Spatial-Temporal Weighted Pyramid using Spatial Orthogonal Pooling

Yusuke Mukuta[1], Yoshitaka Ushiku[1], and Tatsuya Harada[1,2]

[1]The University of Tokyo, 7-3-1 Hongo Bunkyo-ku, Tokyo, Japan
[2]RIKEN Center for Advanced Intelligence Project, 1-4-1 Nihonbashi Chuo-ku, Tokyo, Japan
{mukuta,ushiku,harada}@mi.t.u-tokyo.ac.jp

Abstract

Feature pooling is a method that summarizes local descriptors in an image using spatial information. Spatial pyramid matching uses the statistics of local features in an image subregion as a global feature. However, the disadvantages of this method are that there is no theoretical guideline for selecting the pooling region, robustness to small image translation is lost around the edges of the pooling region, the information encoded in the different feature pyramids overlaps, and thus recognition performance stagnates as a greater pyramid size is selected. In this research, we propose a novel interpretation that regards feature pooling as an orthogonal projection in the space of functions that maps the image space to the local feature space. Moreover, we propose a novel feature-pooling method that orthogonally projects the function form of local descriptors into the space of low-degree polynomials. We also evaluate the robustness of the proposed method. Experimental results demonstrate the effectiveness of the proposed methods.

1. Introduction

In this paper, we consider feature pooling, which summarizes local features in one image into one global feature. When designing feature pooling, it is important for the global feature to contain rich information and be robust to small image translations. Spatial pyramid matching is the feature-pooling method that is most commonly used. It divides an image into subregions according to various resolutions and uses statistics of local features in each subregion, e.g., the mean and maximum values, as global features. However, there is no theoretical guideline for determining the pooling region. In addition, the global feature value changes discontinuously when the local feature strides over the edge of subregions. Also, the spatial pyramid matching representation is verbose because the different spatial pooling regions overlap. Moreover, we cannot obtain useful features when the resolution is too high because robustness to small translations is lost. Thus, we need a large pyramid size to extract spatial information.

To overcome these problems, we propose a novel feature-pooling method that uses the weighted averages of local features based on the position of the local features in an image. To determine the weights, we propose a novel viewpoint that regards local features in one image as a function. Local features have their own feature values associated with positions in the image. Thus, we can see a set of local features as a function from the image space to the local feature space whose output is the value of the local feature at the input position. With this interpretation, we can regard spatial pyramid matching as a projection into the space of piecewise constant functions based on the standard inner product. From this viewpoint, we derive novel pooling weights as orthogonal projections of this function form into the spaces of low-degree polynomials with certain inner products. We obtain this pooling weight by first calculating orthonormal basis of the spaces of low-degree polynomials with the inner products and then calculating the inner product of the delta functions with the basis. Since the pooling weights are polynomials of the position and thus smooth, the proposed global feature is robust to small image translations. Also, since spatial pooling weights are orthogonal with respect to the given metric, it is expected that we can extract spatial information effectively. The feature dimension and the amount of spatial information can be controlled by the degree of the polynomial space.

From the proposed framework, we first derive the spatial pooling weights of the spaces of low-degree polynomials with the standard inner product, which consist of the products of Legendre polynomials. To derive the pooling weights more robust to local translations than Legendre polynomials, we then propose a weighted pooling method that considers the function space with weighted inner products, which are more robust to local translations than the

standard inner product.

Experimental results using image recognition datasets and action recognition datasets show that the proposed methods demonstrate higher accuracy than spatial pyramid matching even when the pyramid size is small and are less saturated when the pyramid size increases.

The contributions of this paper are as follows:

- We demonstrate that spatial pyramid matching can be regarded as an orthogonal projection in the function space.
- We propose Spatial-Temporal Weighted Pyramid, which uses weighted averages as a global feature. The weight can be calculated as an orthogonal projection in the function space.
  - We propose a novel pooling method that uses Legendre polynomials, which can be regarded as an orthogonal projection into a low-degree function space.
  - We also propose a pooling method that uses orthogonal polynomials for weighted inner products, which are more robust to local translations than the standard inner product.

2. Related Works

Feature pooling is a method that combines local descriptors in an image into one global feature. The simplest strategy is average pooling, which uses the means of local descriptors as a global feature. Max pooling [23] is a method that is inspired by the human visual cortex and is used for coding methods using histograms such as Bag of Visual Words [5] and Locality-constrained Linear Coding [32]. Max pooling uses element-wise maximum values instead of the average of local descriptors as a global feature and has been shown to be more robust to noise. A theoretical analysis of these pooling methods was conducted in [3]. In [3], the method that uses the L_p norm of each dimension is proposed as a method that bridges between average pooling and max pooling. These pooling methods are compared exhaustively via experiments in [15].

Lazebnik *et al*. [19] highlighted the importance of using spatial information of local features in image recognition. As an approximation for the pyramid match kernel, Lazebnik *et al*. proposed spatial pyramid matching, which divides the input image into subregions with various resolutions and concatenates Bag of Visual Words [5] in each subregion to obtain the global feature. Spatial pyramid matching is also applied to global features with richer information, such as the Fisher vector (FV) [25], the vector of locally aggregated descriptors (VLAD) [12]. Though other methods can be combined with spatial pyramid matching, spatial pyramid matching using average pooling is standard in feature pooling. Thus, we consider average pooling in the next section. In addition, spatial pyramid matching is combined with convolutional neural networks (CNNs) [17] and demonstrates good performance [11].

As extentions of original spatial pyramid matching, Perronnin *et al*. [22] proposed the non-regular spatial pyramid matching that uses different spatial resolutions for x-axis and y-axis. Shahiduzzoman *et al*. [26] proposed to apply Gaussian blur to the input image before extracting local features. Koniusz & Mikolajczyk [14], Sanchez *et al*. [24] proposed a method that simply concatenates the normalized two-dimensional (2D) position of local features to the feature value and then applies feature coding methods to obtain accuracy comparable to spatial pyramid matching with a smaller global feature dimension. Boureau *et al*. [2] apply pooling based on both image space and local feature space. Krapac *et al*. [16] derived a global feature that models both the local descriptor space and image space using the Gaussian mixture model. Similarly, Cinbis *et al*. [4] assumed a hierarchical probabilistic model that includes the feature position and uses the differential of log-probability with respect to hyper-parameters as the global feature.

Some researchers have considered pooling methods that use the weighted average. In [6], a weight based on saliency is proposed. Generalized Max Pooling [21] calculates the weight using the feature value to suppress the effect of frequent but meaningless local features. Some works [1, 20] adopted Gaussian Weighted average instead of original average pooling. We can regard our method as some extensions of these works because the proposed methods can derive similar weight as the pooling weight that corresponds to 0-th degree polynomial, and also derive the weight with higher order information as higher degree polynomials. Geometric L_p norm Feature Pooling (GLFP) [8] also considers the weighted average with respect to the local feature position. However, while we can apply our method even when the image sizes differ because our method considers the normalized position of the local features, we cannot apply GLFP directly for this situation becuase GLFP considers the adjacent relation between local descriptors. Also, our method is faster than GLFP because GLFP requires the calculation of cubic order of the number of local descriptors to calculate the weight, while our methods requires linear order.

Though our method computes the weight in an unsupervised manner, we can calculate the discriminative weight by combining our method with the methods that learn the weight of spatial pyramid discriminatively [10, 27].

In this paper, we focus on an extension of spatial pyramid matching with average pooling because this method is general and can be easily combined with coding methods.

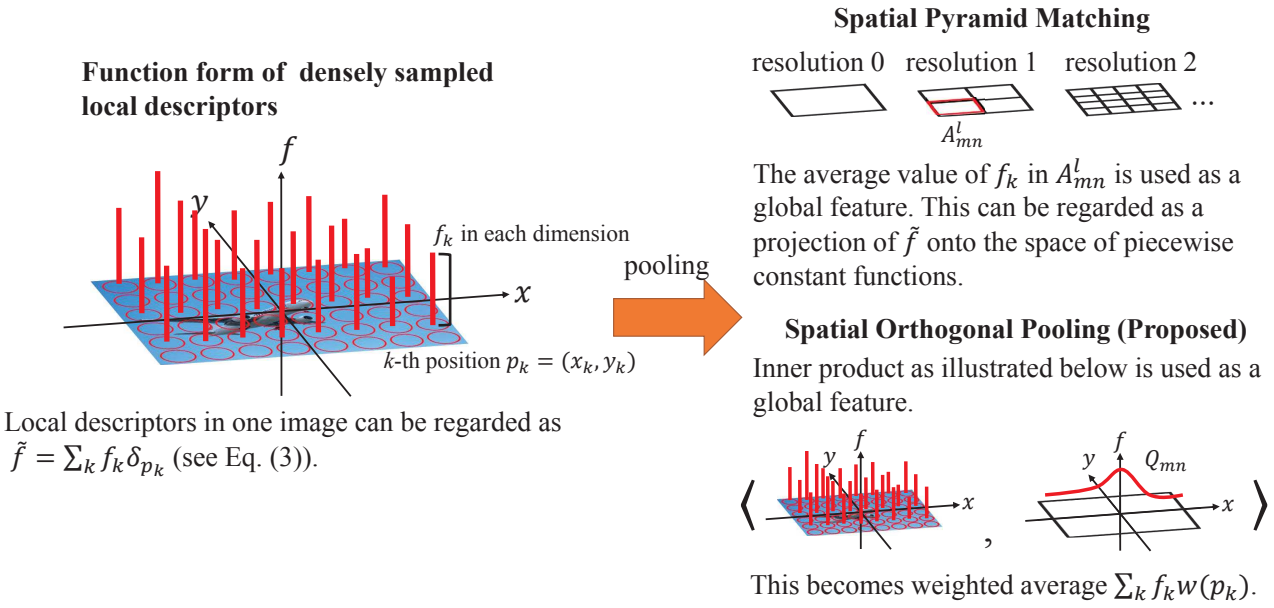


Figure 1. Overview of spatial pyramid matching and the proposed pooling method.

3. Spatial Pooling as a Projection

In Section 3 and 4, we propose an interpretation that regards local descriptors in one image as a function and pooling as a projection in the function space. Figure 1 shows an overview.

We assume that local features $\{(f_k, p_k)\}_{k=1}^N$ in one image are densely extracted, where N denotes the number of local features from an image, $f_k \in \mathbb{R}^d$ denotes the local features of each point after feature coding, such as the FV, and $p_k = (x_k, y_k) \in (-1, 1)^2$ is the normalized 2D position of each local feature with the image center $(0, 0)$. The goal of feature pooling is to construct one global feature $F \in \mathbb{R}^D$ from $\{(f_k, p_k)\}_{k=1}^N$. Since feature pooling is applied element-wise, we also assume that $d = 1$ for simplicity. In the general case, we concatenate the output of feature pooling for each dimension to obtain the global feature.

Average pooling is a method that simply ignores the feature position and uses the mean as the global feature as follows:

$$F = \frac{1}{N} \sum_{k=1}^{N} f_k. \tag{1}$$

Notice from this equation that average pooling completely disregards spatial information, which significantly affects recognition performance.

To include spatial information, spatial pyramid matching divides the image space using various resolutions and uses the feature mean in each subregion A^l_{mn} as the global feature as follows:

$$F^l_{mn} = \frac{1}{N^l_{mn}} \sum_{p_k \in A^l_{mn}} f_k, \tag{2}$$

where N^l_{mn} is the number of local features in A^l_{mn}. We select the image subregion A^l_{mn} such as $\left(\frac{m-1}{l}, \frac{m}{l}\right) \times \left(\frac{n-1}{l}, \frac{n}{l}\right)$, $(-l < m, n \leqq, l)$, where l corresponds to the resolution.

In the following, we propose the interpretation of feature pooling as a projection in the function space to analyze the property of spatial pyramid matching and the proposed spatial weighted pyramid uniformly using the property of the projected function space. Thus, we provide a function representation for both the input local features and the output of feature pooling.

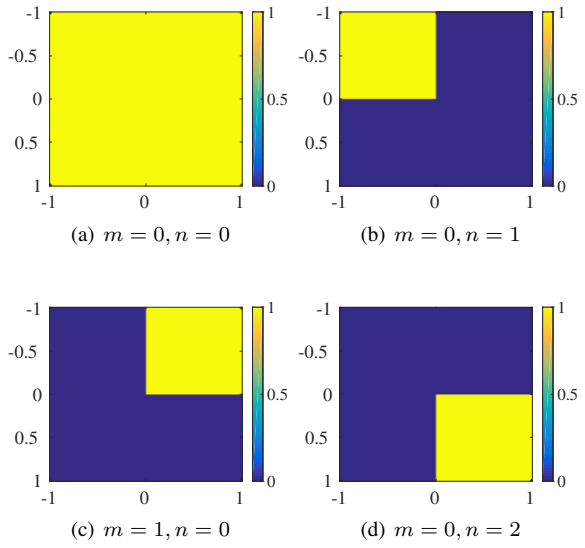
First, as a function representation of local features that includes both feature values and spatial information, we consider a hyper-function in the image space that connects the feature position to the feature value as follows:

$$\tilde{f} = \sum_{k=1}^{N} f_k \delta_{p_k}, \tag{3}$$

where δ_p denotes the delta function that satisfies

$$\langle \delta_p, g \rangle \equiv \int_{-1}^{1} \int_{-1}^{1} dx dy \delta_p(x, y) g(x, y) = g(p), \tag{4}$$

for a function g that is smooth and bounded near p.



(a) $m = 0, n = 0$ (b) $m = 0, n = 1$ (c) $m = 1, n = 0$ (d) $m = 0, n = 2$

Figure 2. Values of Weights for Spatial Pyramid Matching

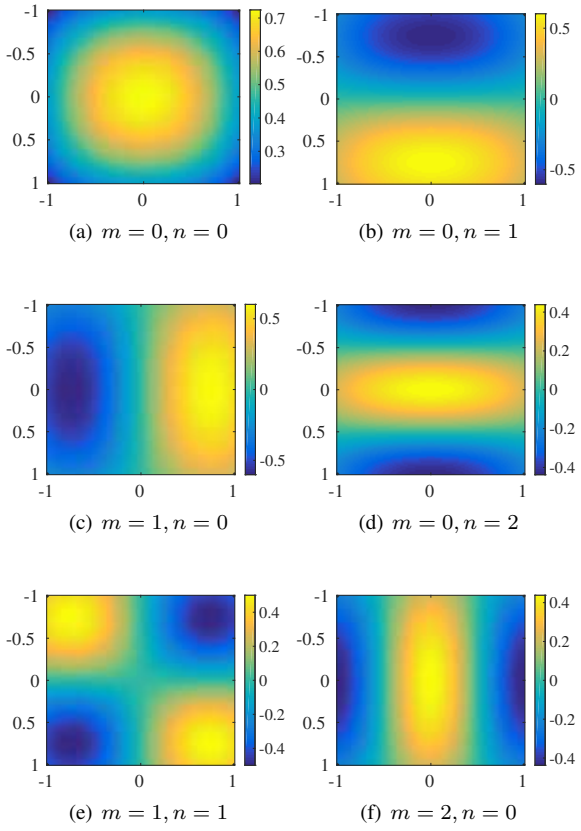
Next, we consider a function space that consists of functions that are constant in each A^l_{mn}: $\mathcal{F}^l_{\text{const}} = \{f | f = \sum_{m,n} c^l_{mn} 1_{A^l_{mn}}, c^l_{mn} \in \mathbb{R}\}$, where $1_{A^l_{mn}}$ is a function that outputs 1 in A^l_{mn} and 0 otherwise and c^l_{mn} is a coefficient. When l is fixed, the set $\{1_{A^l_{mn}}\}_{mn}$ is a base for this space; hence, the orthogonal projection is

$$\tilde{f} \to \sum_{m,n} \langle \tilde{f}, 1_{A^l_{mn}} \rangle 1_{A^l_{mn}}, \tag{5}$$

where each coefficient $\langle \tilde{f}, 1_{A^l_{mn}} \rangle = F^l_{mn} N^l_{mn}$. When we sample local features densely, we assume that N^l_{mn} is approximately equal for each m and n. This implies that the coefficients have almost equal information to F^l_{mn}. Thus, spatial pyramid matching is an orthogonal projection of the function representation of local features $\tilde{f}$ into a space of piecewise constant functions $\mathcal{F}_{\text{const}}$.

4. Spatial Orthogonal Pooling

In the previous section, we showed that spatial pyramid matching can be regarded as an orthogonal projection into a space of piecewise constant functions. The limitations of spatial pyramid matching that we stated in the introduction originate from the properties of the projected function space. Thus, we attempt to consider a different function space with better properties so that it generalizes average pooling, the basis is smooth and has rich information, and the orthogonal projection can easily be calculated. Now, we consider the space of o-degree polynomials $\mathcal{F}^o_{\text{poly}}$ and propose a novel pooling method that projects $\tilde{f}$ into $\mathcal{F}^o_{\text{poly}}$ using the bases of $\mathcal{F}^o_{\text{poly}}$. We call the proposed method spatial orthogonal pooling (SOP).



(a) $m = 0, n = 0$ (b) $m = 0, n = 1$ (c) $m = 1, n = 0$ (d) $m = 0, n = 2$ (e) $m = 1, n = 1$ (f) $m = 2, n = 0$

Figure 3. Values of $\langle \delta_p, Q^a_{mn} \rangle_a$ with small m and n for $\alpha = 0.25$.

4.1. Spatial Orthogonal Pooling Using the Standard Inner Product

First, we consider the standard inner product of $L_2(-1, 1)^2$,

$$\langle g, h \rangle = \int_{-1}^{1} \int_{-1}^{1} dx dy g(x, y) h(x, y). \tag{6}$$

The orthogonal polynomials for this inner product are the products of the orthogonal polynomials for the one-dimensional (1D) inner product

$$\int_{-1}^{1} dx g(x) h(x), \tag{7}$$

for each element x and y, which is the definition of Legendre polynomials. m-th Legendre polynomial P_m, which can be written as

$$P_m(x) = \sqrt{\frac{m+1}{2}} \frac{1}{2^m m!} \frac{d^m}{dx^m} [(x^2 - 1)^m], \tag{8}$$

is a m-degree polynomial and satisfies the following property:

$$\int_{-1}^{1} dx P_m(x) P_n(x) = \delta_{m,n}. \tag{9}$$

Thus, P_ms for $0 \leq m \leq o$ compose a basis of $\mathcal{F}^o_{\mathrm{poly}}$.

Thus, when we denote $Q_{mn}(x, y)$ as $P_m(x)P_n(y)$, then the proposed method concatenates the weighted average

$$F_{mn} = \langle \tilde{f}, Q_{mn} \rangle \quad (10)$$

$$= \sum_{k=1}^{N} f_k P_m(x_k) P_n(y_k), \quad (11)$$

for $0 \leq m, n$ and $m + n \leq o$ to obtain the global feature. The pyramid size is $\frac{(o+1)(o+2)}{2}$. Note that the computational cost of calculating these weights is negligible compared to the computational cost of calculating the feature value.

4.2. Spatial Orthogonal Pooling Using a Weighted Inner Product

Next, we consider the inner product

$$\langle g, h \rangle_a = \int_{(-1,1)^4} dp_1 dp_2 g(p_1) h(p_2) e^{-\frac{\|p_1 - p_2\|^2}{2a}}. \quad (12)$$

This inner product also summarizes the product of function values for a different position with the Gaussian weight $e^{-\frac{\|p_1 - p_2\|^2}{2a}}$ and is thus more robust to image translations than the standard inner product. We can balance the robustness and spatial information by adjusting a. When $a \to +0$, this method converges to average pooling. When $a \to \infty$, this method converges to the pooling method proposed in Section 4.1. Note that we do not use gaussian as a pooling weight directly. Instead, we calculate smooth weight function including both 0-th order information similar to gaussian and high frequency information so as to approximate the gaussian-weighted inner product.

Similar to the previous subsection, orthogonal polynomials for this inner product are products of orthogonal polynomials in the 1D case P^a_n. Let $Q^a_{mn}(x, y) = P^a_m(x)P^a_n(y)$. We concatenate the weighted average

$$F^a_{mn} = \langle \tilde{f}, Q^a_{mn} \rangle_a \quad (13)$$

$$= \sum_{k=1}^{N} f_k \langle \delta_{p_k}, Q^a_{mn} \rangle_a \quad (14)$$

for $0 \leq m, n$ and $m + n \leq o$ as a global feature. The pyramid size is $\frac{(o+1)(o+2)}{2}$. When the Gaussian weight $e^{-\frac{\|p_1 - p_2\|^2}{2a}}$ is used, we can calculate inner products $\langle x^{d_1}, x^{d_2} \rangle_a$ analytically using the error function by applying a variable transformation. Thus, P^a_n can be calculated using Gram-Schmidt orthonormalization. Furthermore, the inner products of orthogonal polynomials and delta functions

$$\langle \delta_p, Q^a_{mn} \rangle_a = \int_{(-1,1)^2} dp_1 Q^a_{mn}(p_1) e^{-\frac{\|p - p_1\|^2}{2a}} \quad (15)$$

can be written as functions of p and a. Thus, the complexity of calculating the weight is approximately the same as when the standard inner product is used and can be ignored. Figure 3 shows an example of the weights used in the experiment. This figure shows that the weights have similar information to those of original spatial pyramid matching. For example, Figure 3 (a) is a smoother version of the weight of layer 0 of spatial pyramid matching. Figure 3 (b), (c), and (d) construct the weights of layer 1. Since the weight is smooth, the proposed weights are both robust to local translation and have the sptial information comparable to spatial pyramid matching.

4.3. Analysis of the Robustness of the Proposed Methods

In this subsection, we analyze how the global feature changes when the positions of local features are slightly changed. We denote the position change as $\tau = (\tau_x, \tau_y)$ and assume that $\tilde{f}$ has a nonzero value only in $(-1 + \|\tau\|, 1 - \|\tau\|)^2$ for simplicity. In this case, the change of the global feature F^a_{mn}, δF^a_{mn}, can be bounded by

$$|\delta F^a_{mn}| = \left| \sum_{k=1}^{N} f_k \left(\langle \delta_p, Q_{mn} \rangle - \langle \delta_{p+\tau}, Q_{mn} \rangle \right) \right| \quad (16)$$

$$\leq \sum_{k=1}^{N} |f_k| \max_k |\langle \delta_{p_k}, Q_{mn} \rangle - \langle \delta_{p_k + \tau}, Q_{mn} \rangle|. \quad (17)$$

Thus, we evaluate the bound for

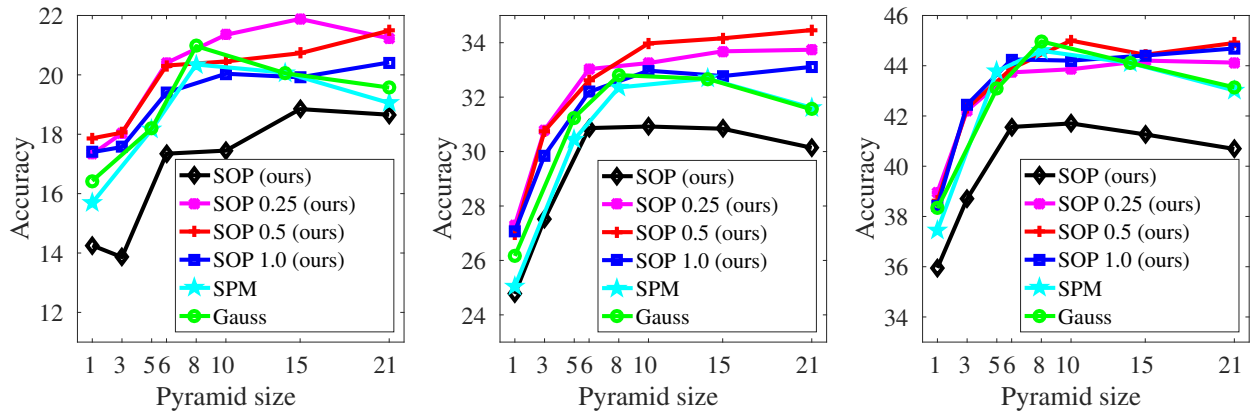
$$|\langle \delta_p, Q_{mn} \rangle - \langle \delta_{p+\tau}, Q_{mn} \rangle|. \quad (18)$$

When the standard inner product is used, by applying the mean value theorem, this value can be written as

$$|Q_{mn}(p) - Q_{mn}(p + \tau)| \quad (19)$$
$$= |\tau_x P'_m(x + \theta\tau_x) P_n(y + \theta\tau_y) + \tau_y P_m(x + \theta\tau_x) P'_n(y + \theta\tau_y)|,$$

for some $0 < \theta < 1$. Because P_m are polynomials, the $P'P$ terms on the right-hand side are bounded. Thus, for some constant c, the right-hand side is bounded by $c\|\tau\|$. Finally, the change of the global feature $|\delta F^a_{mn}|$ can be bounded using $\sum_{k=1}^{N} |f_k|, \tau$.

Similarly, by applying the mean value theorem to $e^{-\frac{\|p - p_1\|^2}{2a}}$ in Eq. (13), a bound can be derived for the case when the weighted inner product is used. This analysis is based on the smoothness of the basis function, so robustness is not necessarily ensured for spatial pyramid matching, which uses non-smooth piecewise constant functions as a basis.



(a) CUB-200 (b) Stanford Dogs (c) Caltech256

Figure 4. Comparison of classification performance using SIFT + FV in (a) CUB-200 dataset, (b) Stanford Dogs dataset, and (c) Caltech256 dataset.

5. Experiments

We tested our pooling methods on standard object recognition datasets and an action recognition dataset. In Section 5.1, we applied our methods on the image recognition datasets. In Section 5.2, we applied our methods on the action recognition dataset.

5.1. Image Recognition

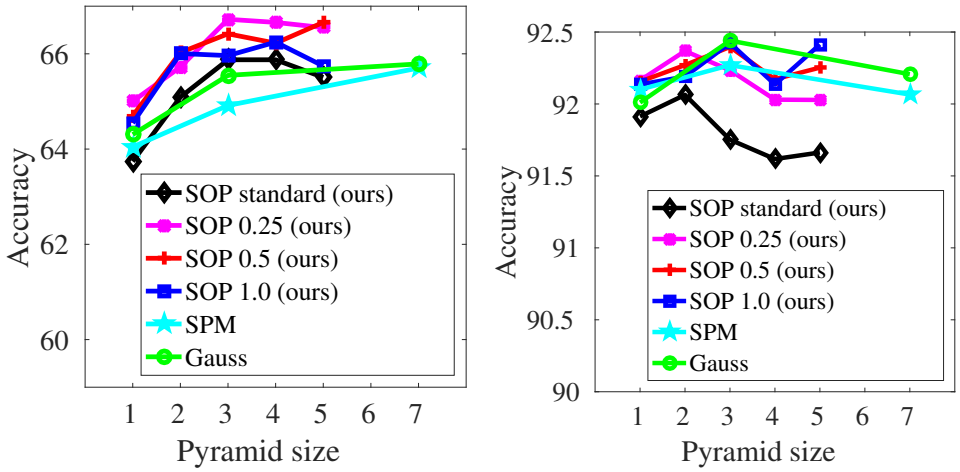
First, we evaluated our methods with SIFT + FV on the image recognition datasets. We tested our methods on three object recognition datasets, including fine-grained datasets (Caltech UCSD Birds 200 (CUB-200), Stanford Dogs, and Caltech256 dataset (Caltech256)).

CUB-200 [34] is a standard fine-grained object recognition dataset that consists of 200 bird species with 60 images per class. The Stanford Dogs dataset [13] consists of approximately 20,000 images of 120 dog classes. The Caltech256 dataset [9] consists of approximately 30,600 images of 256 object classes. We used given train/test split for the CUB-200 dataset and Stanford Dogs dataset and evaluated the accuracy. For the Caltech256 dataset, we randomly sampled 25 images per class as training data and 30 images per class as test data 10 times and evaluated the average of the accuracy.

For all datasets, we extracted 128-dimensional SIFT features densely with a step size of two and scales 4, 6, 8, and 10. We used 'vl_phow' implemented in VLFeat [30] for extraction. We used PCA to reduce the dimensionality of the features to 80. Then, each local descriptor was encoded using the FV with 128 clusters. We used 250,000 local features to learn the codebooks. For each dataset, we applied spatial pyramid matching with scales $[1 \times 1]$, $[1 \times 1, 2 \times 2]$, $[1 \times 1, 2 \times 2, 1 \times 3]$, $[1 \times 1, 2 \times 2, 3 \times 3]$, $[1 \times 1, 2 \times 2, 4 \times 4]$, which had pyramid sizes of 1, 5, 8, 14, and 21, respectively. We also compared the method that applied gaussian weight on the local feature according to the position in each pyramid as a baseline. We evaluated the proposed weighted average pooling using Legendre polynomials of degree 0, 1, 2, 3, 4, and 5 had pyramid sizes of 1, 3, 6, 10, 15, and 21, respectively and the proposed weighted average pooling using a weighted inner product with kernel parameter a = 0.25, 0.5, 1.0 and degree 0, 1, 2, 3, 4, 5 and had pyramid sizes that were the same as those using Legendre polynomials. We did not compare max pooling because this pooling does not work well on Fisher Vector. For post-processing, we applied power normalization plus L_2 normalization on each pyramid for spatial pyramid matching and power normalization plus L_2 normalization on the entire vector for the proposed methods.

For the linear classifier, we used a one-vs-rest SVM. We used the SVM implemented in LIBLINEAR [7] with $C = 100$ for training and plotted the accuracy.

Figure 4 shows the results for the original methods, where SOP indicates the results for Spatial Orthogonal Pooling using standard inner product, SOP with numbers indicate the results for Spatial Orthogonal Pooling using weighted inner product with the number meaning kernel parameter. The figures show that the performance is ranked as follows: *spatial orthogonal pooling with the standard inner product* $<$ *spatial pyramid matching* $<$ *gaussian-weighted spatial pyramid matching* $<$ *spatial orthogonal pooling with an appropriately weighted inner product*. The poor performance of the standard inner product may be because Legendre polynomials are not sufficiently robust to small translations. The appropriate choice of the kernel parameter contributes to the performance. In each dataset, the case for the kernel parameter $a = 0.25, 0.5$ demonstrates good performance. This is close to $1/3$, which is



(a) HMDB51 (b) UCF101

Figure 5. Comparison of classification performance using TDD + FV in (a) HMDB51 dataset and (b) UCF101 dataset.

the variance of the uniform distribution in $(-1, 1)$. In addition, the performance of spatial pyramid matching saturates around $[1 \times 1, 2 \times 2, 1 \times 3]$, but the performance of the proposed methods rapidly improves until the degree is two, and the performance gradually increases until the degree is five. Thus, the proposed methods demonstrate good performance, even when the pyramid size is small; moreover, better accuracy can be achieved using higher degree polynomials.

5.2. Action Recognition

Next, we applied our method to the action recognition task on two movie datasets, HMDB51 dataset [18] that consists of about 7,000 movie clips with 51 action categories, and UCF101 dataset [29] that consists of 13,320 movie clips with 101 action categories.

As a local descriptor, we extracted TDD features [33] that are the mean of output of convolutional layer around each improved dense trajectory [31]. As a CNN, we used VGG16 [28] network pretrained with spatial (RGB) and temporal (opticalflow) images and extracted the output of conv3, 4, and 5 layer as local features. Then we coded TDD feature using FV with dimension of the local descriptor reduced to 64 and the number of clusters 256 and then applied the linear SVM with $C = 100$.

In this case, since the feature dimension is larger than that used in the image recognition dataset and time-axis is finer compared to image-space with respect to the position of TDD features, we only considers spatal pyramid with respect to time-axis.

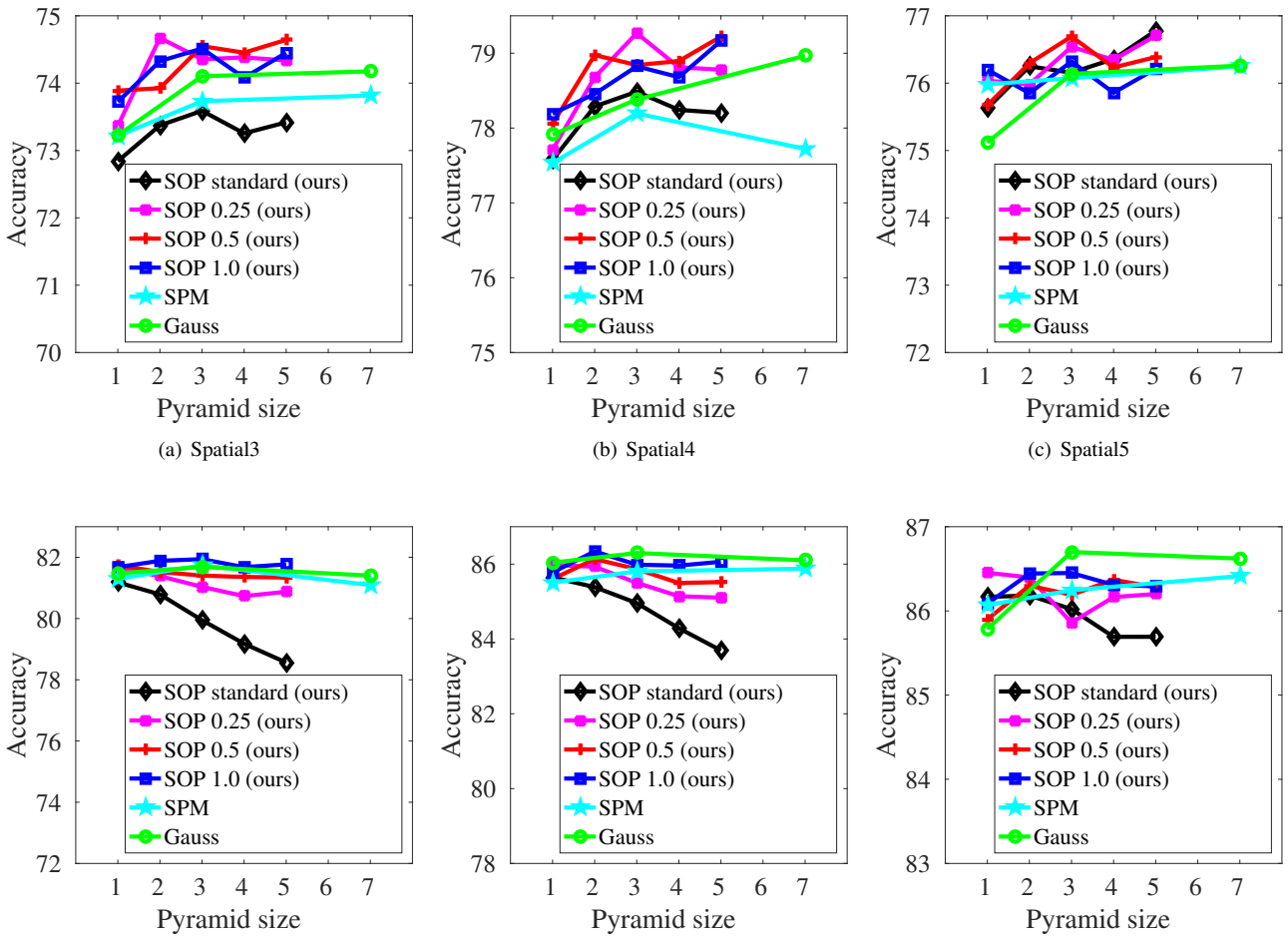
For each dataset, we compared spatial pyramid matching with scales $[1 \times 1 \times 1]$, $[1 \times 1 \times 1, 1 \times 1 \times 2]$, $[1 \times 1 \times 1; 1 \times 1 \times 2; 1 \times 1 \times 4]$, which had pyramid sizes of 1, 5, and 7, respectively and proposed methods with degree 0, 1, 2, 3, and 4 with pyramid sizes 1, 2, 3, 4 and 5 respectively. Each dataset gives 3 train/test splits and we evaluated the average accuracy using these 3 splits.

Figure 5 shows the results. The proposed methods show much better accuracy than spatial pyramid matching on HMDB51 dataset even when the pyramid size is small. Also, the proposed methods with weighted inner products show comparable performance on UCF101 dataset.

Next, we plotted the score of each layer in UCF101 dataset in Figure 6 to evaluate the effect of using the proposed temporal pooling in detail. We can gain performance by considering temporal information on RGB input, while temporal information does not work well on flow image. This is because while we can determine the time position of each RGB image exactly, flow image used the optical flow information of 10 frames around the time position. Since the length of each video clip is of the order of seconds, this time width of the frame is not negligible and time information is noisy in the case of flow images. In such situation, although the proposed method with standard inner product shows poor performance, the proposed method with weighted inner product is as robust as spatial pyramid matching and shows a slightly better score. These results showed that our methods can extract time-domain information better than spatial pyramid matching. Thus our methods are also effective for action recognition.

6. Conclusion

In this paper, we provided an interpretation of spatial pyramid matching as an orthogonal projection into the function space by considering local features in an image as a function on the image space. We also proposed a novel feature pooling methods called Spatial Orthogonal Pooling that



(a) Spatial3 (b) Spatial4 (c) Spatial5

(d) Temporal3 (e) Temporal4 (f) Temporal5

Figure 6. Comparison of classification performance of each layer using TDD + FV in UCF101 dataset. The name 'Spatial' indicates the score for the features extracted from RGB images and 'Temporal' indicates the score for the features extracted from flow images. The number in the name indicates the number of the layer.

used the weighted average as orthogonal projections into a space of low-degree polynomials and evaluated robustness to image translations of the proposed methods. Experimental evaluations using image recognition datasets and action recognition datasets demonstrated that the proposed pooling methods resulted in higher accuracy than spatial pyramid matching in both cases in which the pyramid size was small and large. These results showed that proposed methods exploit spatial information more effectively.

In the paper, We used the basic function space and inner products, but this can be modified to correspond to the problem, e.g., inner products that also consider inverting the x-axis to include mirror invariance can be used. Moreover, it is possible to generalize pooling using the L_p norm by considering the function space with a different metric.

Acknowledgements

This work was supported by CREST, JST and JSPS KAKENHI Grant Number JP17J05725.

References

[1] L. Bo, K. Lai, X. Ren, and D. Fox. Object recognition with hierarchical kernel descriptors. In *CVPR*, 2011.

[2] Y.-L. Boureau, N. Le Roux, F. Bach, J. Ponce, and Y. LeCun. Ask the locals: multi-way local pooling for image recognition. In *ICCV*, 2011.

[3] Y.-L. Boureau, J. Ponce, and Y. LeCun. A theoretical analysis of feature pooling in visual recognition. In *ICML*, 2010.

[4] R. G. Cinbis, J. Verbeek, and C. Schmid. Image categorization using fisher kernels of non-iid image models. In *CVPR*, 2012.

[5] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV Workshop on statistical learning in computer vision*, 2004.

[6] T. De Campos, G. Csurka, and F. Perronnin. Images as sets of locally weighted features. *CVIU*, 116(1):68–85, 2012.

[7] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874, 2008.

[8] J. Feng, B. Ni, Q. Tian, and S. Yan. Geometric lp-norm feature pooling for image classification. In *CVPR*, 2011.

[9] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. 2007.

[10] T. Harada, Y. Ushiku, Y. Yamashita, and Y. Kuniyoshi. Discriminative spatial pyramid. In *CVPR*, 2011.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*. 2014.

[12] H. Jégou, F. Perronnin, M. Douze, C. Schmid, et al. Aggregating local image descriptors into compact codes. *PAMI*, 34:1704–1716, 2012.

[13] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *CVPR workshop on FGVC*, 2011.

[14] P. Koniusz and K. Mikolajczyk. Spatial coordinate coding to reduce histogram representations, dominant angle and colour pyramid match. In *ICIP*, 2011.

[15] P. Koniusz, F. Yan, and K. Mikolajczyk. Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection. *CVIU*, 117(5):479–492, 2013.

[16] J. Krapac, J. Verbeek, and F. Jurie. Modeling spatial layout with fisher vectors for image categorization. In *ICCV*, 2011.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[18] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011.

[19] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[20] J. Mairal, P. Koniusz, Z. Harchaoui, and C. Schmid. Convolutional kernel networks. In *NIPS*, 2014.

[21] N. Murray and F. Perronnin. Generalized max pooling. In *CVPR*, 2014.

[22] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, 2010.

[23] M. Ranzato, Y. lan Boureau, and Y. L. Cun. Sparse feature learning for deep belief networks. In *NIPS*, 2007.

[24] J. Sánchez, F. Perronnin, and T. De Campos. Modeling the spatial layout of images beyond spatial pyramids. *Pattern Recognition Letters*, 33(16):2216–2223, 2012.

[25] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *IJCV*, 105:222–245, 2013.

[26] M. Shahiduzzaman, D. Zhang, and G. Lu. Improved spatial pyramid matching for image classification. In *ACCV*, 2010.

[27] G. Sharma and F. Jurie. Learning discriminative spatial representation for image classification. In *BMVC*, 2011.

[28] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2014.

[29] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. Technical report, CRCV-TR-12-01, 2012.

[30] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[31] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[32] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, 2010.

[33] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, 2015.

[34] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-ucsd birds 200. Technical Report CNS-TR-201, Caltech, 2010.