Procrustean Decomposition for Orthogonal Cascade Detection

Kun Duan
GE Global Research
kun.duan@ge.com

Wei Wang
GE Global Research
wei.wang8@ge.com

Ting Yu
Nest Labs, Alphabet Inc.
yuti@google.com

Abstract

In this paper, our goal is to speed up a standard sliding window detector while maintaining detection accuracies. We do this by decomposing its weight template into multiple component detectors with no redundancy. Each component detector captures partial discriminative appearance, and they can be used in a cascade detection pipeline to aggressively reduce the size of search space. We formulate the decomposition problem as an orthogonal procrustes *problem. We further approximate the component detector in the first cascade layer as separable* 2*D filters (*i.e. *a product between two vector filters). The running time of such component detector is thus reduced from quadratic to linear with the dimension lengths of the detector template. We conduct extensive experiments using our approach on two well-known object detection datasets: INRIA pedestrian detection dataset and PASCAL VOC 2007 detection dataset. We used HOG features and CNN features in our experiments. Our approach is almost "free lunch": without manipulating the feature representations, it makes the detection process several times faster.*

1. Introduction

State-of-art object detectors are usually a classifier (*e.g.* a linear SVM) trained on multi-channel features (*e.g.* HOG histograms [6], SIFT bag-of-words [5] or CNN features [23]) extracted from training images. At test time, they are applied on the images through convolution operations or sliding window search [3, 4, 6, 7, 9, 11, 15]. For example, [3] uses a single non-linear rigid template detector with a feature pooling scheme, and applies sliding window search to achieve very strong performance. Deformable part models [15] require convolving images using root and part filters. The root and part filter scores are then aggregated together to output final detections. These methods show promise in terms of their detection performance, but evaluating the detector requires dense computations.

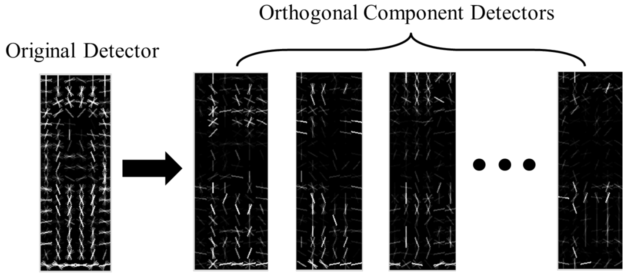


Figure 1: Decompose a person detector into $P = 10$ orthogonal detectors. Note that some component detectors are "part"-like, *e.g.* the first detector is a "shoulders-and-legs" detector, and the third detector is a "head" detector. Best viewed on screen.

Our objective here is to accelerate a pre-trained linear sliding window object detector without additional engineering work on the image features. Although linear models are quite standard, they have produced competitive performance in many computer vision tasks [27]. State-of-art deep learning detectors like Deep Pyramid DPM [16] trains linear SVM on top of deep image features. One classical way to accelerate detection process is to use a *cascade* of detectors [14, 30, 35, 39]. Traditional approaches [35] that learn a detection cascade usually train hundreds or thousands weak classifiers by adaptively reweighting training exemplars. These weak detectors are trained on redundant features, and the training procedure can take very long time.

We propose a novel detector decomposition framework to solve this problem. We start with a pre-trained linear object detector, and decompose it into a set of component detectors by minimizing the reconstruction error between each component detector and the original detector. In the mean time, we require these component detectors to be orthogonal to each other, and therefore they are guaranteed to capture no redundant feature information. We prove that the sum of detection scores of the component detectors are equal to the detection score of the original detector. This allows us to build a fast detection cascade pipeline by learning a series of admissible rejection thresholds. Figure 1 gives an example orthogonal decomposition of a person detector from our decomposition algorithm.

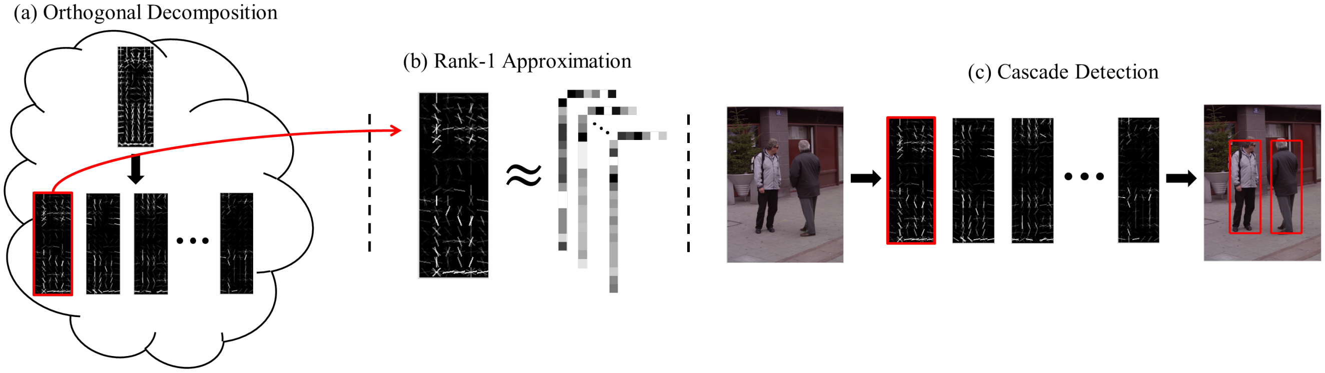


Figure 2: Overview of our proposed approach. (a) Decompose a pre-trained linear detector into orthogonal components. (b) Approximate the first cascade layer using the largest rank-1 components from each feature channel. (c) Cascade detection on test images.

In the next step, we apply rank-1 approximation to the first component detector in the detection cascade, and represent it as a product of a column filter and a row filter. Therefore, this component detector can be evaluated in time linearly rather than quadratically with the detector size. These component detectors capture partial (but orthogonal) discriminative appearances. Our decomposition framework is simple and general: using our approach, one can easily speed up any linear detector by a significant amount. We show an overview of our approach in Figure 2.

Orthogonal procrustes [31] has been widely used in computational geometry and 3D point cloud analysis [10, 20], but it is less well known to other fields of computer vision. We revisit this classical problem and apply it to object detection. Our work also has connections with a lot of works in object recognition, *e.g.* sparse coding [26, 36, 37], low-rank approximation [21], cascade object detection [35], *etc.* We will review representative works in Section 2.

Summary of contributions. In this paper, we propose a novel generic framework to decompose a linear object detector into orthogonal components while minimizing the reconstruction error. We formulate the problem as a generalized orthogonal procrustes problem. We give theoretical guarantees on building detection cascade with the component detectors generated by our algorithm, and we apply rank-1 approximations on the first detection cascade layer to obtain significant speed-up. Finally, we conduct systematic experiments on the INRIA and PASCAL VOC 2007 datasets using HOG and CNN features. Our approach makes a linear detector several times faster, and is almost "free lunch" to any linear object detection system.

2. Related Work

Our work is related with a number of works on object detector acceleration and we review representative ones here. One approach that solves this problem is based on reducing feature computation time. For example, integral channel features [8, 35] calculates simple statistics in each individual feature channel through the use of *summed area tables*, and thus reduces the feature computation time significantly. Other feature engineering approaches include dimension reduction (*e.g.* PCA or sparse coding) on complicated features [15, 28]. However, these methods introduce extra computational expense, and in practice the speed-up is not significant. On the contrary, we intend to speed up object detectors without any modification of the original image feature. Moreover, our approach can be combined with the above methods to obtain even faster detection speed.

Another line of work on speeding up object detectors is based on reducing the search space of object candidates. For example, branch-and-bound approaches by Lampert *et al.* [24, 25] prune out image regions with low confidence scores and does not exclusively evaluate all image windows. However, these methods assume *bag-of-words* image representations; our method is much more general, and can be used with any feature types. Cascade detectors [14, 35] reduces the search space by applying a set of weak detectors sequentially, while each detector only searches within high scoring image regions returned by its predecessor. The classical approach by Viola *et al.* [35] trains weak detectors by iteratively reweighting training exemplars based on errors of current detections. This approach usually requires hundreds or thousands of detectors to be trained, and the final cascade pipeline consists of more than dozens of cascade layers. Our motivation is different with the above approaches. We seek for reconstructions of a pre-trained object detector and enforce these reconstructions to have no redundancy with each other. We also prove that these reconstructions form a detection sequence, where the sum of component detector scores is equal to the original detector score. This gives us a convenient way to learn a series of rejection thresholds for detector accelerations.

Our approach is also relevant with work on separable filters [29] or separable detectors [2, 38]. Rigamonti *et al.* [29] propose to learn a small bank of separable filters to reconstruct a large bank of full-rank filters, but the applications are focused on low-level image processing rather than object detection. Bauckhage *et al.* [2] propose to jointly learn the separable filters and the classifiers, but the approach only works on simple and rigid objects. Pirsiavash *et al.* [27] is an improvement where a biconvex learning framework is used. However their work focus on transfer learning rather than speeding up object detection. Yan *et al.* [38] uses low-rank approximation of root filters to accelerate deformable part models. Their work does not handle the filter redundancy problems, and does not aim to reconstruct existing object detectors. On the other hand, we assume only one linear detector is given, and try to decompose it into multiple component detectors with *no* redundancy.

Other related works include object detection approximations using sparse representations. Song *et al.* [32, 33] propose to learn a set of basis templates which are then used to reconstruct detectors for multiple object categories. Their approach also minimizes the redundant information shared by different object detectors at the same time. Their method improves detection speed when multi-class object detectors are applied; but this advantage disappears if there is only one single-class object detector. The recent "Shufflets" representation proposed by Kokkinos *et al.* [22] adopts a similar idea. It removes redundancy by allowing small basis templates to be shiftable when reconstructing root and part filters of a deformable part model. Our approach is different, in that we seek for orthogonal reconstructions of one instead of multiple object detectors. Also, the detector speed-up come from the fact that these orthogonal detectors can be used in a detection cascade, and that we apply rank-1 approximation on the first cascade layer.

The work most closest to ours is Ambai *et al.* [1], where the authors proposed to decompose an object detector into a weighted sum of linear ternary classifiers trained on binarized HOG features. The major speed-up of their approach comes from efficient computations of ternary template matching with binary features. In our work, each component detector is treated as an individual object detector, and they are constrained to capture discriminative signals from orthogonal directions. More importantly, we avoid any feature engineering and do not have any assumption on feature types. Thus our method is a generic framework to speed up any linear object detector.

3. Model

Our key idea is to reconstruct an object detector using a set of orthogonal detectors such that each one captures partial discriminative appearances. At test time, they can be used in a cascade pipeline for fast object detection. In addition to, we also observe that most computations happened in the first cascade layer, thus we apply rank-1 approximation on the first cascade layer. In practice, these give us significant speed-up with almost no loss in detection accuracy.

3.1. The Decomposition Framework

Let $\vec{\mathbf{w_0}}$ be a column vector reshaped from a pre-trained detector template $\mathbf{T_{s\times t\times c}}$, where $s \times t$ is the detector's height and width dimensions and c is the detector's number of feature channels. Let $p \in \{1\dots P\}$, and $\{\vec{\mathbf{w_p}}\}$ be the reshaped column vector representations of the P component detectors $\left\{\mathbf{T^{(p)}_{s\times t\times c}}\right\}$, which are decomposed from the original detector $\mathbf{T_{s\times t\times c}}$. We seek for $\{\vec{\mathbf{w_p}}\}$, such that they minimize the reconstruction error between each component detector $\mathbf{T^{(p)}_{s\times t\times c}}$ and the original detector $\mathbf{T_{s\times t\times c}}$.

$$\min_{\{\vec{\mathbf{w_p}}\}} \sum_{p=1}^{P} \|\vec{\mathbf{w_0}} - \vec{\mathbf{w_p}}\|_F^2, \;\; s.t. \; \vec{\mathbf{w_p}}^{\mathbf{T}}\, \vec{\mathbf{w_q}} = 0, \; \forall \; p \neq q \quad (1)$$

where $\|\cdot\|_F^2$ denotes the Frobenius norm. The constraint in the above objective function guarantees the component detectors are orthogonal (*i.e.* share no redundant information).

In the above formulation, we aim to minimize the reconstruction error between each *individual* component detector and the original detector. We do this because our framework treats component detectors independently in a cascade, therefore the performance of each component detector is important. Thus, we let each component detector to be similar with the original detector (pretrained, having guaranteed performance).

We now move P into the squared terms. We first replicate $\vec{\mathbf{w_0}}$ P times to obtain $\mathbf{W_0} = \underbrace{(\vec{\mathbf{w_0}}, \vec{\mathbf{w_0}} \dots, \vec{\mathbf{w_0}})}_{P \text{ copies}} \in \mathbb{R}^{d\times P}$, where $d = s \times t \times c$ is the length of $\vec{\mathbf{w_0}}$. We then concatenate all unknown variables $\{\vec{\mathbf{w_p}}\}$ into $\mathbf{W} = (\vec{\mathbf{w_1}}, \vec{\mathbf{w_2}} \dots, \vec{\mathbf{w_P}}) \in \mathbb{R}^{d\times P}$, and therefore the constraint in the objective function (1) is equivalent to enforcing $\mathbf{W}$ to be an orthogonal matrix (*i.e.* column vectors in $\mathbf{W}$ are orthogonal to each other). Therefore we can obtain the following learning objective.

$$\min_{\mathbf{W}} \|\mathbf{W_0} - \mathbf{W}\|_F^2, \;\; s.t. \; \mathbf{W^T W} = \mathbf{D}^2 \quad (2)$$

$\mathbf{D} \in \mathbb{R}^{P\times P}$ in the above formulation is an unknown diagonal matrix. This is a form of *quadratic constrained quadratic programming*, and is difficult to solve in general. Let $\mathbf{W} = \mathbf{VD}$ such that $\mathbf{D}$ is an unknown diagonal matrix and $\mathbf{V}$ is an unknown orthonormal matrix (*i.e.* $\mathbf{V^T V} = \mathbf{I}$). Thus the objective function $\rho = \|\mathbf{W_0} - \mathbf{W}\|_F^2$ is equivalent with $\rho = \|\mathbf{W_0} - \mathbf{VD}\|_F^2$.

Minimizing $\rho(\mathbf{D}, \mathbf{V})$ subject to $\mathbf{V^T V} = \mathbf{I}$ is then reduced to a form of the *generalized orthogonal procrustes* problem, and there exists an iterative optimization approach which alternately solves for $\mathbf{V}$ and $\mathbf{D}$ [13]. Here we give a brief sketch of the algorithm. First, we initialize the diagonal matrix $\mathbf{D}$ to be an identity matrix $\mathbf{I}$. Once $\mathbf{D}$ is fixed, minimizing $\|\mathbf{W_0}-\mathbf{VD}\|_F^2$ has a closed-form solution [31]. This is done by observing that $\rho(\mathbf{D},\mathbf{V}) = \|\mathbf{W_0} - \mathbf{VD}\|_F^2 = \text{Tr } \mathbf{W_0^T W_0} + \text{Tr } \mathbf{D^T V^T V D} - 2\,\text{Tr } \mathbf{W_0^T V D} = \text{Tr } \mathbf{W_0^T W_0} + \text{Tr } \mathbf{D^T D} - 2\,\text{Tr } \mathbf{D W_0^T V}$. Thus minimizing ρ is equivalent to maximizing $\text{Tr } \mathbf{D W_0^T V}$. Let $\mathbf{usv^T} = \mathbf{D W_0^T}$ is a SVD decomposition, we have $\text{Tr } \mathbf{D W_0^T V} = \text{Tr } \mathbf{usv^T V} = \text{Tr } \mathbf{s v^T V u} = \sum_i \mathbf{s}(i,i)\mathbf{Z}(i,i)$, where $\mathbf{Z} = \mathbf{v^T V u}$ is orthonormal. Therefore, $\text{Tr } \mathbf{D W_0^T V} = \sum_i \mathbf{s}(i,i)\mathbf{Z}(i,i) \leq \sum_i \mathbf{s}(i,i)$. The maximum value is achieved if and only if $\mathbf{Z} = \mathbf{I_{d\times P}}$, *i.e.* $\mathbf{V} = \mathbf{v I_{d\times P} u^T}$. Once we have $\mathbf{V}$, we can solve for $\mathbf{D}$ using standard least square methods by setting derivatives of each diagonal entry $\mathbf{D}(k,k)$ to 0, *i.e.* $\mathbf{D}(k,k) = \sum_n \mathbf{W_0}(n,k)\mathbf{V}(n,k)$.

3.2. Learning Detection Cascade

We introduce the following lemma before presenting our cascade learning algorithm.

Lemma 1. *The original pre-trained linear object detector* $\vec{\mathbf{w_0}} = \sum_{p=1}^{P} \vec{\mathbf{w_p}}^*$, *where* $(\vec{\mathbf{w_1}}^*, \vec{\mathbf{w_2}}^* \ldots, \vec{\mathbf{w_P}}^*)$ *is the solution to (2).*

Proof. The optimization objective (2) minimizes $\rho = \|\mathbf{W_0} - \mathbf{W}\|_F^2$, which can be re-written as:

$$\begin{aligned}\rho &= \sum_{p=1}^{P} (\vec{\mathbf{w_0}}^{\mathbf{T}}\vec{\mathbf{w_0}} + \vec{\mathbf{w_p}}^{\mathbf{T}}\vec{\mathbf{w_p}} - 2\vec{\mathbf{w_0}}^{\mathbf{T}}\vec{\mathbf{w_p}}) \\ &= P\|\vec{\mathbf{w_0}}\|_F^2 + \sum_{p=1}^{P} \|\vec{\mathbf{w_p}}\|_F^2 - 2\vec{\mathbf{w_0}}^{\mathbf{T}}(\sum_{p=1}^{P} \vec{\mathbf{w_p}})\end{aligned}$$

Since $\vec{\mathbf{w_p}}$ and $\vec{\mathbf{w_q}}$ are orthogonal for any $p \neq q$, we have $\sum_{p=1}^{P} \|\vec{\mathbf{w_p}}\|_F^2 = \|\sum_{p=1}^{P} \vec{\mathbf{w_p}}\|_F^2$. Thus ρ becomes:

$$\begin{aligned}\rho &= P\|\vec{\mathbf{w_0}}\|_F^2 + \|\sum_{p=1}^{P} \vec{\mathbf{w_p}}\|_F^2 - 2\vec{\mathbf{w_0}}^{\mathbf{T}}(\sum_{p=1}^{P} \vec{\mathbf{w_p}}) \\ &= (P-1)\|\vec{\mathbf{w_0}}\|_F^2 + \|\vec{\mathbf{w_0}} - \sum_{p=1}^{P} \vec{\mathbf{w_p}}\|_F^2\end{aligned}$$

The minimum of the above function is achieved when $\vec{\mathbf{w_0}} = \sum_{p=1}^{P} \vec{\mathbf{w_p}}$. On the other hand, we calculate the value of the objective function (2) using the minimizers $(\vec{\mathbf{w_1}}^*, \vec{\mathbf{w_2}}^* \ldots, \vec{\mathbf{w_P}}^*)$. The alternate optimization algorithm in [13] first calculates $\mathbf{V} = \mathbf{v I_{d\times P} u^T}$ using a fixed $\mathbf{D}$. Observing that $\text{rank}(\mathbf{W_0}) = 1$, we have $\text{rank}(\mathbf{D W_0^T}) = 1$ since $\mathbf{D}$ is diagonal, and thus $\text{Tr } \mathbf{D W_0^T V} \leq \mathbf{s}(1,1) = \|\mathbf{D W_0^T}\|_F = \sqrt{\sum_k \mathbf{D}(k,k)^2 \|\vec{\mathbf{w_0}}\|_F^2} = \|\mathbf{D}\|_F \|\vec{\mathbf{w_0}}\|_F$. So we have $\rho(\mathbf{D},\mathbf{V}) = \text{Tr } \mathbf{W_0^T W_0} + \text{Tr } \mathbf{D^T D} - 2\,\text{Tr } \mathbf{W_0^T V D} \geq P\|\vec{\mathbf{w_0}}\|_F^2 + \|\mathbf{D}\|_F^2 - 2\|\mathbf{D}\|_F\|\vec{\mathbf{w_0}}\|_F = (P-1)\|\vec{\mathbf{w_0}}\|_F^2 + (\|\mathbf{D}\|_F - \|\vec{\mathbf{w_0}}\|_F)^2 \geq (P-1)\|\vec{\mathbf{w_0}}\|_F^2$. The minimum value is obtained with the minimizer $(\vec{\mathbf{w_1}}^*, \vec{\mathbf{w_2}}^* \ldots, \vec{\mathbf{w_P}}^*)$ to (2), thus they must satisfy $\vec{\mathbf{w_0}} = \sum_{p=1}^{P} \vec{\mathbf{w_p}}^*$. □

Let $\phi(I_n)$ extract a feature vector (*e.g.* HOG histograms) from exemplar I_n, and $f(I_n, \vec{\mathbf{w_0}}) = \vec{\mathbf{w_0}}^{\mathbf{T}}\phi(I_n)$ be the corresponding linear scoring function. The above lemma shows that the detection score of the original linear detector $f(I_n, \vec{\mathbf{w_0}}) = \sum_p f(I_n, \vec{\mathbf{w_p}})$ on any exemplar I_n. This property offers a very convenient way to learn cascade detector using the component detectors. Following Ambai *et al.* [1], we compute a series of rejection thresholds for each cascade layer p:

$$R_p = \tau - \sum_{j=p+1}^{P} \alpha_j$$

where τ is the threshold of the original detector, $\alpha_p = \max_n f(I_n, \vec{\mathbf{w_p}})$ is an estimate of upper bound detection score using the individual component detector $\vec{\mathbf{w_p}}$. We calculate α_p on the positive training images. Figure 3 visualizes an actual detection cascade by plotting the evaluation areas of each cascade layer.

Discussions. Our formulation of the decomposition algorithm is *data independent*. We assume $\vec{\mathbf{w_0}}$ is a pre-trained detector, therefore reconstructing $\vec{\mathbf{w_0}}$ from orthogonal directions implicitly learns discriminative information from training data. One can adopt a *data dependent* formulation by modifying the objective function ρ such that $\rho = \|\mathbf{W_0^T X} - \mathbf{W^T X}\|_F^2$, where $\mathbf{X} \in \mathbb{R}^{d\times N}$ encodes the training data. Simple algebra shows $\rho = \|\mathbf{W_0^T X} - \mathbf{W^T X}\|_F^2 = \text{Tr}\,(\mathbf{W_0^T} - \mathbf{W^T})\mathbf{X X^T}(\mathbf{W_0} - \mathbf{W})$. The covariance matrix $\mathbf{\Sigma} = \mathbf{X X^T}$ makes minimizing the data dependent objective function much more difficult. Combining with the orthogonal constraint $\mathbf{W^T W} = \mathbf{D}^2$, the optimization is a form of *weighted orthogonal procrustes* problem [34]. However, the numerical approach in [34] is not able to handle large matrices in our case. Moreover, Lemma 1 does not hold for the data dependent formulation. Thus, it is not clear how to learn the rejection thresholds of such a cascade detector in a principled way.

3.3. Rank-1 Approximation

The orthogonal decomposition algorithm shows that the original detector can be decomposed into multiple component detectors with no redundancy. In addition, Lemma 1 shows that we can build a cascade detector using these component detectors by running a simple algorithm to estimate the rejection thresholds. These theoretical results show that we can always represent a linear object detector as an equivalent cascade detector where the reconstruction error is minimized. At the same time, any two component detectors in the cascade share no redundant information. Note that these results do not help improving detection speed; on the contrary, more evaluations are required for the cascade detector than the original detector.

However, in a detection cascade, each cascade layer makes rejections on image regions that pass the test of its

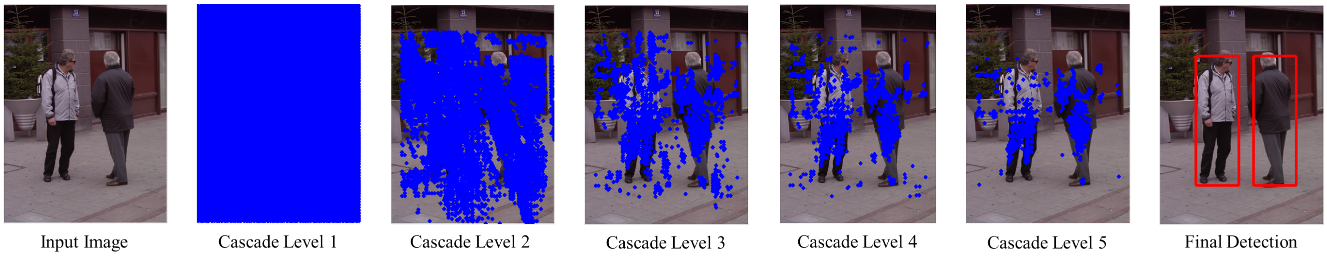


Figure 3: Cascade detection result on an INRIA test image with $P = 5$ cascade layers. For each cascade layer, we plot the image locations where the corresponding component detector is evaluated. This particular cascade detector achieves an average of $2\times$ speed-up on INRIA test set with *no* loss in detection AP (average precision). Final detections show bounding boxes after non-maximum suppression (NMS).

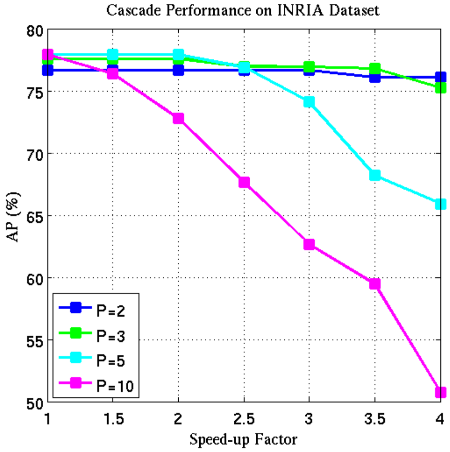


Figure 5: Cascade detection performance on INRIA test set with HOG features. Detection accuracy (AP) decreases as the speed-up factor increases. We vary the global detection threshold for each cascade structure, and plot the AP scores with corresponding speed-up factors. By allowing $2\times$ speed-up, the $P = 3$ cascade detector has 0.3% loss in AP and the $P = 5$ cascade detector has no loss in AP.

preceding cascade layers. Therefore, detectors at earlier cascade stages do not have to be strong classifiers. Based on these intuitions, we perform SVD decomposition of the detector at the first cascade layer to obtain its $\mathrm{rank}\text{-}1$ component corresponding to the largest singular value. We do this for each feature channel separately, and concatenate all these $\mathrm{rank}\text{-}1$ approximations together to form our $\mathrm{rank}\text{-}1$ approximation of the detector. This approximate representation makes the resulting detector less discriminative, but it reduces the time of the convolution operations significantly. On the other hand, $\mathrm{rank}\text{-}1$ approximations of the other cascade layers do not give any speed-up benefit, in that the rest of the detectors are only evaluated at specific image locations, rather than convolving the entire image. Experiments show that our design of such a detection cascade is able to achieve a lot of detection speed-up compared with the original detector.

To summarize, our proposed framework consists of several key steps. Given an pre-trained linear object detector, we first apply the iterative decomposition method in Section 3.1 to obtain a set of component detectors. Then we organize these component detectors in a cascade pipeline and learn the cascade thresholds using the approach in Section 3.2. Finally, we apply $\mathrm{rank}\text{-}1$ approximation on the first cascade layer as described in Section 3.3. The result is a fast cascade detection pipeline that preserves the original detection accuracy.

4. Experiments

We now evaluate the cascade detectors generated by our orthogonal decomposition algorithm. We use two well-known object detection datasets: INRIA Person dataset (INRIA) [6] and PASCAL VOC 2007 dataset (PASCAL) [12]. On both datasets, we conduct ablation studies by varying P (number of component detectors) which gives different cascade structures. We also study the relation between detector speed-up versus the loss in detection performance.

Implementation details. We first use HOG as our image features. We train LDA detectors [19] on both INRIA and PASCAL datasets due to their efficiency in training time. We use "root-only" versions for all such detectors. On INRIA dataset, we train a single-template detector; on PASCAL dataset, we follow [17] to train default "root-only" **LLDA-0** detectors (three aspect ratio clusters with each grouped into left/right splits) on all 20 PASCAL object categories. We modify the single thread convolution code in C provided by [18] into one version that handles two-pass filtering using two vector filters, and another version that evaluates dot product at specified locations on image feature maps. We omit the cost on computing features when measuring the detector speed since our goal is not to accelerate feature computation. The only parameter of our orthogonal decomposition algorithm is the number of component detectors P. The code is implemented in Matlab, and it takes just a few seconds to decompose a pre-trained

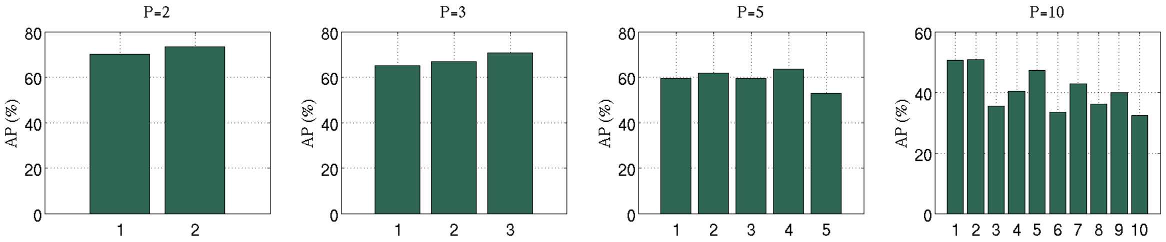


Figure 4: Component detector performance on INRIA test set with HOG features by varying $P = 2, 3, 5, 10$. These component detectors are decomposed from a HOG LDA person detector with $AP = 77.9\%$.

detector. In order to show its generality, we also use our approach to speed up a DPM detector with CNN features (*i.e.* DPDPM [16]) on the PASCAL dataset. All of our experiments were conducted on a 8-core 3.0GHz PC. We use a Tesla K40 GPU to finetue an ImageNet CNN network on PASCAL 2007 dataset and extract the conv5 layer output as CNN features. We use average precision (AP) to measure detection accuracies.

Detector orders in a cascade. We take a simple strategy to order the component detectors in the final cascade. The first detector in the cascade is selected to have the strongest rank-1 approximation. We do this by taking each component detector and measure the ratio between the largest singular value and the sum of all its singular values. Component detectors with highest ratio is placed in the first cascade layer. The rest detectors are simply placed in ascending order in terms of their detection accuracies on a small validation set.

4.1. Results and Discussions

Component detector performance. We test individual component detectors and report their AP (Figure 4) on the INRIA test set. As P increases, the average performance of individual component detectors become weaker. This makes intuitive sense: our decomposition algorithm tries to "split" the original detector into several components while enforcing their discriminative power to match with the original detector as much as possible. Note that we do not restrict the orthogonality constraints to appear in the spatial domain or the feature channel domain. However, the resulting component detectors still show "part-like" properties (Figure 1).

Cascade performance on INRIA. We run experiments with different cascade structures (*i.e.* different P), and analyze the trade-off between detector speed-up and AP loss. We show results in Figure 5. Note that our re-trained version of the HOG LDA detector has an average precision of 77.9% [1] on the INRIA test set. The corresponding average running time of such a detector on each test image is about 0.14 seconds. Our detector size is 18×6. Therefore, the theoretical maximum speed-up factor is about 4.5 (rejecting "all" false positives in the first cascade layer). we observe that different choices of P have an impact on the cascade speed-up factors. The cascade detector performance is more stable when P is small. For example, $P = 2$ allows $4.1\times$ speed-up with a 1.8% AP loss, and $P = 3$ allows $3.8\times$ speed-up with a 1.2% AP loss. The corresponding AP drops quickly when the speed-up factor approaches the theoretical limit (*e.g.* $P = 2$ has 70.0% AP with $4.4\times$ speed-up and $P = 3$ has 70.5% with $4.2\times$ speedup). $P = 10$ gives worst speed-up effect mainly because more complicated structures make the rejection threshold learning step difficult. When P gets larger, the AP drop happened earlier than small P's as the speed-up factor increases. Among all our choices of P, the AP performance of $P = 5$ is slightly better than others within $2.5\times$ speed-up. This reflects a trade-off between the complexity of cascade structures and detector performance. By comparing $P = 2, 3, 5, 10$ on INRIA dataset, we observe that a smaller P tends to give more stable and relatively high AP scores, but the peak performance is somewhat worse than a larger P. Component detectors of a larger P gives worse average performance (Figure 4), but the AP loss after rank-1 approximation is smaller. We conduct a simple experiment to verify our hypothesis. For $P = 2, 3, 5, 10$, we use the first cascade layer in our cascade detector as a separate person detector, and calculate its AP score on INRIA test set. Compared with the original full-rank component detectors, the corresponding AP loss is 23.7%, 23.5%, 16.9%, 15.1% respectively. For large P's, although each component captures less discriminative information, their rank-1 approximations lose less information than their small P counterparts.

Cascade performance on PASCAL. We report mean APs and their corresponding average speed-up factors on the PASCAL VOC 2007 test set (Table 1) for both HOG detector and CNN detector. We retrain the **LLDA-0** [17] detector and **DPDPM** [16] as the baselines. These LDA and DPDPM detectors are then used in our decomposition framework to learn component detectors corresponding to various P. Performance of different cascade structures on sample PASCAL categories are summarized in Figure 6.

[1] Our re-trained detector's AP is higher than the 75.1% reported in [19].

(a) **LLDA-0** [17]

	Original	$P = 2$			$P = 3$			$P = 5$			$P = 10$		
mAP (%)	18.0	16.0	15.4	14.5	16.6	15.9	15.2	17.0	16.3	13.9	16.8	14.7	12.5
Speed-up	1.0	2.6	3.0	3.7	1.9	2.7	3.4	2.1	2.6	3.5	1.8	2.7	3.3

(b) **DPDPM** [16]

	Original	$P = 2$			$P = 3$			$P = 5$			$P = 10$		
mAP (%)	44.4	42.2	42.0	40.2	42.9	42.8	39.7	42.8	42.3	39.1	42.9	40.8	34.6
Speed-up	1.0	2.8	3.0	3.4	2.0	2.8	3.5	2.1	2.5	3.3	1.8	2.7	3.2

Table 1: Mean average precisions on the PASCAL VOC 2007 test set with different speed-up factors and different P's. Top: LDA detectors trained on HOG features. Bottom: DPM detectors trained on CNN features.

	Average Precision (%)							
Speed-up	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.4
U+L	76.4	75.1	69.8	69.8	62.6	62.6	54.2	45.4
RankOne+ORG	76.6	76.0	76.0	76.0	76.0	76.0	70.4	62.8
Proposed	**76.7**	**76.7**	**76.7**	**76.7**	**76.7**	**76.1**	**76.1**	**70.0**

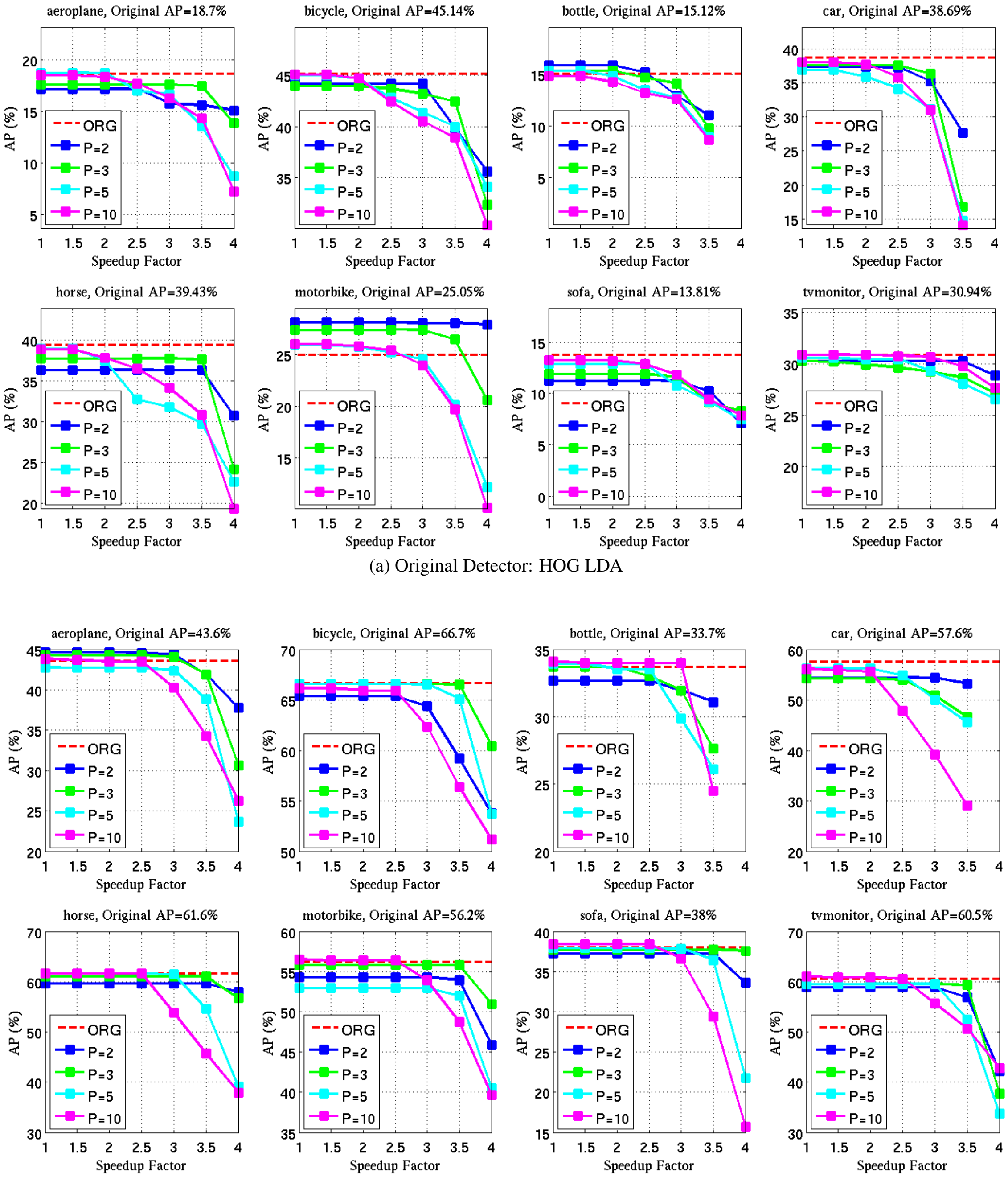
Table 2: Comparison between our $P = 2$ orthogonal cascade detector (**Proposed**), the **RankOne+ORG** cascade detector (original detector in the second layer and its largest rank-1 component in the first layer) and the **U+L** cascade detector (upper half's rank-1 approximation in the first layer and the lower half in the second layer) on the INRIA test set.

The change in AP scores of different detectors could be different when the threshold change is same; but the mean AP scores and the average speed-up factors show a similar trend compared with the cascade performance on INRIA test set. On most categories, $P = 2$ gives lowest peak performance under all speed-up factors, while it gives higher AP score than other P's when the speed-up factor is close to 4. It is interesting to see that, for some classes, our HOG and CNN cascade structures give even better performance than the original detectors. On "motorbike", our 2-layer HOG cascade detector gives a 28.1% AP with 3.3× speed-up; and the 3-layer HOG cascade detector gives AP scores 27.5% with 1.9 speed-up. Similarly on "aeroplane", our 2-layer and 3-layer CNN cascade detectors give 44.6% AP with 2.5× speedup and 44.3% AP with 2.0× speedup, respectively. We believe these are due to the effect of rank-1 approximation of the first cascade layer: although rank-1 approximation makes individual component detectors become worse, the cascade learning stage and the fact that we use the first layer as "weak classifiers" may introduce extra benefit to our proposed cascade detection system (*e.g.* less overfit to training data). A holistic approach that incorporates our decomposition algorithm, rank-1 approximation, and the cascade learning may give a better answer to this question. We leave this as interesting future work.

Connection with SVD. Our decomposition framework seeks for orthogonal decompositions while minimizing the reconstruction error. On the other hand, it is worth pointing out that one can directly apply SVD on the original detector template to obtain a series of rank-1 orthogonal components. However, such decomposition does not minimize the reconstruction error in (1), thus there is no guarantee on the performance of the component detectors. We run two experiments to verify this hypothesis. First, we perform SVD analysis on the 18×6 pre-trained HOG detector trained on INRIA dataset (AP score= 77.9%), and obtain its 6 rank-1 component detectors (corresponding to the 6 singular values). The detection APs are 59.6%, 9.3%, 27.3%, 9.1%, 5.4%, and 0.8% respectively. Such SVD decomposition gives quite poor performance compared with the original detector and our orthogonal components (Figure 4). Next, we design two 2-layer baseline cascade detectors. One is built with the original detector in the second layer and its rank-1 appproximation in the first layer. The other one is an orthogonal cascade detector, and is built with the upper and lower halves of the original detector (zeroing out the rest). The upper half is approximated by its largest rank-1 component and is used as the first cascade layer. We compare these two baseline detectors with our $P = 2$ cascade detector and report the results in Table 2. Our proposed cascade detector clearly outperforms the baseline detectors at all speed-up factors. The difference becomes quite significant when the speed-up factor is larger. This shows that direct rank-1 approximation on the original detector or its naive decomposition gives a much weaker detector.

5. Conclusions and Future Work

In this paper, we present a generic orthogonal decomposition algorithm for decomposing any linear sliding window detector. Theoretical results guarantee a convenient algorithm to learn a cascade detector with the component detectors. In addition, by using rank-1 approximations in the first cascade layer, our cascade detector achieves a lot of detection speed-up while maintaining minimal loss in detection AP. Our approach is not contrained by the choice of image features. We conduct extensive experimental studies using



(a) Original Detector: HOG LDA

(b) Original Detector: DPDPM

Figure 6: Sample cascade detection performance on PASCAL VOC 2007 test set. For all categories, we vary the global threshold of the learned cascade detectors. We report AP changes versus speed-up factors for $P = 2, 3, 5, 10$, as well as the AP for the original detector (**ORG**). Note that the max speed-up factors for "bottle" and "car" classes are both less than 4.0.

HOG and CNN features on two well-known object detection datasets (INRIA and PASCAL) to compare our cascade detector against baseline methods. Without modifying the image features, our proposed method achieves significant speed-up compared with the original model. In future work, we study the possibility of jointly learning orthogonal components as well as the cascade structures with rank-1 approximations. We will also study the detection performance when combining our approach with other speed-up techniques (*e.g.* binary features, integral channel features).

References

[1] M. Ambai and I. Sato. SPADE: scalar product accelerator by integer decomposition for object detection. In *ECCV*, 2014. 3, 4

[2] C. Bauckhage and J. K. Tsotsos. Separable linear classifiers for online learning in appearance based object detection. In *Computer Analysis of Images and Patterns*, pages 347–354. Springer, 2005. 2, 3

[3] R. Benenson, M. Mathias, T. Tuytelaars, and L. Van Gool. Seeking the strongest rigid detector. In *CVPR*. IEEE, 2013. 1

[4] R. Benenson, M. Omran, J. Hosang, and B. Schiele. Ten years of pedestrian detection, what have we learned? In *2nd ECCV Workshop on Computer Vision for Road Scene Understanding and Autonomous Driving*, 2014. 1

[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*, 2014. 1

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*. IEEE, 2005. 1, 5

[7] P. Dollár, R. Appel, and W. Kienzle. Crosstalk cascades for frame-rate pedestrian detection. In *ECCV*. Springer, 2012. 1

[8] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral channel features. In *BMVC*, 2009. 2

[9] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *PAMI*, 34(4):743–761, 2012. 1

[10] L. Dorst. First order error propagation of the procrustes method for 3d attitude estimation. *PAMI*, 27(2):221–229, 2005. 2

[11] K. Duan, D. Batra, and D. J. Crandall. A multi-layer composite model for human pose estimation. In *BMVC*, 2012. 1

[12] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 5

[13] R. Everson. Orthogonal, but not orthonormal, procrustes problems. *Advances in Computational Mathematics*, 1998. 3, 4

[14] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *CVPR*. IEEE, 2010. 1, 2

[15] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9):1627–1645, 2010. 1, 2

[16] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. In *CVPR*, 2015. 1, 6, 7

[17] R. Girshick and J. Malik. Training deformable part models with decorrelated features. In *ICCV*, 2013. 5, 6, 7

[18] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. http://people.cs.uchicago.edu/ rbg/latent-release5/. 5

[19] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *ECCV*. Springer, 2012. 5, 6

[20] M. Hornácek and S. Maierhofer. Extracting vanishing points across multiple views. In *CVPR*. IEEE, 2011. 2

[21] M. Jaderberg, A. Vedaldi, and A. Zisserman. Speeding up convolutional neural networks with low rank expansions. *BMVC*, 2014. 2

[22] I. Kokkinos. Shufflets: shared mid-level parts for fast object detection. In *ICCV*. IEEE, 2013. 3

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1

[24] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *CVPR*. IEEE, 2008. 2

[25] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *PAMI*, 31(12):2129–2142, 2009. 2

[26] J. Luo and H. Qi. Distributed object recognition via feature unmixing. In *ACM/IEEE International Conference on Distributed Smart Cameras*, pages 73–80. ACM, 2010. 2

[27] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Bilinear classifiers for visual recognition. In *NIPS*, 2009. 1, 3

[28] X. Ren and D. Ramanan. Histograms of sparse codes for object detection. In *CVPR*. IEEE, 2013. 2

[29] R. Rigamonti, A. Sironi, V. Lepetit, and P. Fua. Learning separable filters. In *CVPR*. IEEE, 2013. 2

[30] B. Sapp, A. Toshev, and B. Taskar. Cascaded models for articulated pose estimation. In *ECCV*. Springer, 2010. 1

[31] P. H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966. 2, 4

[32] H. O. Song, T. Darrell, and R. B. Girshick. Discriminatively activated sparselets. In *ICML*, pages 196–204, 2013. 3

[33] H. O. Song, S. Zickler, T. Althoff, R. Girshick, M. Fritz, C. Geyer, P. Felzenszwalb, and T. Darrell. Sparselet models for efficient multiclass object detection. In *ECCV*. Springer, 2012. 3

[34] T. Viklands. *Algorithms for the weighted orthogonal Procrustes problem and other least squares problems*. Phd thesis, Umeå University, 2006. 4

[35] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*. IEEE, 2001. 1, 2

[36] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *CVPR*, pages 3360–3367. IEEE, 2010. 2

[37] W. Wang, L. He, P. Markham, H. Qi, Y. Liu, Q. C. Cao, and L. M. Tolbert. Multiple event detection and recognition through sparse unmixing for high-resolution situational awareness in power grid. *IEEE Transactions on Smart Grid*, 5(4):1654–1664, 2014. 2

[38] J. Yan, Z. Lei, L. Wen, and S. Z. Li. The fastest deformable part model for object detection. In *CVPR*. IEEE, 2014. 2, 3

[39] Q. Zhu, M.-C. Yeh, K.-T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *CVPR*. IEEE, 2006. 1