre-trained model weights are kept fixed without fine-tuning. All of the BoSP/S-BoSP features are L2 normalized before the experiment. The LR classifier is implemented by utilizing the open source library: LIBLINEAR [248].

For the global-part prediction, we resize the images of Caltech101 to let its smallest side $S = 256$, while $S = 512$ for Oxford102, Indoor67 and SUN397.

4.5.1 Analysis of our method

4.5.1.1 Which classifier to use?

It has been demonstrated that the features delivered from CNN models are highly discriminative, and can be combined with classifiers to boost the image classification performance [24, 231]. However, most of these researches, if not all, adopt linear SVM (LSVM) classifier for their tasks, and ignore other classifiers which may corporate better with the features.

As we have demonstrated that each of the convolutional layers can be viewed as a surrogate part, we assume that it would be more reasonable if we can explicitly consider the differences among the feature maps, and propose to utilize the L2-regularized Logistic Regression (LR) classifier to handle our feature.

To verify our assumption, in this section, we compare the performance of three classifiers, i.e. LSVM, histogram intersection kernel SVM (HIKSVM) [249] and LR. All of the classifiers take the global BoSP feature derived from the pool_5 layer as the input. The results are shown in Table 4.1.

Table 4.1: The comparison of the accuracy and efficiency (training/test time) for different classifiers.

	LSVM	HIKSVM	LR
Caltech101	86.68%	87.01%	**88.28%**
time(s)	19.17+16.27	35.16+34.39	6.46+0.13
Oxford102	72.42%	80.84%	**81.28%**
time(s)	10.24+27.35	13.39+40.22	3.51+0.22
Indoor67	67.46%	69.40%	**69.48%**
time(s)	43.92+13.14	107.84+26.13	13.49+0.05
SUN397	51.23%	53.31%	**53.68%**
time(s)	587.96+954.74	1791.90+1969.10	320.62+2.01

In terms of accuracy, both the HIKSVM and LR perform better than the commonly used LSVM, demonstrating that we could have more options for the classifier, aside from LSVM. Particularly, the improvement of LR over LSVM is quite remarkable, from 1.6% to 8.86%. This verifies our assumption that, it is more reasonable to explicitly consider the differences among the feature maps.

In terms of efficiency, as HIKSVM needs to build non-linear kernels, it is the most computationally expensive, both for training and testing. Compared to these two classifier, LR is significantly faster due to its simple operations.

Owing to the advantages of LR, we utilized the LR classifier in all the following experiments. We further investigate the influence of the regularization parameter λ, by ranging the values from 5 to 50. As is shown in Figure 4.12, the influence on the accuracy is negligible when we change λ, and for fair comparisons, we set a fixed regularization term $\lambda = 20$.

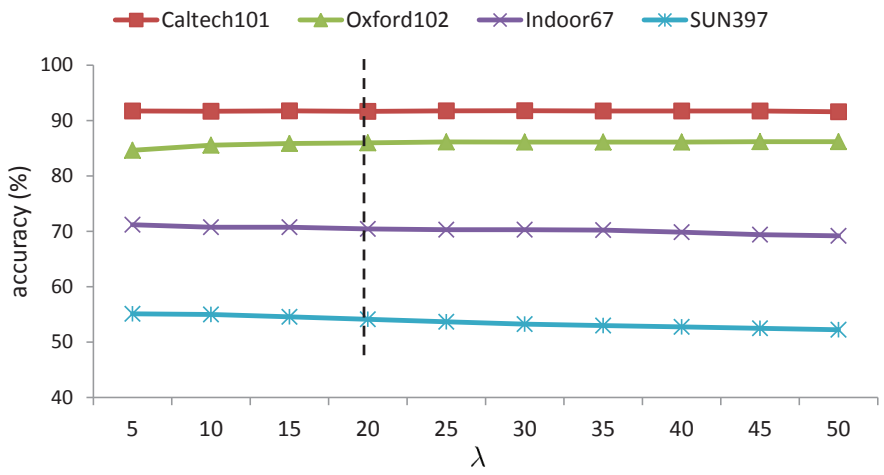

Figure 4.12: The performance of different regularization parameter values for training LR.

4.5.1.2 The comparison of BoSP from different layers

The proposed BoSP is achieved from the convolutional layers, and we can formulate multiple BoSP features from different layers of the network. Intuitively, deeper layer activations would contain more semantic information compared to

shallower layer activations, thus should deliver better performance. To verify this, we evaluated the accuracy and efficiency of global BoSP from different layers of VGG on the Caltech101 dataset.

From Figure 4.13, we can see that, in terms of accuracy, the performance of BoSP would increase along with the layer depth, in which the feature derived from the pool_5 layer obtains the best result. This phenomenon confirms our assumption on the advantage of using deeper layers.

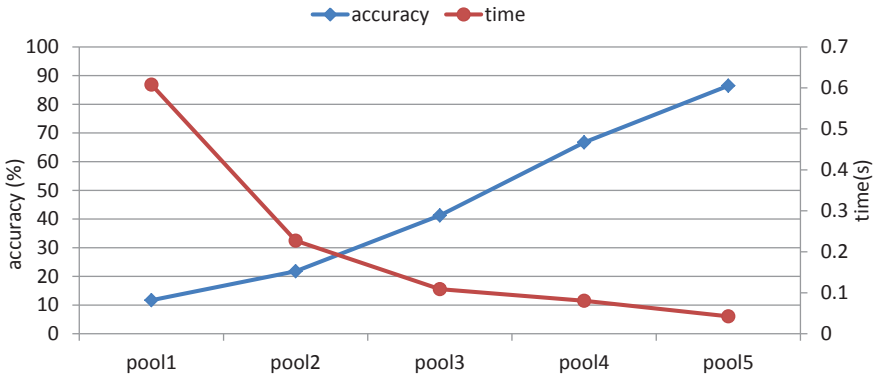

Figure 4.13: The performances of BoSP for different layers of VGG. In VGG, there are 5 major convolutional layers. We take the last sub-layer (i.e. the pooling layer) as the representative of the major layer.

As for the efficiency, it also improves with the layer depth, since the deeper feature maps are smaller than shallower ones, we need to assign fewer spatial units for the deeper layers. To combine the visual contents from different levels, we propose to extract features from pool_4 and pool_5 layers (Although concatenating more layers may further improve the performance, it would increase the feature dimension, thus is not good for large-scale data processing). To accelerate the feature extraction process from the pool_4 layer, we first make an average pooling using 2×2 kernel with the stride 2. In this way, the resulting feature maps from pool_4 layer share the same size with those from pool_5 layer.

4.5.1.3 Evaluation of the Scale Pooling

In this part, we first evaluate the benefits brought by the proposed scale pooling scheme, and then compare the BoSP/S-BoSP with the commonly used CNN feature. All the features are extracted after resizing the images to 224×224.

Figure 4.14 reveals the merit of scale pooling on BoSP. For all the four datasets, the BoSP extracted with scale pooling outperforms the corresponding BoSP without it, and the advantage can be very large. For instance, for the Caltech101 dataset, scale pooling increases the pool_4 BoSP, pool_5 BoSP and concatenated BoSP by 10.12%, 3.37% and 3.94% respectively. Moreover, scale pooling would not enlarge the feature dimension, which demonstrates its great potential.

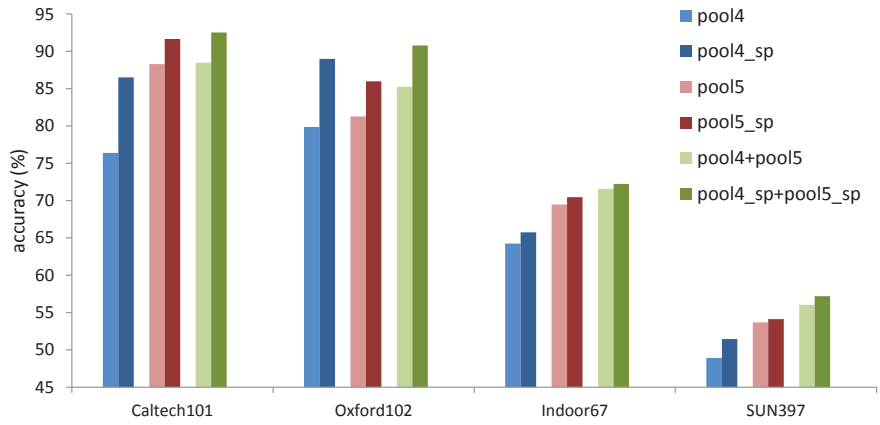

Figure 4.14: The comparison between BoSP with and without using scale pooling (The features using scaling pooling have the suffix: '_sp', and '+' means to concatenate features).

We further compare the proposed BoSP/S-BoSP with commonly used CNN feature (i.e. the activation from the last fully connected layer) in Table 4.2. As we can see, the BoSP/S-BoSP from the pool_5 layer could already achieve remarkably better performance than the CNN feature. After incorporating the BoSP from the pool_4 layer, the advantage become even more obvious. Take the Oxford102 as an example, the pool_5 BoSP brings 5.38% accuracy increase over the CNN feature, from 80.60% to 85.98%, while comes in much lower dimension (512 v.s.

4096). After concatenating the $pool_4$ BoSP, the improvement comes to 10.18% (from 80.60% to 90.78%).

Although the scale pooling is proposed to handle the objects of different scale for the BoSP/S-BoSP, technically, it can also be employed to the average pooling (AP). For a fair comparison, we list the AP with scale pooling in Table 4.2. We can see that BoSP/S-BoSP can achieve overall better performance than AP, especially for the concatenated features. A more detailed description of the relationship between BoSP and AP is presented in the discussion section.

Table 4.2: The comparison of AP/BoSP/S-BoSP with scale pooling and the CNN feature extracted from the last fully-connected layer.

	Dim	Caltech101	Oxford102	Indoor67	SUN397
CNN	4096	89.22%	80.60%	68.06%	53.26%
$pool_5$, with scale pooling:					
AP	512	91.62%	85.82%	69.63%	53.93%
BoSP	512	91.65%	85.98%	70.45%	54.12%
S-BoSP	4608	93.99%	85.54%	71.19%	55.42%
[$pool_4$, $pool_5$], with scale pooling:					
AP	1024	92.49%	90.4%	70.97%	56.04%
BoSP	1024	92.55%	**90.78%**	72.24%	57.19%
S-BoSP	9216	**94.09%**	89.92%	**73.21%**	**58.20%**

From Table 4.2, we can also conclude another two findings: (1) Compared with using the $pool_5$ BoSP/S-BoSP individually, concatenating the $pool_4$ feature would double the feature dimension, but is always beneficial to the accuracy. Remarkably, the increases on Oxford102 and SUN397 are about 5% and 3%, respectively. (2) Except for Oxford102, S-BoSP performs better than the corresponding BoSP on the other three datasets, which demonstrates the effectiveness of the spatial scheme.

4.5.1.4 Evaluation of the global-part prediction

In this subsection, we evaluate the proposed global-part prediction on both the BoSP and S-BoSP features. From the results in Table 4.3, it is obvious that the part prediction performs better than the global prediction, demonstrating that extracting multiple features in a single image is useful. Furthermore, we can also observe the advantage of global-part prediction over the global/part prediction: regardless of the differences between the global prediction and the part prediction, it is always beneficial to combine them by multiplication. Notably, the improvement on the predictions of SUN397 can be about 3%.

Table 4.3: The comparison of the different predictions on BoSP and S-BoSP. Pre_{global}: global prediction; Pre_{part}: part prediction; Pre_{g-p}: global-part prediction. The features are the concatenated features from pool_4 and pool_5 layer.

	Caltech101		Oxford102		Indoor67		SUN397	
	BoSP	S-BoSP	BoSP	S-BoSP	BoSP	S-BoSP	BoSP	S-BoSP
Pre_{global}	92.52%	94.09%	90.78%	89.92%	72.24%	73.21%	57.19%	58.20%
Pre_{part}	92.62%	94.12%	93.54%	92.94%	77.46%	77.31%	60.48%	60.97%
Pre_{g-p}	93.02%	**94.92%**	**94.02%**	93.10%	**78.21%**	78.13%	63.21%	**63.79%**

4.5.2 Comparison with the state-of-the-art

In Table 4.4, we list the comparison between our scheme and the published state-of-the-art schemes on the four datasets. All of the features are extracted based on the VGG network. We do not list the methods which employed additional information to improve classification, such as utilizing the part annotations for Oxford102, or using the large-scale scene-specific Places2 dataset[250] for Indoor67/SUN397.

We observe that, for the Caltech101 dataset, our scheme obtains the top accuracy. Notably, our BoSP feature from VGG Net-D achieves slightly better than the published result of VGG [24], which is 92.7%. However, their result is obtained by concatenating the fully-connected features from two models (VGG Net-D & VGG Net-E) and three scales ($S = 256, 384, 512$), making the dimension of feature

Table 4.4: The comparison with the state-of-the-art (All of the methods are based on the VGG Network. Best results are in bold face).

Method	Caltech101	Oxford102	Indoor67	SUN397
VGG [24]	92.7%	-	-	-
ONE [251]	-	86.82%	70.13%	54.87%
CrossLayer-OConv [234]	-	-	74.4%	-
CrossLayer-AConv [234]	-	-	78.2%	-
Deep19-DAG [252]	-	-	77.5%	56.2%
FV-CNN [233, 236]	-	-	**81%**	-
NML [253]	-	84.3%	-	-
BoE [254]	-	-	77.63%	-
D3(K=8) [242]	93.80%	-	77.76%	60.22%
Best Single [255]	-	-	76.42%	59.71%
Dual [255]	-	-	79.04%	61.07%
Bayesian LS-SVM [256]	93.3%	91.5%	77.8%	56.1%
SCDA [257]	-	92.1%	-	-
BoSP ($\text{Pre}_{global-part}$)	93.02%	**94.02%**	78.21%	63.21%
S-BoSP ($\text{Pre}_{global-part}$)	**94.92%**	93.10%	78.13%	**63.79%**

much larger than ours (12288 v.s. 1024). The proposed S-BoSP with global-part prediction further improves the state-of-the-art from 93.80% to 94.92%.

For the fine-grained Oxford102 dataset, the S-BoSP achieves inferior performance than BoSP, suggesting that the spatial scheme does not work for this dataset. We suspect this is because the small parts of the fine-grained objects are similar in appearance and do not distinguish well. Therefore, it is better to process the input as a whole image, without partitioning. Nevertheless, both BoSP and S-BoSP get better results than the state-of-the-art. Particularly, BoSP achieves considerable improvement over the previous best result, from 92.1% to 94.02%, with a dimension of 1024, demonstrating the effectiveness of our scheme.

For the Indoor67 dataset, the BoSP delivers competitive performance with the state-of-the-art (FV-CNN). In contrast to FV-CNN, BoSP has a much smaller (1K vs 64K) dimension, which can be a significant advantage in many situations. Compared with another recent work [255], our method achieves better

performance than its best single network, only slightly worse than its best dual architectures. However, the dual architectures can only be obtained after extensive comparisons, and it is not clear which two networks shall we choose before the experiment.

It is worth noting that [234] also takes the feature maps as indicator maps of surrogate parts, but it explicitly constructs and concatenates the features of each surrogate part to formulate the image feature. As a consequence, the resulting feature would be quite high dimensional, i.e. 262144-D for CrossLayer-OConv and 200000-D for CrossLayer-AConv. In contrast, we do not explicitly construct the surrogate part features and only focus on the assignment strength for these surrogate parts, making the feature dimension much lower. Nevertheless, we still get better performance than the CrossLayer-OConv, and competitive performance with the CrossLayer-AConv.

For the large scale, general scene classification on SUN397 dataset, our method obtains remarkably better performance than the current best result, improving the state-of-the-art from 61.07% to 63.79%.

4.6 Discussion

In this work, we regard the feature maps in convolutional layers as surrogate parts, and propose to utilize Eq 4.3 as the soft assignment for these surrogate parts. Under this assumption, we can also take advantage of other assignment schemes. For example, we have done some additional experiments to test the traditional soft assignment coding. The definition of traditional soft-assignment coding [258] is:

$$U_j^i = \frac{\exp(-\beta||D_j^i||)}{\sum_{k=1}^{n}\exp(-\beta||D_k^i||)} \tag{4.10}$$

Where U_j^i denotes the membership of the ith local feature to the jth visual word, and $||D_j^i||$ is the distance between them. β is the smoothing factor controlling the softness of the assignment.

In our scheme, the activation value has opposite assignment meaning with the distance, as a larger A_j^i means the ith local feature is more like to be assigned to the jth surrogate part. Therefore, we modify the standard soft assignment as:

$$U_j^i = 1 - \exp(-\beta A_j^i) \tag{4.11}$$

Note that we do not explicitly constrain $\sum_{k=1}^{n} U_k^i = 1$ since there are many cases in which $A^i = 0$.

For the pool_5 layer of VGG, when we set $\beta = 0.02$, we can get similar results with this work, as shown in Table 4.5. This further verifies the reasonableness of our assumption to regard the feature maps as surrogate parts, and demonstrates the extensibility of our scheme.

Table 4.5: The comparison of BoSP with different soft assignment schemes. The BoSP with superscript $*$ means that we use Eq 4.11 as the soft assignment for the surrogate parts.

	Caltech101	Oxford102	Indoor67	SUN397
BoSP	91.65%	85.98%	70.45%	54.12%
BoSP*	91.55%	86.31%	70.45%	54.29%

Comparison to average pooling: Operationally, the straightforward assignment Eq 4.3 is a special case of average pooling. When we do not normalize the local activations and do not make them sparse, the BoSP would evolve to AP. However, conceptually, BoSP and AP have different views of the feature maps. AP processes the feature maps individually by averaging the values on each feature map, while BoSP handles the image local regions individually by normalizing the local activations. When we utilize other soft assignment schemes, e.g. Eq 4.11, their fundamental differences would become clearer.

4.7 Conclusion and Future Work

We proposed a new feature from the convolutional layers of VGG, which is highly discriminative and can be efficiently extracted. Along with the feature, we further introduced another three schemes to enhance the performance: spatial aggregation, scale pooling and global-part prediction. In addition, we also explored the semantic meaning of the surrogate parts and combined the BoSP feature from different layers. Our experiments in several popular classification tasks demonstrated the success of our scheme.

In the future, we would extend our work in three possible directions: (1) We would extract the feature from more advanced network (e.g. Res [227]) for further improvement; (2) We intend to directly utilize the scale pooling scheme for training the deep network; (3) We would employ our proposed feature for different applications, such as object detection and image retrieval.