# MULTI-SCALE REGION CANDIDATE COMBINATION FOR ACTION RECOGNITION

*Zhichen Zhao, Huimin Ma, Xiaozhi Chen*

Department of Electronic Engineering, Tsinghua University, Beijing, 100084, China
{zhaozc14, chenxz12}@mails.tsinghua.edu.cn, mhmpub@tsinghua.edu.cn,

## ABSTRACT

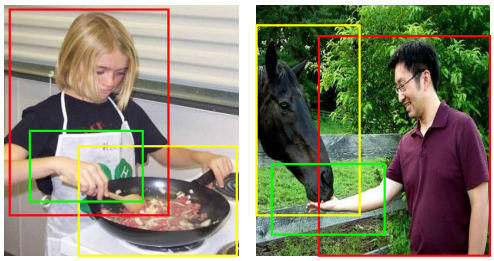
In still images, multi-scale regions contain rich information of different granularity. However, only semantically meaningful regions provide auxiliary cues for action recognition. Moreover, regions at different scales contribute differently. Motivated by the two observations, we propose an approach that is composed of three components: 1) detecting semantic region candidates at multiple scales, 2) training networks at each scale, 3) extracting features and learning to fuse them. The proposed approach captures multi-scale cues and highlights the optimal scale for each action, Experimental results show that our approach reaches the state-of-the-art performance on two challenging benchmarks: 1) PASCAL VOC 2012 and 2) Stanford-40.

***Index Terms***— action recognition, multi-scale candidate, feature fusion, neural network

## 1. INTRODUCTION

For the task of action categorization in still images, no motion information can be used, which makes it difficult to identify the action a human is performing in a single image. For this task, two main cues have been studied: interactive objects and discriminative pose parts. Many approaches model and detect them to obtain informative features. Khan et al. [1] use detectors to capture head and upper body regions, and concatenate feature fragments uniformly. Similarly, Gkioxari et al. [2] employ deep version of poselets [3] on head, torso, and legs to detect discriminative human parts. Yao and Fei-Fei [4] have also studied how a deformable part model [5, 6] performs for action recognition. Some other methods learn to model objects or human-object interactions, Gkioxari et al. [7] employ generic object proposals (Uijlings et al. [8]) to find proper interactive objects as cues.

Different with most of the related work that utilizes cues at the same scale, in this paper, we make two important observations that 1) for one specific action, there may be many semantic regions of different scales, all containing reasonable cues. 2) such cues contribute differently. Fig.1 illustrates our idea: in both images various cues can be found. For the action "cooking" (Fig.1a), the upper body, the pot and the hand holding a spatula all provide auxiliary cues, they are semantic regions at the scales of 1/2, 1/4 and 1/8, respectively (in this paper, the scale of a certain patch refers to the ratio between its area and the area of bounding box). However, they contribute differently. For "feeding a horse" (Fig.1b), the horse's head is a semantic region, but it can not help this action to be distinguished from "riding a horse". Interaction between the human hand and the horse head (green window) is the most informative cue, which suggests that multi-scale semantic regions can be combined to highlight the most contributory one.



(a) (b)

**Fig. 1**. Two actions and their corresponding cues at multiple scales. Colorized windows show semantic regions at different scales (red: 1/2, yellow: 1/4 and green: 1/8).

Motivated by those two observations, we propose a method that detects multi-scale semantic regions and learns to fuse the corresponding features. Unlike some approaches that combine all patches at each scale [9, 10], we detect and choose a few of them, which are called as "candidates". In this paper, candidates refer to regions that are probable to provide cues for recognition. Our method is composed by several steps: first, We train SVM models to detect semantic regions. To avoid detection missing, we recall more than one candidates. We found that searching and combining only a few more candidates works reasonably well at finding semantic regions (see Fig.2) and yielding more discriminative features (see Sec.4.2). We feedforward an image to obtain the convolutional feature map, and score all probable boxes by the SVM model. By apply non-maximum suppression (NMS) to reject regions that have high intersection-over-union (IoU) overlap with a higher scored region, more than one candidates containing various semantics can be generated. At each scale,

This work was supported by National Natural Science Foundation of China (No. 61171113).

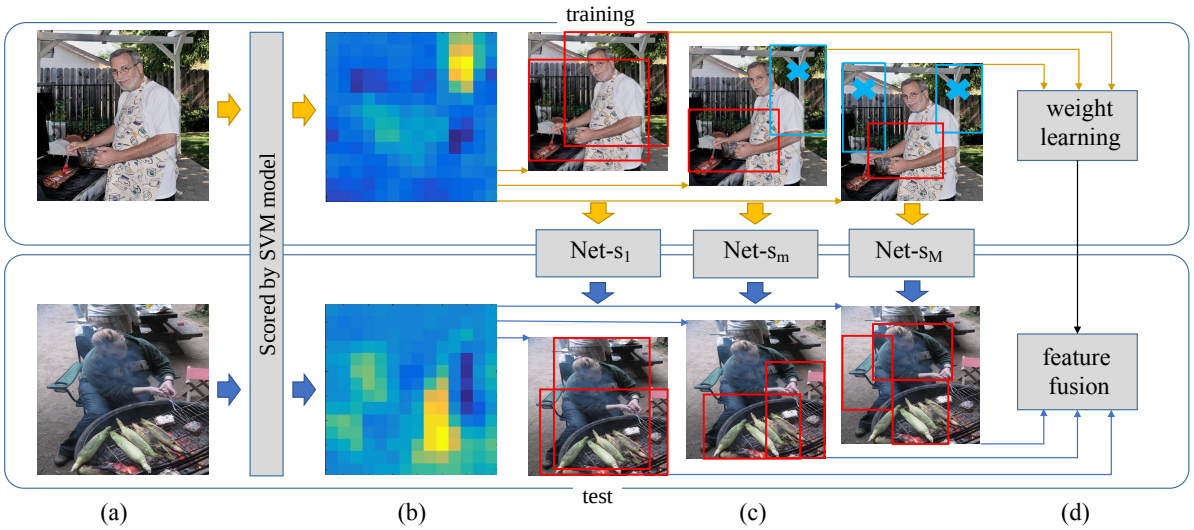



**Fig. 2**. Overview of our approach. The yellow arrows denote training stage and blue ones denote test. We take (a) input images, feedforward, and use SVM models to obtain (b) score maps. Given specific scales, multi-scale semantic region candidates can be generated in (c). In the training stage we remove some noise-prone candidates (denoted by blue windows and crosses) and train networks on filtered candidates. After feature extraction, a set of fusion coefficients are learned. In the test stage all candidates are used and weighted by the learned coefficients.

we train a network to distinguish candidates of different actions. Multiple candidates from the same image are treated as individual samples, and all of them are filtered to remove noise-prone ones by a human annotator. Lastly, we train a set of coefficients to weight feature fragments of different scales and fuse them, quantitative results show that weighted concatenation outperforms uniform concatenation.

We perform experiments on two challenging datasets: 1) PASCAL VOC 2012 and 2) Stanford-40 to evaluate our approach. We show that multi-scale semantic regions provide comprehensive cues for action recognition, and weighted concatenation highlights discriminative fragments. Our approach reaches the state-of-the-art on the two datasets.

## 2. SEMANTIC REGION CANDIDATE

Our framework is shown in Fig.2. The proposed approach is composed of semantic region candidate generation and feature fusion. The first stage detects semantic regions at multiple scales, and the second learns to provide more discriminative representations.

In this paper we employ Convolutional Neural Networks (CNNs) to extract features of images. In CNNs, data of a convolutional layer is a three-dimensional array of size $H \times W \times D$, where $H$ and $W$ denote height and width of the feature map, $D$ is the feature dimension. Since convolution, pooling and activation functions only operate on local regions, convolutional feature map $F$ is actually a set of features, each of which belongs to the spatially corresponding image region. Benefit from this, to obtain feature of a certain patch with size $(w', h')$, we can easily pool a $h' \times w' \times D$ block at the corresponding location, yielding a $D \times 1$ vector. Such technique has been widely used in object classification [9] and object detection [11, 12] to obtain features of local regions efficiently.

Similarly, to obtain feature of an image, the entire feature map can be pooled to an $D \times 1$ vector, and the vector can be directly sent into classifiers (e.g. SVM). In this way we train a one-vs-all SVM detector $\mathbf{w}$ for each action. The detection process is implemented as follow steps: 1) feedforward an image by the same network as above to a certain convolutional layer, yielding a feature map $\mathbf{F}$ of size $H \times W \times D$. 2) Given a specific scale $s$, choose one sub-block that yields maximum response to the detector:

$$\arg\max_{x,y,w,h} \mathbf{w}^T \left( \sum_{p=x}^{x+w} \sum_{q=y}^{y+h} \mathbf{F}(p,q) \right) \tag{1}$$

$$s.t. \quad wh = sWH, \tag{2}$$

where $x$ and $y$ denote locations, and each $\mathbf{F}(p,q)$ $(1 \le p \le W, 1 \le q \le H)$ is a $D \times 1$ vector. As mentioned above, $s$ denotes the ratio between area of the region and area of bounding box. Such optimization problem requires hundreds of dot product and pooling operations. Equivalently, Eq.1 can be reformulated as:

$$\arg\max_{x,y,w,h} \sum_{p=x}^{x+w} \sum_{q=y}^{y+h} \left( \mathbf{w}^T \mathbf{F}(p,q) \right) \tag{3}$$

which requires $WH$ dot product operations and hundreds of scalar sum operations. The term $\mathbf{w}^T\mathbf{F}(p,q)$ can be visualized as a score map (see Fig.2b).

The SVM model trained on the whole images contains various semantics and is sensitive to cluttered background.

That makes it fail to detect semantic regions sometimes. One kind of reliable technique is training SVM models on a fixed set of semantic parts [13, 14], but requiring full part annotations. In this paper, we found that even with models trained on whole images, semantic regions can be hit within a few more candidates. As demonstrated in Fig.2, the detector gives high score on top right corner of the upper sample and yields noise-prone candidates, However, we can recall only 1 or 2 more candidates to hit semantic regions (the hand holding a bowl). Detecting multiple candidates is implemented as follows: given a specific scale $s \in \{s_1, s_2, ..., s_M\}$, we score all probable boxes that satisfy $wh = sWH$ by calculating $\sum_{p=x'}^{x'+w} \sum_{q=y'}^{y'+h} \left(\mathbf{w}^T \mathbf{F}(p,q)\right)$. We apply non-maximum suppression (NMS) to select the top $K$ scored boxes as candidates, with fixed intersection-over-union (IoU) overlap. Another benefit of recalling more candidates is that using multiple candidates capture comprehensive semantics. As shown in Fig.2, red windows in the lower sample describe the whole person, her hands and the food, they all provide auxiliary cues for this action.

At the training stage, we remove some noise-prone candidates by a human annotator, and train new networks: Net-$s_1$, Net-$s_2$,..., Net-$s_M$. The postfix of network denotes that the network is trained on filtered candidates at this scale. Each new network still has the same architecture and input size with the former network. Since candidates are magnified gradually, subsequent networks need to be trained recursively (see Sec.4.1). At the test stage, for each candidate we extract features of the last fully connected layer by the corresponding network, which are all $4096 \times 1$ vectors. Thus we obtain feature fragments at all scales and candidates, denoted by $\{\{\mathbf{u}_{1,1}, \mathbf{u}_{2,2}, ..., \mathbf{u}_{1,K}\}, ..., \{\mathbf{u}_{m,1}, \mathbf{u}_{m,2}, ..., \mathbf{u}_{m,K}\}, ...\{\mathbf{u}_{M,1}, \mathbf{u}_{M,2}, ..., \mathbf{u}_{M,K}\}\}$, $\mathbf{u} \in \mathbb{R}^{4096\times 1}$.

## 3. MULTI-SCALE FEATURE FUSION

To obtain final representations, feature fragments should be fused. A fragment refers to a certain part of whole concatenated feature. Because it is hard to tell which candidates actually hit semantic regions, feature fragments of all candidates belonging to the same scale $s_m$ should be treated equally, we pool $\{\mathbf{u}_{m,k}\}_{k=1}^{K}$ to generate one feature fragment at each scale:

$$\mathbf{f}_m = \frac{1}{K} \sum_{k=1}^{K} \mathbf{u}_{m,k}, \quad m = 1, 2, ...M. \tag{4}$$

However, such pooling operations can only be applied on features extracted by the same network so $\mathbf{u}_m$ and $\mathbf{u}_{m+1}$ can not be fused by Eq.4. It is more reasonable to concatenate $\{\mathbf{f}_m\}_{m=1}^{M}$ than pooling them. Concatenating all fragments directly is a widely used method, however, Semantic regions at different scales contribute differently, and a set of coefficients is necessary to weight them.

Our goal is to learn a set of weights, or coefficients, $\{\lambda_m\}_{m=1}^{M}$, to concatenate features as $\mathbf{x} = [\lambda_1 \mathbf{f}_1^T, ..., \lambda_m \mathbf{f}_m^T, ..., \lambda_M \mathbf{f}_M^T]^T$, and minimize the loss function:

$$\min_{\lambda} \ \frac{\gamma}{2}||\lambda||^2 + C \sum_i \max(0, 1 - y^{(i)}(\mathbf{w}^T \mathbf{x}^{(i)} + b)) \tag{5}$$

$$s.t. \quad \lambda_m > 0, \quad m = 1, 2, ...M, \tag{6}$$

where $\mathbf{w}$ denotes SVM weights, $b$ denotes bias and $\gamma$ is a regularization coefficient. The $\mathbf{w}$ is trained on uniformly concatenated features, and fixed when we learn the coefficients $\{\lambda_m\}_{m=1}^{M}$. We utilize some zero-padding operations: $\widetilde{\mathbf{f}}_m = [0, 0, ...\mathbf{f}_m^T, ..., 0]^T$ $(m = 1, 2...M)$ where $\widetilde{\mathbf{f}}_m$ has the same length with $\mathbf{x}$ but only has non-zero elements at the $m$th fragments. Equivalently, $\mathbf{x}$ can be represented as $\mathbf{x} = \sum_m \lambda_m \widetilde{\mathbf{f}}_m$. Eq.5 can be reformulated as:

$$\min_{\lambda} \ \frac{\gamma}{2}||\lambda||^2 + C \sum_i \max(0, 1 - y^{(i)}(\mathbf{w}^T \sum_m \lambda_m \widetilde{\mathbf{f}}_m^{(i)} + b)). \tag{7}$$

By exchanging the term $\mathbf{w}^T \sum_m \lambda_m \widetilde{\mathbf{f}}_m$ into $\sum_m \lambda_m (\mathbf{w}^T \widetilde{\mathbf{f}}_m)$, the problem is equal to:

$$\min_{\lambda} \ \frac{\gamma}{2}||\lambda||^2 + C \sum_i \max(0, 1 - y^{(i)}(\lambda^T \widetilde{\mathbf{x}}^{(i)} + b)) \tag{8}$$

$$s.t. \quad \lambda_m > 0, \quad m = 1, 2, ...M, \tag{9}$$

where $\widetilde{\mathbf{x}} = [\mathbf{w}^T \widetilde{\mathbf{f}}_1, ..\mathbf{w}^T \widetilde{\mathbf{f}}_m., ...\mathbf{w}^T \widetilde{\mathbf{f}}_M]^T$. The problem is transformed into a fixed-bias SVM problem, and it can be easily solved by SVM solver.

## 4. EXPERIMENTS

In this section, we describe our experimental setup, analyze the proposed approach, and compare it with the state-of-the-art.

### 4.1. Experimental setup

Our experiments employ the very deep network models [15] implemented with the Caffe [16] framework. We take feature map of the last convolutional layer and remove the last pooling layer to obtain more accurate region locations, thus $H = W = 14$ and $D = 512$. We preserve $K = 2$ candidates at each scale and the scales are chosen as $\{1, 1/2, 1/4, 1/8\}$ where 1 denotes the bounding box, IoU is fixed as 0.5, 0.3 and 0.1 for $1/2$, $1/4$ and $1/8$ scales, respectively. We fine-tune the very deep network for $s_0$, and subsequent networks. Note that before training Net-$s_m$, we initialize it by the trained Net-$s_{m-1}$, otherwise it is hard to be trained because of rapid change of patch scale. Learning rates are set to $10^{-5}$, $10^{-6}$ and $10^{-7}$.

To evaluate our method, we perform experiments on two challenging datasets: PASCAL VOC 2012 [17] and Stanford 40 [18]. The PASCAL VOC dataset consists of 10 different actions, In each category about 400∼500 images are used for training and validation. The Stanford-40 dataset consists of 40 actions, and 100 images are used for training in each category. Both of them contain abundant actions and have been used to measure various approaches for action recognition.

**Table 1**. Comparison on two datasets, our method reaches the state-of-the-art performance on PASCAL VOC 2012, and outperforms existing methods on Stanford-40 by 3.4%.

| dataset | PASCAL | Stanford-40 |
|---|---|---|
| Regularized Max Pooling [10] | 76.3 | - |
| Deep Part Detectors [2] | 82.6 | - |
| R*CNN [7] | 90.2 | - |
| EPM [19] | - | 72.3 |
| Very Deep Network [15] | 84.0 | 71.7 |
| Action-Specific Detectors [20] | 77.0 | 75.4 |
| Ours ($M = 2$, $K = 1$) | 88.1 | 75.1 |
| Ours ($M = 2$, $K = 2$) | 88.8 | 76.2 |
| Ours ($M = 4$, $K = 2$) | 89.5 | 77.2 |
| Ours ($M = 4$, $K = 2$, with fusion) | **90.2** | **78.8** |

### 4.2. A detailed analysis of our method

First, we analyze our approach under different conditions in Tab.1. We found that multi-scale candidates improve the performance clearly, even the network has been fine-tuned. Employing single candidate ($K = 1$) provide limit improvement, while multiple candidates actually improve the performance (+0.7% and +1.1%). We also found that mAP becomes saturated when we use candidates of 4 different scales, maybe because that in a fine scale, there are more noise-prone regions which make fragments less discriminative. The feature fusion method proposed in Sec.3 also contributes to the performance (+0.7% and +1.6%).

In Fig.3 we compare our approach with the baseline which directly extracts features by fine-tuned very deep networks on the Stanford-40 dataset. Our approach improves the performance on all categories. Since we detect and train networks at multiple scales, we obtain remarkable gains on "drinking" (+14.4%), "phoning" (+14.5%) and "smoking" (+15.0%) where fine-scale semantic regions are critical for recognition. It suggests that our approach captures multi-scale cues, especially for fine-scale semantic regions.

### 4.3. Comparison with published results

In Tab.1 we also compare our approach with the state-of-the-art. Hoai [10] models relationship of two windows and learns holistic representations. Gkioxari *et al*. [2] train networks and detectors on head, torso, and legs, achieving 82.6% mAP on PASCAL VOC 2012. Gkioxari *et al*. [7] also employ the very deep network and Fast-RCNN [11] to combine contextual cues for actions. Sharma *et al*. [19] learn a set of typical action parts and measure the detection response as features. Khan *et al*. [20] employ transfer learning to learn action-specific detectors, and obtain 75.5% mAP on the Stanford-40. Our approach reaches the state-of-the-art performance on PASCAL VOC 2012, and outperforms existing methods on Stanford-40 by 3.4%. The major improvement comes from comprehensive and multi-scale cues, which cover both interactive objects and pose parts.

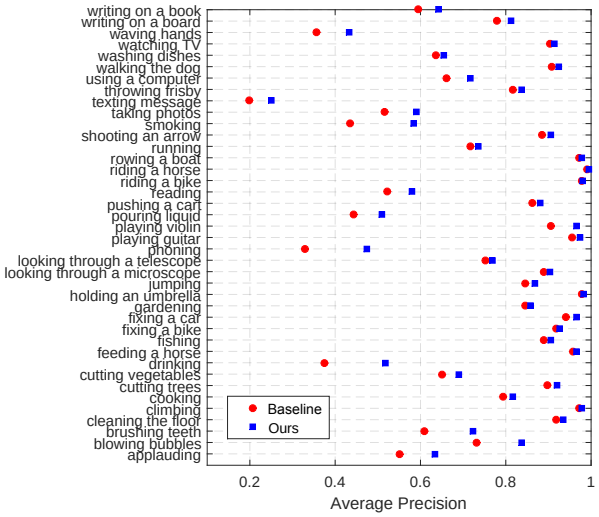



**Fig. 3**. Per-category performance (AP) of our approach and the baseline (fine-tuned very deep network) on the Stanford-40 dataset. Our method improves the performance by a large margin, especially on categories where interactive objects are small.

### 4.4. Comparison on candidate filtering methods

To measure the influence of candidate filtering when training networks, we compare the performance under different conditions: 1) train networks on raw candidates, 2) employ a fully automatic process, and 3) employ a human annotator to filter. For the fully automatic process, we employ features of conv4 layer, as suggested in [21]. We discard some outliers measured by $\Sigma^{-1}(f - \mu)$, where $f$ denotes feature, $\Sigma$ denotes cross-covariance matrix and $\mu$ denotes mean value. The performance under three conditions on the two datasets is 89.0%/89.0%/89.5% and 76.4%/76.6%/77.2%, respectively. With the change of scales, it is more difficult for a network to distinguish semantics. In practice the automatic process inevitably removes some semantically meaningful candidates, and provides only weak improvement. However, it needs only seconds for a human annotator to decide, who can remove noise-prone candidates correctly with little labour effort.

## 5. CONCLUSION

This paper aims at combining multi-scale cues to help recognize actions. We learn SVM models to detect multiple candidates, and train networks on them. The proposed fusion method also helps to generate more discriminative features. Our method provides the best results on both PASCAL VOC 2012 and Stanford-40 datasets. Experimental results demonstrate that in actions there are multiple auxiliary cues, existing at multiple scales and in various aspects. By combing these cues, one action is easier to be distinguished from another.

## 6. REFERENCES


[1] Fahad Shahbaz Khan, Joost van de Weijer, Roa Muhammad Anwer, Michael Felsberg, and Carlo Gatta, "Semantic pyramids for gender and action recognition," *TIP*, vol. 23, no. 8, pp. 3633–3645, 2014.

[2] Georgia Gkioxari, Ross Girshick, and Jitendra Malik, "Actions and attributes from wholes and parts," in *ICCV*, 2015.

[3] Lubomir Bourdev and Jitendra Malik, "Poselets: Body part detectors trained using 3d human pose annotations," in *ICCV*, 2009.

[4] Bangpeng Yao and Li Fei-Fei, "Grouplet: A structured image representation for recognizing human and object interactions," in *CVPR*, 2010.

[5] Pedro Felzenszwalb, David McAllester, and Deva Ramanan, "A discriminatively trained, multiscale, deformable part model," in *CVPR*, 2008.

[6] Santosh K. Divvala, Alexei A. Efros, and Martial Hebert, "How important are deformable parts in the deformable parts model?," in *ECCV*, 2012.

[7] Georgia Gkioxari, Ross Girshick, and Jitendra Malik., "Contextual action recognition with r*cnn," in *ICCV*, 2015.

[8] Jasper R. R. Uijlings, Koen E. A. van de Sande, Theo Gevers, and Arnold W. M. Smeulders, "Selective search for object recognition," in *IJCV*, 2013.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *ECCV*, 2014.

[10] Minh Hoai, "Regularized max pooling for image categorization," in *BMVC*, 2014.

[11] Ross Girshick, "Fast r-cnn," in *ICCV*, 2015.

[12] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[13] Ning Zhang, Je Donahue, Ross Girshick, and Trevor Darrell, "Part-based r-cnns for fine-grained category detection," in *ECCV*, 2014.

[14] Xiaopeng Zhang, Hongkai Xiong, Wengang Zhou, and Qi Tian, "Fused one-vs-all features with semantic alignments for fine-grained visual categorization," *TPAMI*, vol. 25, no. 2, pp. 878–892, 2016.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[16] Yangqing Jia, Evan Shelhamer, Je Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *arXiv: 1408.5093*, 2014.

[17] Mark Everingham, Luc Van Gool, Christopher Williams, John Winn, and Andrew Zisserman, "The pascal visual object classes challenge 2012 (voc2012) results," 2012.

[18] Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas Guibas, and Li Fei-Fei, "Human action recognition by learning bases of action attributes and parts," in *ICCV*, 2011.

[19] Gaurav Sharma, Fr ed eric Jurie, and Cordelia Schmid, "Expanded parts model for semantic description of humans in still images," in *arXiv:1509.04186*, 2015.

[20] Fahad Shahbaz Khan, Jiaolong Xu, Joost van de Weijer, Andrew D. Bagdanov, Rao Muhammad Anwer, and Antonio M. Lopez, "Recognizing actions through action-specific person detection," *TIP*, vol. 24, no. 11, pp. 4422–4432, 2015.

[21] Jonathan Krause, Hailin Jin, Jianchao Yang, and Li Fei-Fei, "Fine-grained recognition without part annotations," in *CVPR*, 2015.