Scalable Object Detection by Filter Compression with Regularized Sparse Coding

Ting-Hsuan Chao, Yen-Liang Lin, Yin-Hsi Kuo, and Winston H. Hsu
National Taiwan University, Taipei, Taiwan

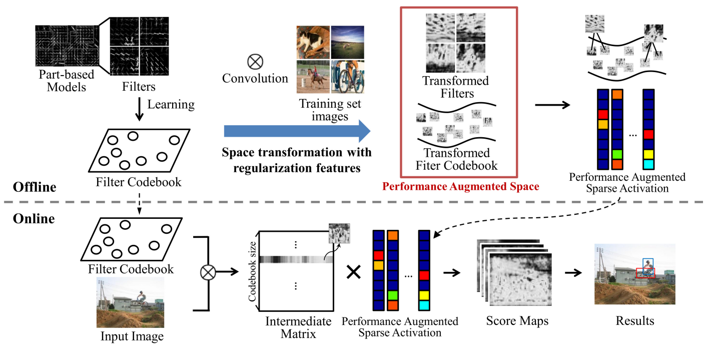

Figure 1: System architecture.

For practical applications, an object detection system requires huge number of classes to meet real world needs. Many successful object detection systems use part-based model which trains several filters (classifiers) for each class to perform multiclass object detection. However, these methods have linear computational complexity in regard to the number of classes and may lead to huge computing time. To solve the problem, some works learn a codebook for the filters and conduct operations only on the codebook to make computational complexity sublinear in regard to the number of classes.

In sparse coding method [2] proposed, we have filters $\mathbf{X} = \{X_1, ..., X_N\}$ which are collected from a set of part-based models trained on a chosen dataset. Next, we use these filters to learn a codebook $\mathbf{D} = \{D_1, ..., D_K\}$. Codewords of the codebook D can be thought as a set of basic elements consisting of edges and shapes. Redundant information of filters can be eliminated, reserving only informative elements to reconstruct filters. The optimization method of sparse coding can be formulated as follows.

$$\begin{aligned} \min_{\alpha_{ij}, D_j} \sum_{i=1}^{N} \quad & \left\| X_i - \textstyle\sum_{j=1}^{K} \alpha_{ij} D_j \right\|_2^2 \\ \text{subject to} \quad & \|\alpha_i\|_0 \le \varepsilon, \forall i = 1, ..., N \\ & \|D_j\|_2 = 1, \forall j = 1, ..., K \end{aligned} \tag{1}$$

By leveraging the Orthogonal Matching Pursuit algorithm (OMP) [1], we can efficiently compute an approximate solution. The original convolution process in object detection can be transformed into the following form:

$$X_i \approx \sum_{j=1}^{K} \alpha_{ij} D_j = \alpha_{\mathbf{i}} \mathbf{D} \tag{2}$$

$$\begin{bmatrix} \Psi * X_1 \\ \vdots \\ \vdots \\ \Psi * X_N \end{bmatrix} \approx \begin{bmatrix} \alpha_{\mathbf{1}} \\ \vdots \\ \vdots \\ \alpha_{\mathbf{N}} \end{bmatrix} \begin{bmatrix} \Psi * D_1 \\ \vdots \\ \Psi * D_K \end{bmatrix} = AM \tag{3}$$

In Equation 3 we denote the feature pyramid of an image as Ψ, $*$ the convolution operation, α_i the i^{th} filter's sparse activation, D_i the codeword of the learned codebook and X_i the filter. We can get the brief representation AM as the last term in Equation 3. A is a matrix of sparse activation and M is a matrix of intermediate representation. We illustrate these representation in Figure 1. Notice that we can calculate the matrix of intermediate representation M instead of calculating the score of each class separately for multiclass object detection. In other words, we can amortize the computation cost of the intermediate matrix. The original DPM requires Nd operations to compute N filters where d corresponds to the dimension of the filters, while this method requires only $Kd + N\|\alpha\|_0$. We can yield speedup

$$speedup = \frac{Nd}{Kd + N\|\alpha\|_0} \tag{4}$$

In Equation 4, N and d are fixed for a given dataset such that we should reduce K and $\|\alpha\|_0$ to yield more speedup.

Filters in object detection system are the weights trained by SVM and we have known that image and SVM weight are very different in statistical characteristics. The sparse coding method simply takes filters as images, learns a codebook and use codewords of the codebook to reconstruct filters by minimizing L2 distance between original and reconstructed filters; that is, it minimizes difference of filter appearance. However, we should directly reconstruct filter functionality instead of filter appearance. Our method, Regularized Sparse Coding, proposes to solve this problem by introducing regularization features from the training set of a given dataset to regularize the whole process. It makes functionality receive higher weights than appearance in reconstruction. In detail, we can divide our method into two stages, in the first stage we train a general codebook for filters collected from part-based models. In the second stage we use regularization features to transform filters and the codebook into performance augmented space. Then we reconstruct the transformed filters with the transformed codebook in performance augmented space as shown in Figure 1. As mention before, the physical meaning of performance augmented space will enforce filters to reconstruct functionality in it. The whole process of Regularized Sparse Coding can be formulated as an optimization problem:

$$D = \min_{D \in C} \sum_{i=1}^{N} \min_{\alpha_i} \left(\frac{1}{2} \left\| X_i - \textstyle\sum_{j=1}^{K} \alpha_{ij} D_j \right\|_2^2 + \lambda_1 \|\alpha_i\|_0\right) \tag{5}$$

$$\begin{aligned} \min_{\alpha_i} \left(\frac{1}{2} \left\| F_{reg} \cdot (X_i - \textstyle\sum_{j=1}^{K} \alpha_{ij} D_j) \right\|_2^2 + \lambda_2 \|\alpha_i\|_0\right) \\ \forall i = 1, ..., N \end{aligned} \tag{6}$$

Equation 5 suggests that the codebook is trained in the original sparse coding way. Equation 6 describes how we involve regularization features in filter reconstruction and generate performance augmented sparse activations. F_{reg} is a matrix of regularization features which is the key to our method. To construct this matrix, we collected lots of images from the training set of the dataset. Next, we extract feature pyramids from the collected images and randomly pick some feature patches which have the same size of filters. Finally, we use the feature patches as the rows of F_{reg} and use this matrix to regularize sparse coding. Multiply F_{reg} into clause of Equation 6, $\|\cdot\|_2^2$ term in Equation 6 becomes

$$F_{reg} \cdot X_i - F_{reg} \cdot \left(\sum_{j=1}^{K} \alpha_{ij} D_j\right) = Score_i - Score_i{}'. \tag{7}$$

We can consider it as norm2 distance between the original and the reconstructed score, that is, the classification error between two filters. As we proposed earlier, we should reconstruct filter functionality, i.e., the ability of filter to produce accurate score map in object detection, instead of filter appearance.

[1] Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41 (12):3397–3415, 1993.

[2] Hyun Oh Song, Stefan Zickler, Tim Althoff, Ross Girshick, Mario Fritz, Christopher Geyer, Pedro Felzenszwalb, and Trevor Darrell. Sparselet models for efficient multiclass object detection. In *Computer Vision–ECCV 2012*, pages 802–815. Springer, 2012.

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.