Data-driven Exemplar Model Selection

Ishan Misra Abhinav Shrivastava Martial Hebert
Robotics Institute, Carnegie Mellon University
{imisra,ashrivas,hebert}@cs.cmu.edu

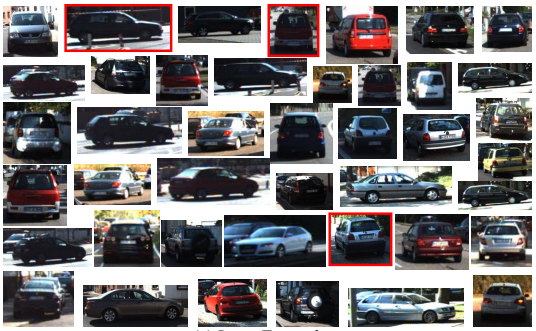
Abstract

We consider the problem of discovering discriminative exemplars suitable for object detection. Due to the diversity in appearance in real world objects, an object detector must capture variations in scale, viewpoint, illumination etc. The current approaches do this by using mixtures of models, where each mixture is designed to capture one (or a few) axis of variation. Current methods usually rely on heuristics to capture these variations; however, it is unclear which axes of variation exist and are relevant to a particular task. Another issue is the requirement of a large set of training images to capture such variations. Current methods do not scale to large training sets either because of training time complexity [31] or test time complexity [26]. In this work, we explore the idea of compactly capturing task-appropriate variation from the data itself. We propose a two stage data-driven process, which selects and learns a compact set *of exemplar models for object detection. These selected models have an inherent ranking, which can be used for anytime/budgeted detection scenarios. Another benefit of our approach (beyond the computational speedup) is that the selected set of exemplar models performs better than the entire set.*

1. Introduction

Object detection in images is a challenging problem because objects in the real world vary greatly in visual appearance. Even objects of a single category, e.g., car, exhibit a lot of diversity in color, shape, size, viewpoint, illumination etc. Capturing all such variation is the key to modeling a good object detector.

Current object detection methods e.g., [8, 11, 18] are trained on training sets e.g., [10] curated to represent such variations. These methods try to address several aspects of this diversity by segmenting the training set into components corresponding to different axes of variation. For example, [11] trains a mixture of detectors, where each mixture corresponds to a "canonical-viewpoint" (based on aspect-ratio); [8] further enriches this model by training mixtures based on visual-subcategories.



(a) Input Exemplars

(b) Selected Exemplars

Figure 1: We start with a large set of exemplars and remove redundancy to select a compact set that captures variation in the data. The selected exemplars are highlighted in red.

An attractive approach advocated in recent works [18, 26] bypasses the problem of determining the "right" mixture by training a separate model for each instance of an object (exemplar model). These exemplar based approaches are attractive because (by design) they represent each instance separately, hence trying to capture all possible variations. But they become impractical as the size of the training data increases; primarily because, at test time, the computation time required to evaluate an ensemble of detectors grows linearly with the size of the training set. Even adding heuristics for greedily selecting (or pruning) detectors [25] does not address this problem.

Ideally, one would like to use as many training samples as possible, e.g., going from the few hundreds in current datasets to many tens of thousands, to get a complete coverage of appearance variations. Unfortunately, this would yield unacceptably high cost at test time for exemplar detectors. Therefore, in this paper we explore the basic question: *Is it possible to process the training images to select a manageable set of detectors at training time, while maintaining detection performance?* Specifically, we propose a two-stage approach: we start by pruning the set of training samples to reduce redundancy, and then we select a subset of the surviving (or pruned) samples that maximizes overall

performance on a held-out set (see Figure 1). Our approach is appropriate for large datasets, that have *both redundancy and variation* e.g., [15], and thus can be compacted. However, it is not relevant for smaller datasets, that do capture variation but have minimal redundancy e.g., [10].

Our contributions are three-fold. First, we show a principled approach to go from raw images to a compact set of ranked detectors, *without making any assumptions about the underlying object category*. Second, we show an effective task-based pruning method and an ensemble selection method, designed for object detectors. Together, they account for both diversity and performance. Finally, we show that it is possible for a smaller set of detectors to give *better performance at a lower computational time* when compared to using the entire set (section 5.4).

We strongly believe that the problem of processing large amounts of training data to produce compact set of detectors has important practical ramifications. For example, a natural way of training vision systems would be to present them with training samples from a large set of videos, e.g., as acquired from discovery/always-on/wearable systems like first-person-vision cameras, surveillance cameras, robotics systems [4, 24], instead of providing a small set of individual frames. Such systems would have to deal with hundreds of thousands of samples, for which intelligent detector selection is imperative. In robotics, or in any other applications requiring time-bounded response with bounded computation [16], it is critical to select and rank detectors out of a large pool. In that respect, we also explore the feasibility of ranking the detectors to make the approach suitable for budgeted detection.

Finally, we would like to emphasize that our aim is to show a scalable way of using Exemplar-SVM (ESVM) detectors, but not to improve them or to introduce a new, complete object detection system. In particular, our focus is on scenarios with a lot of redundancy, e.g., learning from videos of objects. Exemplar detectors are gaining popularity in situations where detection performance is not the only goal. They provide benefits like geometry transfer, label transfer [25, 26], segmentation transfer [35] etc. We hope that the vision community finds this to be a scalable approach of using exemplar detectors.

2. Related Work

The question we explore in this paper has connections to many areas of machine learning and computer vision.

It is related to the general problem of classifier selection which has attracted a lot of interest [2, 3, 7, 19, 32] in the past decade. The selection of a diverse, discriminative and compact set of models is becoming fundamental to many problems today, as we start dealing with increasing amounts of data. These ensemble selection methods can be broadly classified in two categories: static and dynamic. Static methods perform this selection offline without any test time (or run-time) knowledge. For example, [21] which uses k-means to cluster the data, and then find best classifiers for each cluster. Dynamic methods [14, 27] on the other hand, use test time information to do this selection, e.g., recommender systems [27] rely on probing the test set to provide the final ensemble. Our method follows the former category as all the computation is done offline without any test time information.

Caruana et al. [3] pose the ensemble selection problem as a greedy search which minimizes the ensemble error. Their ensemble is formed using a library of various classifiers like Support Vector Machines, Artificial Neural Networks, Decision Trees etc.; and at test time, the predictions are averaged over all models in the ensemble. Their follow-up work [2] performs in-depth analysis of these methods, studying the importance of pre-processing (or model library pruning), dependence on varying training set sizes, initialization of the ensemble, performance metrics etc. We analyze our approach using control experiments in section 5 on similar lines. Kuncheva et al. [20] study various metrics of diversity in classifier ensembles and their relationship with ensemble accuracy. It highlights the issue that there is no strict definition of diversity of ensembles, and some interpretations may not be useful for accuracy. In special cases, diversity has been recognized as the key factor for success of these methods [5, 22].

Most of these methods [3, 16] require that the "scores", within an ensemble, be combined by weighted averaging or majority voting schemes; which is suited for classifier ensembles. However, for the task of object detection, a weighted average strategy is not feasible. A standard object detector (model) is designed to give sparse detections over an image, which may or may not exactly correspond to detections from other models. Hence, the accepted strategy is to do a max-pooling operation on all the detections to produce the final result. Additional complications come from the fact that the performance criteria for the detection task (average precision [10]) also penalizes multiple detections of the same instance. This is a subtle but important difference between classification and detection, which impacts the choice of ensemble selection methods. In this paper, we use the terms classifier and detectors interchangeably.

Other approaches in the computer vision community are to either find "canonical-viewpoints" [11] or visual subcategories [8] for object detectors. However, they specifically require the number of clusters to be specified. Other approaches deal with specific variations; e.g. Bourdev et al. [1], Gu et al. [17] and Park et al. [28], each captures only one type of variation in the object category - either the pose, the viewpoint or the scale respectively. Our proposed approach does not target any particular variation. In fact, it tries to capture discriminative variation in the data in a data-driven manner. In a concurrent work [23], the authors propose an approach for selecting (ranking and greed-

ily sorting) training samples, and show that training on a subset maybe better than training on the entire set (which are also corroborated by our findings).

3. Problem Statement

We start with a large set of labeled images $\mathcal{I}$ containing instances of the object category of interest. We assume that $\mathcal{I}$ is very large and may include redundant images. Because of the size of $\mathcal{I}$, it is not possible to use the entire library of "exemplar" detectors $\mathcal{L}$ trained on each instance of $\mathcal{I}$. Hence, our first objective is to get a considerably smaller set of detectors $\mathcal{D} \subset \mathcal{L}$ while preserving as much of the detection performance from the original set as possible. At test time, we apply all of the detectors in $\mathcal{D}$ to the image and combine them through max-pooling. In addition, we also wish to generate a ranking of the detectors in $\mathcal{D}$ (at training time) so that at test time, we can dynamically choose the top ranked detectors from $\mathcal{D}$ to fit a limited computation budget.

4. Our Approach

We follow a two-step process to obtain $\mathcal{D}$ from $\mathcal{I}$. The first step is to prune the large image set $\mathcal{I}$ to a much smaller set $\mathcal{I}_p$. In the second step, we perform detector selection and ranking using the pruned set $\mathcal{I}_p$. As we shall see later, for $|\mathcal{I}_p| = n$, our selection algorithm is $\mathcal{O}(n^2)$. Additionally, the selection would require training n Exemplar-SVMs (ESVMs) using hard negative mining which is computationally very expensive. Thus, selection becomes tractable only if we obtain a small n by pruning first. Throughout these steps we maintain a separate validation set $\mathcal{V}$ which remains constant. Figure 2 illustrates this process.

4.1. Pruning

The pruning stage is important for two main reasons - computational tractability and preventing overfitting [3]. A natural baseline for pruning is by directly grouping the training exemplars into a smaller set of clusters using appearance-based similarity. For this, we use the recent approach of sparse modeling [9] that considers image features alone. The authors show that on image data this technique outperforms existing methods. Sparse modeling considers all the data points as columns in a matrix Y and tries to find a sparse coefficient matrix C that minimizes the reconstruction error. The coefficients of the matrix C also indicate which data points are more relevant for reconstruction. Thus, they provide a ranking amongst data points. The optimization problem is written in Equation 1. λ represents the tunable Lagrange multiplier.

$$\min_C \lambda \|C\|_{1,2} + \frac{1}{2}\|Y - YC\|_F^2 \text{ s.t. } \mathbf{1}^\top C = \mathbf{1}^\top \quad (1)$$

The main drawback of this approach is that it does not take the detection task into account. We propose an alternative method that uses task-based similarity and show that it is better suited for the pruning.

In our pruning method, we train a weak Exemplar-LDA (ELDA) [18] detector for each bounding box in the set $\mathcal{I}$. We use this approximate detector in place of the full fledged ESVM because training these detectors involves just a few matrix multiplication operations (a very fast operation), which makes training feasible even on very large sets $\mathcal{I}$. We compute detections and corresponding scores of these ELDAs on the set $\mathcal{V}$. We then form a bipartite graph between the detectors and their detections on $\mathcal{V}$. Let M denote the adjacency matrix (of size ELDAs$\times$detections) of this bipartite graph with edge-weights corresponding to the score of the detection. We define a set of common detections ($\mathcal{B}_{ij}$) between detectors i, j to be those detections w which have high overlap, e.g., detected bounding boxes overlapping by at least 80%. We construct an indegree matrix D (of dimension detections$\times$detections) such that $D_{ww} = \text{indegree}_w$ is the number of detectors firing on w, with threshold > 0. Our affinity matrix between the detectors is computed as

$$A = MD^{-1}M^\top \Leftrightarrow A_{ij} = \sum_{w \in \mathcal{B}_{ij}} \frac{\text{score}_w^i \ \text{score}_w^j}{\text{indegree}_w}$$

The term A_{ij} captures the detection task-based similarity between detectors i and j. Hence, A_{ij} is high even if the detectors are poorly trained but give similar detections. Note that we do not use ground-truth labels in $\mathcal{V}$ to form A, so we can use any arbitrary unlabeled dataset instead of $\mathcal{V}$.

The matrix A obtained above has sparse connectivity and accounts only for pairwise similarities between detectors. We can extend this to n-hop similarities by using A^n instead. We then perform affinity propagation clustering [13] on A^n. For each cluster, we nominate its best ELDA as found by affinity propagation. Many existing clustering techniques like k-means, spectral clustering etc. explicitly require the number of clusters as input. In our case the number of clusters captures the variation in the data, and is not known *apriori* for a given dataset. Therefore, we find affinity propagation suitable for our clustering problem. Figure 3 shows some qualitative results of this approach.

It is only natural to ask if we throw away too much information by picking only one sample from a cluster. If the result of clustering is good, then each cluster has detectors which are similar in performance and capture similar variations in the object. Hence, the search space for one cluster is expected to be flat in terms of performance. Picking a locally best representative is an effective technique in such flat search spaces [29].

The cluster centers thus found give us our pruned set of ELDAs. Since each ELDA corresponds to exactly one instance of the object, we get our pruned set $\mathcal{I}_p$.

4.2. Detector selection

Given the considerably reduced subset $\mathcal{I}_p$ generated by the pruning step, we can now train stronger ESVMs on $\mathcal{I}_p$. These form our library $\mathcal{L}_p$. Our goal is to find a compact,

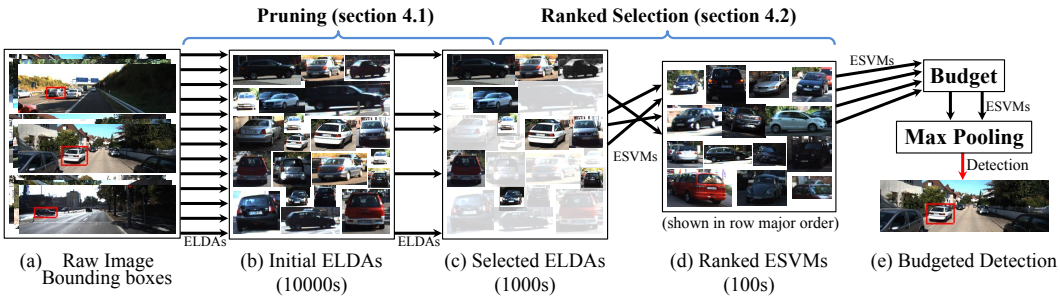


Figure 2: The proposed two stage pruning and selection process for object detectors. We learn a compact, ranked set of detectors starting from the raw images in a purely data-driven manner. This ranked set can be used for budgeted detection.

Figure 3: Qualitative results for Exemplar-LDA based pruning. We show two clusters found by affinity propagation, and highlight the selected representatives. The images are scaled to equal height for better visualization.

ranked subset $\mathcal{D} \subset \mathcal{L}_p$ with diverse and accurate classifiers. Since pruning gives us a considerably reduced library to start with, we can afford to use moderately expensive selection procedures.

Similar to the naïve way of pruning, one can select detectors based on just the exemplar images of the detectors, without using task-based information. For this method, we try the ranking provided by sparse-modeling (obtained by the matrix C in Equation 1). Another obvious strategy is to consider the individual performance of detectors in $\mathcal{L}_p$ on $\mathcal{V}$. We can pick the top g individually best performing detectors. One can use this individual ranking and select entire ensemble ($g = |\mathcal{L}_p|$) [25]. Alternatively, this can be the basis for seeding the ensemble ($g < |\mathcal{L}_p|$) [3]. We call these the "greedy ranking" and the "greedy seeding" algorithms respectively.

In our proposed approach, we use forward selection [32] to incrementally select models from $\mathcal{L}_p$ to form $\mathcal{D}$, while performing hill-climbing on the validation set $\mathcal{V}$. The algorithm does this by a series of greedy choices (also referred to as forward selection). At each iteration, we select, *without replacement*, from $\mathcal{L}_p$ a single classifier such that the *overall* error of the ensemble is minimized on the validation set. We can use any metric m_p to measure this error. We stop the selection as soon as this error starts increasing. The order in which we select the detectors gives us their ranking. Our selection algorithm is detailed in Algorithm 1 (complexity of $\mathcal{O}(n^2)$ for $|\mathcal{I}_p| = |\mathcal{L}_p| = n$). Our algorithm is similar to that of [3]. Such simple selection algorithms have been shown to achieve state-of-the-art results when compared to more complicated approaches [3].

Caruana et al. [2, 3] show that such a selection strategy is prone to overfitting. In addition to pruning, they use methods like bagging and selection *with replacement* to counter this. They use majority voting in the selected ensemble for computing the output. In our case, we cannot use these methods. Our detectors are "exemplar" based and hence cannot use bagging. Selection *with replacement* and majority voting do not help since we do a max-pooling operation on the ensemble. Hence, it does not matter how many times a detector appears in the ensemble. Not using these methods puts our forward selection at a risk of overfitting and not ensuring diversity. However, as we show empirically in sections 5.1 and 5.4, using a large validation set $\mathcal{V}$ avoids overfitting, and diversity may be incorporated in the forward selection metric m_p.

Algorithm 1 Our ensemble selection algorithm

Input: Library of detectors $\mathcal{L}_p$, validation set $\mathcal{V}$, performance metric m_p
Output: A ranked set $\mathcal{D}$ of detectors. $\mathcal{D} \subset \mathcal{L}_p$
Initialize: $\mathcal{D} = \phi$; $\mathcal{R} = \mathcal{L}_p$
while $\mathcal{R} \neq \phi$ **do**
  $t^* = \underset{t \in \mathcal{R}}{\text{argmin}}\ \text{Error}(D \cup t, \mathcal{V}, m_p)$
  if Error($\mathcal{D} \cup t^*$,$\mathcal{V}$,m_p) < Error($\mathcal{D}$,$\mathcal{V}$,m_p) **then**
    $\mathcal{R} = \mathcal{R} \backslash \{t^*\}$; $\mathcal{D} = \mathcal{D} \cup t^*$
  else
    break
  end if
end while
return $\mathcal{D}$ and the order of selection of t^*s as ranking

5. Experiments

We now provide experimental analysis which highlight the advantages of the proposed pruning and selection approach. Our experiments are designed to show that both pruning and selection are extremely important when dealing with large datasets and exemplar detectors. We first present controlled experiments in Sections 5.1-5.3 which analyze both these stages in isolation. Finally, in Section 5.4, we compare our two-stage approach to the relevant baselines. As explained in Section 1, our technique is suitable for datasets that have *both redundancy and variation*.

Dataset: We use the training set from the KITTI dataset [15]. This set contains about 7500 labeled images, sampled from videos captured by a camera mounted on a vehicle, and thus have a lot of redundancy. These images capture a wide range of conditions (illumination, view-point and scale). Therefore, the dataset meets both our requirements of *redundancy and variation*. We consider cars as our object of interest.

Metrics: We use the PASCAL VOC 2007 [10] criterion for detection and measure performance using mean Average Precision (mAP). This detection criterion determines true positives as those with greater than 0.5 intersection over union with ground-truth i.e., the Jaccard coefficient, and counts redundant detections as false positives.

Training: We select a test set of 4450 images and split the remainder set equally for training and validation. Following [15] we consider only large ($>$ 2000 pixels in area), non-occluded and non-truncated bounding boxes for training and testing. We train ESVMs using 5000 random images from Flickr as our negative set [34]. Since, these ESVMs are trained individually, their raw scores cannot be compared. Hence, we use Platt calibration [26, 30] (on validation set) to interpret the SVM scores as probabilities.

Pruning baselines: We use sparse modeling [9] on HOG features [6]. Sparse modeling gave competitive results for finding representative images and like our method, it discovers the number of clusters. We first cluster the images into 5 clusters based on aspect ratio. We then use sparse modeling on each cluster. In their code, we set a very high regularization parameter ($\approx 10^5$) to get a reasonably sized pruned set. We also tried k-means on the ELDAs, HOG Features, RGB values. We varied k in the range $[100, 250]$. Note that we had to repeat the entire pruning/selection pipeline (and training ESVMs) for **each** value of k. We report the results for the best value of k

ELDA Pruning: We perform ELDA Pruning after aspect ratio clustering (similar to the Sparse modeling baseline). We consider 5-hop distances in our affinity matrix.

Selection baselines: We use appearance-based ranking provided by sparse modeling [9]. We use greedy ranking [25] as another baseline. We also evaluate using greedy seeding [3] for the selection algorithm.

Selection Method: We use our selection algorithm as described in Algorithm 1 with the mAP metric [10].

All these pruning/selection methods are described in sections 4.1 and 4.2.

5.1. Diversity in the selection algorithm

We analyze the relation between diversity and performance in our selection algorithm. Specifically, we use sample weighting strategies [12, 33] to see if we gain performance by increasing diversity. Such strategies downweight correct detections and upweight incorrect detections, thus helping gain diversity (we use the downweighting scheme from Adaboost [12]). We also use an extreme version of downweighting ("removal"), where we set the weights of correct detections to zero, essentially "removing" them from further consideration. This sets a bound on performance for the weighting strategies.

Figure 4 shows the results for different weighting schemes on the models obtained after ELDA pruning. We see that doing selection without any downweighting gives the best performance. The primary reason is that the pruned set still has a few bad detectors. Since these detectors make uncorrelated mistakes, they are particularly attractive for any approach focusing just on diversity. Thus, our results suggest that focusing too much on diversity may hurt performance. This observation agrees with [20], where the authors show that just focusing on diversity is not useful when building ensembles on real life data. Our criterion for selection maintains the diversity vs. performance balance.

5.2. Importance of task-based information

In this experiment, we compare the appearance-based ranking generated by sparse modeling to the ranking generated by our selection algorithm. We use sparse modeling [9] to get a pruned set of detectors, which are then ranked. To rank the detectors, sparse modeling ranking looks at the similarity in appearance space of the underlying exemplar images. Figure 5a shows that the ranking provided by the sparse modeling performs poorly at the detection task (the mAP of this baseline is always lower than that of our method). Using our ranking, with just the first 20 detectors we are able to cover 56% of the ground truth as opposed to 24% for sparse modeling.

Figure 5b shows two binary matrices between ground truth boxes and detectors. A black dot at row i, column j indicates that the detector j correctly detects ground truth box i. When using our selection method, we cover more ground truth detections within the first few detectors. Thus, our selection method provides a better coverage of ground truth with smaller number of models as opposed to the selection based on sparse modeling.

5.3. Seeding the selection

We analyze greedily selecting seed models (greedy seeding) (Section 4.2) by varying the number of models picked as a starting point for the selection process. [3] suggests that

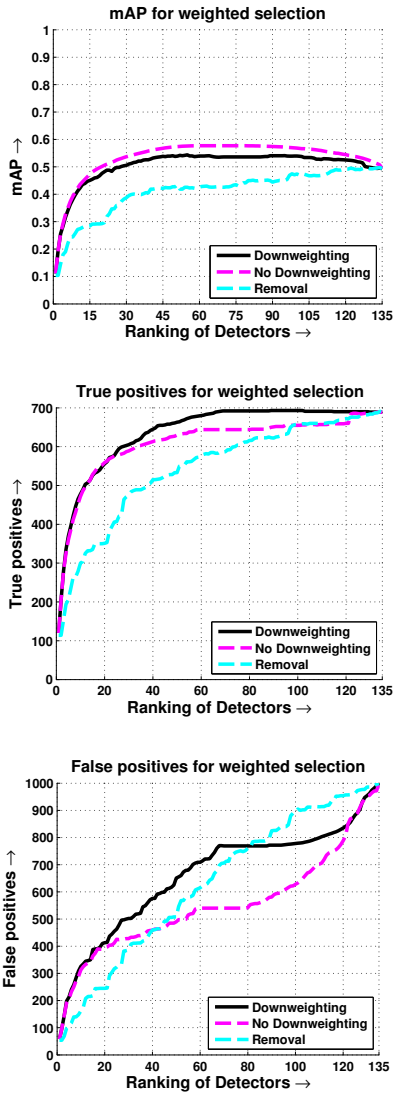
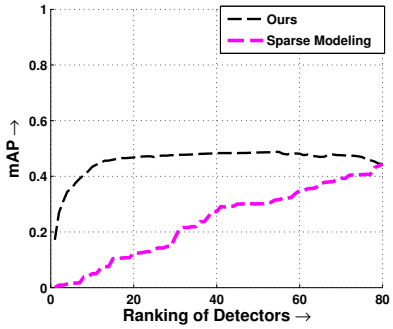
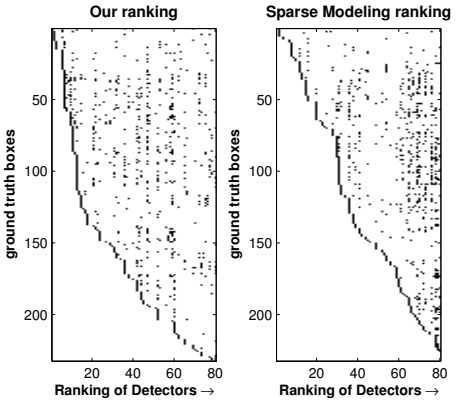


Figure 4: Analysis of different weighting schemes for the selection process. We see that selection without down-weighting performs the best (see section 5.1 for discussion).

this may help improve performance by providing a good search direction for the forward selection step. We try varying number of detectors used to seed our ensemble, and use that as the starting point for our selection algorithm. In our experiments, we found that such seeding harms the performance. Greedy seeding does not take into account how the ensemble behaves *as a whole*. While measuring detection performance, it is also important to minimize redundant detections which the greedy seeding does not account for. Figure 6 shows performance for varying values of initially chosen seeds g. We see that the performance is better without greedy seeding ($g = 0$) and worsens as we increase g.



(a) mAP of sparse ranking vs ours



(b) Our ranking order compared to sparse modeling [9]. See Section 5.2 for details.

Figure 5: Comparison of appearance based ranking vs. detection-performance based ranking. Using detection-performance greatly improves the ranking, both in terms of accuracy and coverage.

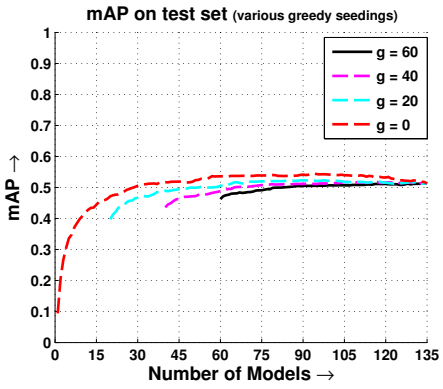


Figure 6: Greedy seedings for our selection procedure. The performance worsens as we increase the number of models picked by greedy seeding. Our approach performs better without any greedy seeding.

5.4. Pruning and Selection

Now we compare the output of our two stage pruning and selection algorithms for different pruning selection combinations. We consider both the pruning methods - sparse

modeling and ELDA clustering. We use our selection algorithm on this pruned output. We also use the greedy ranking [25] for selection as a baseline. We perform five-fold experiments with different training and validation splits, and report the standard error for the mAP values. The results are shown in Figure 7. In these experiments, the minimum size of pruning sets was 114 and 76 for the ELDA and sparse modeling approaches respectively. We see that using our selection method with the ELDA pruning gives the best results. This shows that it is both the pruning and the selection step that help get the performance boost. In our selection algorithm, we stop as soon as performance on the validation set drops (indicated by red dots in Figure 7). We see that this point (63 detectors) holds well even for the much larger unseen test distribution. This gives empirical evidence that our pruning and selection methods do not overfit. Table 1 shows the mAP values for all these methods as soon as the selection terminates, i.e., validation error increases. k-means gives better results than the sparse modeling baseline, but requires an exhaustive search for the value of k.

It would be interesting to see how the pruning/selection approach performs against the entire library of detectors. To study this, we train and use all the 1684 ESVMs for detection without any pruning/selection. This uncovers a surprising and interesting result: it actually *hurts performance* if we use the entire library of detectors (0.311 mAP with 1684 detectors as opposed to 0.632 mAP with 63 detectors). Even after pruning, using the entire library without selection hurts performance as seen in Figure 7 - the mAP value peaks and then decreases as number of detectors increase. This result underscores the importance of the problem being tackled in this paper, and shows that the selection process is *doubly vital*, i.e., for reducing computation complexity and for improving performance.

Table 1: Comparison of our method with baselines (mAP$\pm$ std error). Abbreviations: Sparse Pruning (SP), Greedy Ranking (GR), ELDA Pruning (EP), k-means on ELDA (KE); on HOG (KH); on RGB (KC), Our selection method (OS), Using all ESVMs without selection/pruning (All), Our final method (EP+OS). Best k values for KE=200, KH=100, KC=150.

Method	Detectors selected	mAP	
		Validation	Test
SP+GR	35	0.386 ± 0.011	0.338 ± 0.009
SP+OS	34	0.486 ± 0.008	0.423 ± 0.006
EP+GR	26	0.413 ± 0.012	0.351 ± 0.007
KE+OS	51	0.532 ± 0.009	0.465 ± 0.002
KE+GR	31	0.466 ± 0.009	0.367 ± 0.008
KH+OS	55	0.501 ± 0.011	0.454 ± 0.010
KH+GR	27	0.446 ± 0.007	0.345 ± 0.005
KC+OS	53	0.496 ± 0.010	0.452 ± 0.004
KC+GR	32	0.439 ± 0.012	0.346 ± 0.006
EP+OS	**63**	$\mathbf{0.632 \pm 0.005}$	$\mathbf{0.520 \pm 0.002}$
All	1684	-	0.311

6. Discussion

We proposed a data-driven approach for discriminative and compact exemplar selection. The empirical results show that the problem of ensemble selection for object detectors has significant differences from the one for classifiers. These key differences manifest in our choice for task-based pruning, no seeding and no downweighting to minimize the ensemble error. It is interesting to see that our ranked selection also discards bad classifiers, which helps us to get better performance with fewer detectors on the KITTI dataset.

Acknowledgement: This work was supported in part by NSF Grant IIS1065336 and a Siebel Scholarship. The authors wish to thank Francisco Vicente and Ricardo Cabral for helpful discussions.

References

[1] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *ECCV*, 2010. 2

[2] R. Caruana, A. Munson, and A. Niculescu-Mizil. Getting the most out of ensemble selection. In *ICDM*, 2006. 2, 4

[3] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes. Ensemble selection from libraries of models. In *ICML*, 2004. 2, 3, 4, 5

[4] A. R. Collet, B. Xiong, C. Gurau, M. Hebert, and S. Srinivasa. Exploiting domain knowledge for object discovery. In *CVPR*, May 2013. 2

[5] P. Cunningham and J. Carney. Diversity versus quality in classification ensembles based on feature selection. In *ECML*, 2000. 2

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

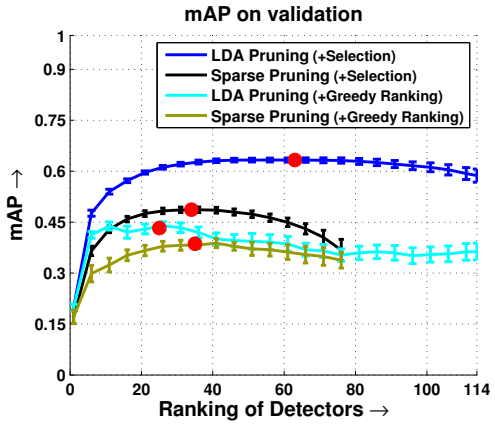
[7] T. G. Dietterich. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*, 2000. 2

[8] S. Divvala, A. A. Efros, and M. Hebert. How important are 'deformable parts' in the deformable parts model? In *Parts and Attributes Workshop, ECCV*, 2012. 1, 2

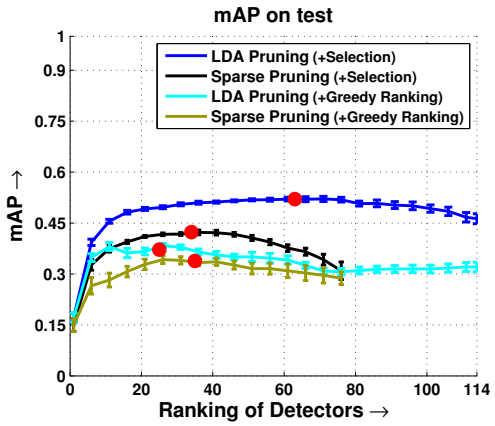
[9] E. Elhamifar, G. Sapiro, and R. Vidal. See all by looking at a few: Sparse modeling for finding representative objects. In *CVPR*, 2012. 3, 5, 6

[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007). 1, 2, 5

[11] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 2010. 1, 2



(a) mAP of on validation set



(b) mAP on test set

Figure 7: Comparison of our system with the baselines. The red dots indicate when our selection algorithm terminates based on validation error. We see that these points generalize to favorable mAP values on the test set (see Table 1). Increasing the number of detectors beyond a certain value, hurts performance.

[12] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *EuroCOLT*, 1995. 5

[13] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 2007. 3

[14] T. Gao and D. Koller. Active classification based on value of classifier. In *NIPS*, 2011. 2

[15] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 2, 5

[16] A. Grubb and J. A. Bagnell. Speedboost: Anytime prediction with uniform near-optimality. In *ICAIS*, 2012. 2

[17] C. Gu and X. Ren. Discriminative mixture-of-templates for viewpoint classification. In *ECCV*, 2010. 2

[18] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *ECCV*, 2012. 1, 3

[19] L. I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004. 2

[20] L. I. Kuncheva and C. J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Mach. Learn.*, 2003. 2, 5

[21] L.I. Kuncheva. Clustering-and-selection model for classifier combination. In *Knowledge-Based Intelligent Engineering Systems and Allied Technologies*, 2000. 2

[22] L. Lam. Classifier combinations: Implementations and theoretical issues. In *Multiple Classifier Systems*, volume 1857 of *Lecture Notes in CS*. 2000. 2

[23] Àgata Lapedriza, Hamed Pirsiavash, Zoya Bylinskii, and Antonio Torralba. Are all training examples equally valuable? *arXiv preprint*, 2013. 2

[24] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, 2012. 2

[25] T. Malisiewicz. *Exemplar-based Representations for Object Detection, Association and Beyond*. PhD thesis, Carnegie Mellon University, 2011. 1, 2, 4, 5, 7

[26] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011. 1, 2, 5

[27] P. Matikainen, R. Sukthankar, and M. Hebert. Classifier ensemble recommendation. In *ECCV*, 2012. 2

[28] D. Park, D. Ramanan, and C. Fowlkes. Multiresolution models for object detection. In *ECCV*, 2010. 2

[29] D. Partridge and W. B. Yates. Engineering multiversion neural-net systems. *Neural Comput.*, 1996. 3

[30] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. *Advances in Large Margin Classifiers*, 1999. 5

[31] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012. 1

[32] D. Ruta and B. Gabrys. Classifier selection for majority voting. *Information Fusion*, 6, 2005. 2, 4

[33] R. E. Schapire. A brief introduction to boosting. In *IJCAI*, 1999. 5

[34] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. A. Efros. Data-driven visual similarity for cross-domain image matching. *ACM Trans. on Graphics*, 2011. 5

[35] J. Tighe and L. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *CVPR*, 2013. 2