Regularized Max Pooling for Image Categorization

Minh Hoai[12]
http://www.robots.ox.ac.uk/~minhhoai/

[1] Visual Geometry Group
Department of Engineering Science
University of Oxford
Oxford, UK

[2] Department of Computer Science
Stony Brook University
Stony Brook, NY, USA

Abstract

We propose Regularized Max Pooling (RMP) for image classification. RMP classifies an image (or an image region) by extracting feature vectors at multiple subwindows at multiple locations and scales. Unlike Spatial Pyramid Matching where the subwindows are defined purely based on geometric correspondence, RMP accounts for the deformation of discriminative parts. The amount of deformation and the discriminative ability for multiple parts are jointly learned during training. RMP outperforms the state-of-the-art performance by a wide margin on the challenging PASCAL VOC2012 dataset for human action recognition on still images.

1 Introduction

One of the most fundamental and challenging tasks in computer vision is image categorization. Image categorization aims at recognizing the semantic category of an image, such as whether the image depicts a certain scene (e.g., street, office), contains a certain object (e.g., backpack, car), or captures a certain action (e.g., reading, riding bicycle).

A popular approach for image categorization is Spatial Pyramid Matching (SPM) [15]. SPM works by partitioning the image into increasingly fine sub-regions and aggregating local features found inside each sub-region (e.g., computing histograms [26] of quantized SIFT descriptors [17]). This approach has shown impressive levels of performance on various image categorization tasks. However, SPM relies on rigid geometric correspondence of grid division, which ignores the importance of semantic or discriminative localization. This model has limited discriminative power for recognizing semantic category with huge variance in location or large deformation.

To avoid the problem of rigid alignment, various works have proposed to model objects or scenes by parts in a deformable configuration [1, 8, 11, 20, 21, 24]. One particularly successful model is the Deformable Part Model (DPM) [8], which has yielded impressive results for object detection [6]. However, the flexibility of DPM comes with several restrictions. DPM requires iterative learning of latent discriminative parts. This learning procedure requires DPM to use visual features that can be swiftly extracted. Otherwise, the learning

procedure would be prohibitively expensive. That is perhaps why DPM is often associated with HOG [4]. In contrast, SPM can be used with any type of features. Furthermore, DPM is often used with a small number of parts at the same scale. While it is conceptually possible to increase the number of parts of a DPM, the practical benefit has not been established in practice. Ott and Everingham [19] showed that increasing the number of parts can even hurt the performance. Zhu and Ramanan [31] used DPM with a large number of parts for face detection, but their method requires fully supervised training data, i.e., the locations of parts are known during training. If the locations of parts are unknown, the performance significantly drops [32]. Also, parts might not be so essential [5].

In this paper, we propose Regularized Max Pooling (RMP). RMP combines the flexibility of a SPM and the deformability of a DPM. RMP is applicable to any type of features. It considers a large number of parts at different locations and scales. Parts are geometrically anchored, but can be discriminatively deformed. The learning of a RMP classifier is simple, without the need for expensive iterative updates.

We will demonstrate the benefits of RMP in recognizing human actions in still images. RMP outperforms DPM and SPM, especially for action classes with high level of deformation. Furthermore, the simplicity and flexibility of RMP allow it to be used with any type of features, including Convolutional Neural Network (CNN) features [14, 16]. Together with CNN features, RMP establishes the new state-of-the-art performance for human action recognition in still images, evaluated on the challenging dataset of PASCAL VOC2012 [7].

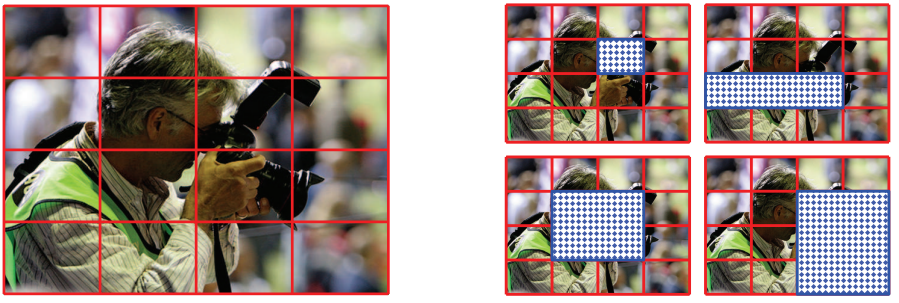
Related Work. Since the introduction of SPM [15], several improvements have been proposed. Harada *et al.* [10] propose a method to learn cell weights of SPM. Sharma and Jurie [25] learn spatial division beyond regular grid. Yan *et al.* [30] propose to pool features from densely sampled areas. Similarly, Jia *et al.* [12] learn adaptive receptive fields instead of manually defining special regions for feature pooling. These methods improve SPM, but they still rely on geometric correspondence between images.

There exist matching models that overcome the geometric rigidity of spatial pyramid. Kim *et al.* [13] propose a pyramid graph model that simultaneously regularizes match consistency at multiple spatial extents. Weinzaepfel *et al.* [29] propose DeepFlow, a multi-layer architecture for large displacement matching. These methods, however, are designed to recover dense correspondence between similar (stereo) images. They are not applicable to image categorization.

2 Regularized Max Pooling

An RMP model is a collection filters. Each filter is anchored to a specific image subwindow and associated with a set of deformation coefficients. The anchoring subwindows are predetermined at various locations and scales, while the filters and deformation coefficients are learnable parameters of the model. Fig. 1 shows a possible way to define subwindows. To classify a test image, RMP extracts feature vectors for all anchoring subwindows. The classification score of an image is the weighted sum of all filter responses. Each filter yields a set of filter responses, one for each level of deformation. The deformation coefficients are the weights for these filter responses.

In this section, we first review Least-Squares Support Vector Machines (LSSVM) [28], the corner stone for learning the filters. Subsequently, we will describe how the filters and deformation coefficients are learned.



(a) Grid division (b) Examples of considered subwindows

Figure 1: **From grid division to subwindows**. (a): An image is divided into 4×4 blocks. We consider rectangular subwindows that can be formed by a contiguous chunk of blocks. There are 100 such subwindows, and (b) shows four examples.

2.1 Review of Least-Squares SVM

LSSVM [28], also known as kernel Ridge regression [23], has been shown to perform equally well as SVM in many classification benchmarks [27]. LSSVM has a closed-form solution, which is a computational advantage over SVM. Furthermore, once the solution of LSSVM has been computed, the solution for a reduced training set obtained by removing any training data point can be found efficiently. This enables reusing training data for further calibration, as in cross-validation. This section reviews LSSVM and the leave-one-sample-out formula.

Given a set of n data points $\{\mathbf{x}_i | \mathbf{x}_i \in \Re^d\}_{i=1}^n$ and associated labels $\{y_i | y_i \in \{1,-1\}\}_{i=1}^n$, LSSVM optimizes the following:

$$\underset{\mathbf{w},b}{\text{minimize}}\ \lambda||\mathbf{w}||^2 + \sum_{i=1}^{n}(\mathbf{w}^T\mathbf{x}_i + b - y_i)^2. \tag{1}$$

For high dimensional data ($d \gg n$), it is more efficient to obtain the solution for $(\mathbf{w}, b)$ via the representer theorem, which states that $\mathbf{w}$ can be expressed as a linear combination of training data, i.e., $\mathbf{w} = \sum_{i=1}^n \alpha_i \mathbf{x}_i$. Let $\mathbf{K}$ be the kernel matrix, $k_{ij} = \mathbf{x}_i^T \mathbf{x}_j$. The optimal coefficients $\{\alpha_i\}$ and the bias term b can be found using closed-form formula: $[\alpha^T, b]^T = \mathbf{M}\mathbf{y}$. Where $\mathbf{M}$ and other auxiliary variables are defined as:

$$\mathbf{R} = \begin{bmatrix} \lambda\mathbf{K} & \mathbf{0}_n \\ \mathbf{0}_n^T & 0 \end{bmatrix}, \mathbf{Z} = \begin{bmatrix} \mathbf{K} \\ \mathbf{1}_n^T \end{bmatrix}, \mathbf{C} = \mathbf{R} + \mathbf{Z}\mathbf{Z}^T, \mathbf{M} = \mathbf{C}^{-1}\mathbf{Z}, \mathbf{H} = \mathbf{Z}^T\mathbf{M}. \tag{2}$$

If $\mathbf{x}_i$ is removed from the training data, the optimal coefficients can be computed:

$$\begin{bmatrix} \alpha_{(i)} \\ b_{(i)} \end{bmatrix} = \begin{bmatrix} \alpha \\ b \end{bmatrix} + \left(\frac{[\alpha^T\ b]\mathbf{z}_i - y_i}{1 - h_{ii}} \right) \mathbf{m}_i. \tag{3}$$

Here, $\mathbf{z}_i$ is the i^{th} column vector of $\mathbf{Z}$ and h_{ii} is the i^{th} element in the diagonal of $\mathbf{H}$. Note that $\mathbf{R}, \mathbf{Z}, \mathbf{C}, \mathbf{M}$, and $\mathbf{H}$ are independent of the label vector $\mathbf{y}$. Thus, training LSSVMs for multiple classes is efficient as these matrices need to be computed once. A more gentle derivation of the above formula is given in [2].

2.2 Learning formulation for RMP

Given a set of images $\{\mathbf{I}_i\}_{i=1}^n$ and associated labels $\{y_i | y_i \in \{1,-1\}\}_{i=1}^n$, consider a particular set of geometrically defined subwindows which can encode semantic content of an image at different locations and scales (e.g., Fig 1). Let $\{\mathbf{I}^j\}_{j=1}^m$ denote the set of subwindows for image $\mathbf{I}$. Let ϕ be the feature function of which the input is an image region and the output is a column vector. Let $\mathbf{D}^j$ be the feature matrix computed at location j for all images and $\mathbf{K}^j$ the corresponding kernel, i.e., $\mathbf{D}^j = [\phi(\mathbf{I}_1^j)\cdots\phi(\mathbf{I}_n^j)]$ and $\mathbf{K}^j = (\mathbf{D}^j)^T\mathbf{D}^j$. The joint kernel for all subwindows is the sum of all kernels: $\mathbf{K} = \sum_{j=1}^m \mathbf{K}^j$; this corresponds to concatenating all feature vectors computed at all subwindows. Given the kernel $\mathbf{K}$, we train an LSSVM (Sec. 2.1) and obtain a coefficient vector and bias term α, b. The filter for subwindow j can be computed as $\mathbf{w}^j = \mathbf{D}^j\alpha$.

For a particular subwindow j and an image $\mathbf{I}$, the regularized maximum score is defined:

$$f^j(\gamma) = \max_{k\in\{1,\cdots,m\}} \left\{(\mathbf{w}^j)^T\phi(\mathbf{I}^k) - \gamma.dist(\mathbf{I}^k,\mathbf{I}^j)\right\}. \tag{4}$$

Here γ is a non-negative regularization parameter and $dist(\cdot,\cdot)$ is the square geometric distance between two regions. The square geometric distance from a region R' to a reference region R is defined as:

$$dist(R',R) = \left(\frac{x'-x}{w}\right)^2 + \left(\frac{y'-y}{h}\right)^2 + \left(\log_2\left(\frac{w'}{w}\right)\right)^2 + \left(\log_2\left(\frac{h'}{h}\right)\right)^2, \tag{5}$$

where (x,y,w,h) and (x',y',w',h') are the center locations, the widths, and the heights of regions R and R' respectively. This distance function is asymmetric. It is invariant to the scale of the coordinate system. The last two terms of Eq. (5) measure the scale distance between R' and R. We use $\log_2(\cdot)$ to ensure that the scale distance from R' to R is the same for the following two cases: (i) R' is k times bigger than R; (ii) R' is k times smaller than R.

The value of $f^j(\gamma)$ is the regularized maximum response; it seeks a location with high filter response and low deformation cost w.r.t. to the anchor region $\mathbf{I}^j$. If γ is 0, $f^j(\gamma)$ is the maximum filter response. If γ is big, $\gamma.dist(\mathbf{I}^k,\mathbf{I}^j)$ will be big except for $k=j$ where $dist(\mathbf{I}^j,\mathbf{I}^j)=0$. Thus, for a big γ, $f^j(\gamma) = (\mathbf{w}^j)^T\phi(\mathbf{I}^j)$, which is the filter response of the anchor region.

The right setting for γ depends on the level of deformation of region j of the semantic class in consideration. Since the deformation level of a region is unknown, we start with an over-complete set of γ's and learn the tradeoff between deformation and discrimination. For each region j of an image $\mathbf{I}$, we construct a feature vector by varying the value of $\gamma \in \{\gamma_1,\cdots,\gamma_k\}$ and compute the regularized maximum response. Let $\mathbf{f}^j$ be the vector of obtained values, i.e., $\mathbf{f}^j = [f^j(\gamma_1),\cdots,f^j(\gamma_k)]^T$. For each image, we obtain a feature matrix by accumulating the filter responses for all regions $\mathbf{F} = [\mathbf{f}^1\cdots\mathbf{f}^m]$. Let $\mathbf{F}_i$ be the feature matrix for image $\mathbf{I}_i$. We jointly learn the deformation and discriminative ability of all regions by solving the following optimization problem:

$$\underset{\mathbf{S},\overline{b}}{\text{minimize}} \sum_{i=1}^n (trace(\mathbf{S}^T\mathbf{F}_i) + \overline{b} - y_i)^2 \tag{6}$$

$$\text{s.t. } s_{lj} \geq 0 \;\forall l = 1,\cdots,k,\; \forall j = 1,\cdots,m. \tag{7}$$

The above optimizes over a weight matrix $\mathbf{S} \in \Re^{k\times m}$ and a bias term $\overline{b}$. Each column of $\mathbf{S}$ is a weight vector for a particular region; it learns weights for the regularized maximum

responses for different values of γ's. The weights should be non-negative to emphasize the relative importance of higher filter responses. The objective of the above formulation minimizes the sum of L_2 losses. This is consistent with the loss terms of LSSVM (Eq. 1).

We start with an over-complete set of γ's and let the algorithm determines the right level of allowable deformation. In our experiments, we use $\gamma_1 = 0$, $\gamma_k = \infty$, $\gamma_l = 2^l/10^4$ for $l = 2, \cdots, k-1$, with $k = 15$. The feasible set of $\mathbf{S}$ is suitable for different levels of deformation, including the following two extreme cases:

1. Well-aligned semantic concept. For an image categorization task where the semantic concepts are well aligned, rigid geometric alignment is the right model. In this case, the weight matrix $\mathbf{S}$ could be all zeros except for the last row of all ones (the last row corresponds to $\gamma = \infty$).
2. Highly deformed semantic concept. For categorization tasks where the semantic concepts have high level of deformation, geometric correspondence should be ignored. In this case, the weight matrix $\mathbf{S}$ could be all zeros except for the first row of all ones (the first row corresponds to $\gamma = 0$).

There is a practical concern regarding the double use of training data. The filters are learned from non-deformed subwindows and are overfitted to them. Thus, no deformation will be incorrectly favored. In other words, the optimization of Eq (6) will return a solution that corresponds to the first aforementioned extreme case.

To avoid double use of training data, we compute $\{\mathbf{F}_i\}$ using the leave-one-out versions of α, b. Recall from Eq. (3) that the coefficient vector and bias term $\alpha_{(i)}, b_{(i)}$ of an LSSVM trained without the i^{th} training image can be computed efficiently. Let $\mathbf{F}_{(i)}$ denote the leave-one-sample-out version of $\mathbf{F}_i$, the learning formulation is adjusted as follows:

$$\underset{\mathbf{S},\overline{b}}{\text{minimize}} \sum_{i=1}^{n} (trace(\mathbf{S}^T \mathbf{F}_{(i)}) + b_{(i)} + \overline{b} - y_i)^2 \tag{8}$$

$$\text{s.t. } s_{lj} \geq 0 \; \forall l = 1, \cdots, k, \; \forall j = 1, \cdots, m. \tag{9}$$

The learning formulation in Eq. (8) is similar to Eq. (6), except for the inclusion of the bias adjustment terms $\{b_{(i)}\}$. This formulation corresponds to a linear program, which can be optimized efficiently using a linear programming solver such as Cplex[1].

It is important to note that the technique proposed here is applicable to any type of classifiers. If LSSVM is used, the leave-one-sample-out versions of $\{\mathbf{F}_i\}$ can be computed efficiently. If, for some reason, LSSVM cannot be used, the training data should be divided into two subsets, one for computing $\{\mathbf{F}_i\}$ and the other for learning $\mathbf{S}$ and $\overline{b}$.

3 Experiments

We perform a set of experiments on the *Action dataset* from the PASCAL VOC2012 Challenge [7]. This dataset contains action classes with different levels of deformation, making it ideal for analyzing the performance of various methods with respect to the amount of deformation. The performance measure is Average Precision (AP), which is the standard measurement used by PASCAL VOC Challenge.

[1] http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

Figure 2: **Random images from VOC2012 Action Dataset**. From left to right and top to bottom, the classes are: jumping, phoning, playinginstrument, reading, ridingbike, ridinghorse, running, takingphoto, usingcomputer, and walking.

3.1 VOC2012 Action dataset

The Action dataset is the dataset for the "Action Recognition from Still Images" challenge. This dataset contains 11 action classes: jumping, phoning, reading, playing instrument, riding bike, riding horse, running, taking photograph, using computer, walking, and others (images that do not belong to any of the first 10 classes). Some examples are shown in Fig. 2.

The Action dataset consists of three disjoint subsets for training, validation, and testing respectively. The annotation of the test subset is not available to us, and the performance on this subset can only be obtained by submitting the results to the PASCAL VOC evaluation server. Most results presented on this paper are computed on the validation data. We only run our method on the test data once, and the results returned by the PASCAL VOC evaluation server are used to compare with competition entries and a state-of-the-art method. The number of human subjects (*not* images) in the train, validation, and test subsets are are 3134, 3144, and 6283 respectively. Note the ROI (bounding box) is provided for each person.

3.2 Feature extraction

We use CNN features [16], which have been shown to yield good performance for ImageNet classification [14] and object detection [9]. We extract a 4096-dimensional feature vector for each subwindow using a Caffe implementation [3] of the CNN described by Krizhevsky et al. [14]. To compute the CNN feature vector for a subwindow, we first resize the image to 224×224 pixels (the desired input for our CNN network). This resized image is mean-subtracted and forward propagated through five convolutional layers and two fully connected layers. We refer readers to [3, 14] for more network architecture details.

3.3 Comparison with DPM and SPM

We first compare the performance of RMP with SPM and DPM. We consider two separate cases: unknown and known ROI (human bounding box). In the former case, the ROI is unknown and the task is to categorize the action of a human in an image without knowing where the person is (for both testing and training data). In the latter case, the ROI is known and the task is to categorize the ROI (instead of the image).

	SPM-SVM	SPM	MultiReg	RMP
jumping	65.4	64.2	67.6	**69.6**
phoning	36.3	37.9	38.1	**39.7**
play'instru	78.3	78.5	80.6	**83.8**
reading	43.9	42.1	45.8	**47.7**
ridingbike	78.7	79.4	80.9	**81.9**
ridinghorse	85.4	85.3	87.0	**88.1**
running	72.3	73.6	75.8	**78.6**
takingphoto	33.5	29.8	36.3	**45.0**
usingcomp	71.4	71.5	73.8	**75.7**
walking	34.5	34.6	36.1	**40.5**
mean	60.0	59.7	62.2	**65.1**

Table 1: **Whole image classification - unknown human bounding boxes**. This table shows the average precision values. SPM and SPM-SVM perform similarly. RMP outperforms the other methods on all action classes.

Table 1 reports the performance of several methods for the case where the human bounding box is unknown. All methods shown in the table use the same feature type. SPM is the method that uses 3-level spatial pyramid of CNN features. MultiReg is similar to SPM. It is also based on rigid geometric correspondence, but extracts features from a different set subwindows. In particular, MultiReg divides an image into a grid of 16 blocks (4×4) and considers all 100 rectangular subwindows that can be obtained by a contiguous set of blocks (Fig. 1). RMP uses the same set of subwindows as MultiReg, but the subwindows can deform (translation and scale). SPM, MultiReg, and RMP are all based on LSSVM. SPM-SVM is a variant of SPM; it uses SVM instead of LSSVM. As can be seen, RMP outperforms SPM, SPM-SVM, and MultiReg. Since all of these methods use the same feature type, the superiority of RMP can be accounted for by its ability to handle deformation. Notably, SPM with SVM and SPM with LSSVM perform equally well. In subsequent experiments, we choose LSSVM because of its computational advantage.

Tab. 2 shows the APs for recognizing the actions inside the provided human bounding boxes. SPM, MultiReg, and RMP are used as described above, except the inputs are ROIs instead of whole images. These methods use the linear kernel. We also experimented with RBF kernel, and the results are shown as SPM-RBF and MultiReg-RBF in Table 2. The RBF kernel is defined as: $k(\mathbf{x},\mathbf{y}) = \exp(-\frac{1}{\sigma}||\mathbf{x}-\mathbf{y}||^2)$. The kernel width σ is set to be the average value of $||\mathbf{x}-\mathbf{y}||^2$; which is computed over all pairs of training examples $(\mathbf{x},\mathbf{y})$. DPM is the method that is based on a the output of a DPM object detector [8]. For each action class, we train a DPM detector. To classify a test image, we first resize the image so that the larger dimension of bounding box of the person performing the action is 300 pixels. Centering the image on the bounding box, we cropped the image so that its width and height are 1.5 the width and height of the bounding box. We run the object detector on the cropped image and use the highest detection score for categorization.

As can be seen from Tab. 2, DPM performs poorly, except for ridingbike, ridinghorse, running, and walking. This is perhaps because these action classes are less deformed than the other classes. For example, a running or walking pose has lower variance than playinginstrument or takingphoto poses (see Fig. 3). This suggests the limited ability of DPM for

	DPM	SPM	SPM-RBF	MultiReg	MultiReg-RBF	RMP
jumping	41.6	75.1	75.1	77.6	76.5	**78.2**
phoning	25.6	39.0	39.3	40.9	40.6	**42.4**
play'instru	29.8	77.6	78.0	80.0	79.5	**82.2**
reading	24.7	49.7	52.0	53.0	**54.0**	53.2
ridingbike	69.0	88.2	88.4	89.9	89.2	**90.5**
ridinghorse	77.5	89.3	89.4	**91.2**	90.8	90.7
running	74.4	80.7	79.6	83.0	81.5	**84.1**
takingphoto	15.8	53.3	55.3	58.4	59.6	**63.1**
usingcomp	37.8	62.1	64.2	63.3	63.9	**64.8**
walking	52.3	60.0	61.4	64.1	64.5	**64.7**
mean	44.9	67.5	68.3	70.1	70.0	**71.4**

Table 2: **ROI (human bounding box) classification**. This shows the APs for various methods. The benefits of RMP are more significant for classes with high level of deformation such as takingphoto and playinginstrument.

Figure 3: **Example images for takingphoto (left) and walking (right)**. The ROIs (human bounding boxes) are drawn in black and yellow. The ROIs for takingphoto have higher level of deformation than for walking. This explains why RMP yields higher benefit for takingphoto than for walking.

modeling semantic classes with huge deformation (as also observed in [22]). Furthermore, DPM is outperformed by all other methods. This is perhaps because HOG is not as powerful as CNN features, as also shown in [9]. It is probably fairer to use DPM with CNN features. However, DPM requires iterative training and sliding window evaluation, so a direct combination of DPM and CNN is computationally prohibitive.

There are some other notable facts from Tab. 2. First, a non-linear kernel does not consistently improve the performance. The RBF kernel increases the performance of SPM but decreases the performance of MultiReg. Second, RMP outperforms the other methods, especially for classes with large deformation such as takingphoto and playinginstrument.

To understand the conditions under which RMP is expected to yield significant benefits, compare Table 1 and Table 2. In particular, consider the performance differences between RMP and MultiReg. RMP and MultiReg extract features from the same set of subwindows, but RMP allows deformation while MultiReg does not. In both tables, RMP outperforms MultiReg. However, the gap between RMP and MultiReg is bigger in Tab. 1 than in Tab. 2.

	Oxford Univ.	Stanford & MIT	Shenzhen Univ.	Hacettepe & Bilkent Univ.	Oquab et al. [18]	RMP (ours)
jumping	77.0	75.7	73.8	59.4	78.4	**82.3**
phoning	50.0	44.8	45.0	39.6	46.0	**52.9**
play'instru	65.3	66.6	62.8	56.5	75.6	**84.3**
reading	39.5	44.4	41.4	34.4	45.3	**53.6**
ridingbike	94.1	93.2	93.0	75.7	93.5	**95.6**
ridinghorse	95.9	94.2	93.4	80.2	95.0	**96.1**
running	87.7	87.6	87.8	74.3	86.5	**89.7**
takingphoto	42.7	38.4	35.0	27.6	49.3	**60.4**
usingcomp	68.6	70.6	64.7	55.2	66.7	**76.0**
walking	74.5	**75.6**	73.5	56.6	69.5	72.9
mean	69.5	69.1	67.0	56.0	70.2	**76.4**

Table 3: **Comparison with state-of-the-art methods.**. The first four methods are entries from the VOC2012 Challenge. Oquab *et al.* [18] is a method that also uses deep learning. Best results are printed in bold. Our method is the new state-of-the-art for 9 out of 10 classes, and it performs best overall.

Recall that Tab. 1 corresponds to image categorization while Tab. 2 corresponds to ROI categorization. The semantic content of whole images have higher degree of deformation than the content delineated by human bounding boxes. Thus the relative benefits of RMP are bigger for higher level of deformation.

3.4 Comparison with the state-of-the-art

We compare the performance of RMP with the entries of PASCAL VOC2012 Challenge and the recent work of Oquab *et al.* [18]. The results on the test set are obtained by submitting the output of RMP to the PASCAL evaluation server. This submission is done once; it conforms to the rules of the PASCAL VOC Challenge. For this VOC challenge, the human bounding boxes are provided, so we also use them in training and testing. For each action class, we train an RMP classifier for the human bounding boxes and another RMP classifier for the whole images, using images in both training and validation data. The latter classifier provides useful contextual cue for recognizing human action. Given an ROI and its containing image, let `roi_score` and `image_score` be the classifier scores for the ROI and the image, respectively. We adopt a simple combination scheme to compute the action score: `action_score` = 2×`roi_score` + `image_score`. Tab. 3 shows the results of RMP and the state-of-the-art methods. Our method achieves the best performance overall, and it exceeds the state-of-the-art for 9 out of 10 classes.

4 Conclusions

We have proposed RMP for image categorization. RMP combines the deformability of DPM and the flexibility of SPM. It classifies an image by extracting features at multiple deformable subwindows. An RMP classifier can be learned efficiently, without the need for expensive iterative update. The benefits of RMP have been demonstrated over DPM and SPM, especially

for classes with high level of deformation. The simplicity and flexibility of RMP allow it to be used with any type of features. Together with CNN features, RMP produces results that exceed state-of-the-art performance on the challenging task of recognizing human actions in still images.

Acknowledgements

This work was developed while the author was at the Visual Geometry Group, Department of Engineering Science, University of Oxford. It was supported by EPSRC grant EP/I012001/1. The author would like to thank Professor Andrew Zisserman for feedback, Karen Simonyan and Karel Lenc for CNN binaries, Relja Arandjelovic and Omkar Parkhi for proofreading the manuscript.

References

[1] Y. Amit and A. Trouve. POP: Patchwork of parts models for object recognition. *International Journal of Computer Vision*, 75(2):267–282, 2007.

[2] G. C. Cawley and N. L. Talbot. Fast exact leave-one-out cross-validation of sparse least-squares support vector machines. *Neural Networks*, 17:1467–1475, 2004.

[3] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *Proceedings of the British Machine Vision Conference*, 2014.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[5] S. K. Divvala, A. A. Efros, and M. Hebert. How important are 'deformable parts' in the deformable parts model? In *Parts and Attributes Workshop, ECCV*, 2012.

[6] M. Everingham, L. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2009.

[7] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. www.pascal-network.org/challenges/VOC/voc2012/workshop/, 2012.

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[10] T. Harada, Y. Ushiku, Y. Yamashita, and Y. Kuniyoshi. Discriminative spatial pyramid. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[11] M. Hoai, L. Ladicky, and A. Zisserman. Action recognition from weak alignment of body parts. In *Proceedings of the British Machine Vision Conference*, 2014.

[12] Y. Jia, C. Huang, and T. Darrell. Beyond spatial pyramids: Receptive field learning for pooled image features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[13] J. Kim, C. Liu, F. Sha, and K. Grauman. Deformable spatial pyramid matching for fast dense correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[14] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[15] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[16] Y. LeCun, B. Boser, J. S. Denker, and D. Henderson. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[17] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[18] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[19] P. Ott and M. Everingham. Shared parts for deformable part-based models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[20] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *Proceedings of the International Conference on Computer Vision*, 2011.

[21] S. N. Parizi, J. Oberlin, and P. F. Felzenszwalb. Reconfigurable models for scene recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[22] O. Parkhi, A. Vedaldi, C. Jawahar, and A. Zisserman. The truth about cats and dogs. In *Proceedings of the International Conference on Computer Vision*, 2011.

[23] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *Proceedings of the International Conference on Machine Learning*, 1998.

[24] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *International Journal of Computer Vision*, 56(3):151–177, 2004.

[25] G. Sharma and F. Jurie. Learning discriminative spatial representation for image classification. In *Proceedings of the British Machine Vision Conference*, 2011.

[26] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, 2003.

[27] J. A. K. Suykens, T. V. Gestel, J. D. Brabanter, B. DeMoor, and J. Vandewalle. *Least Squares Support Vector Machines*. World Scientific, 2002.

[28] J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9(3):293–300, 1999.

[29] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: Large displacement optical flow with deep matching. In *Proceedings of the International Conference on Computer Vision*, 2013.

[30] S. Yan, X. Xu, D. Xu, S. Lin, and X. Li. Beyond spatial pyramids: A new feature extraction framework with dense spatial sampling for image classification. In *Proceedings of the European Conference on Computer Vision*, 2012.

[31] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[32] X. Zhu, C. Vondrick, D. Ramanan, and C. C. Fowlkes. Do we need more training data or better models for object detection? In *Proceedings of the British Machine Vision Conference*, 2012.