Do We Need More Training Data or Better Models for Object Detection?

Xiangxin Zhu[1]
xzhu@ics.uci.edu
Carl Vondrick[2]
vondrick@mit.edu
Deva Ramanan[1]
dramanan@ics.uci.edu
Charless C. Fowlkes[1]
fowlkes@ics.uci.edu

[1] Computer Science Department
University of California
Irvine, CA, USA

[2] CSAIL
Massachusetts Institute of Technology
Cambridge, MA, USA
(Work performed while at UC Irvine)

Abstract

Datasets for training object recognition systems are steadily growing in size. This paper investigates the question of whether existing detectors will continue to improve as data grows, or if models are close to saturating due to limited model complexity and the Bayes risk associated with the feature spaces in which they operate. We focus on the popular paradigm of scanning-window templates defined on oriented gradient features, trained with discriminative classifiers. We investigate the performance of mixtures of templates as a function of the number of templates (complexity) and the amount of training data. We find that additional data does help, but only with correct regularization and treatment of noisy examples or "outliers" in the training data. Surprisingly, the performance of problem domain-agnostic mixture models appears to saturate quickly ($\sim$10 templates and $\sim$100 positive training examples per template). However, compositional mixtures (implemented via composed parts) give much better performance because they share parameters among templates, and can synthesize new templates not encountered during training. This suggests there is still room to improve performance with linear classifiers and the existing feature space by improved representations and learning algorithms.

1 Introduction

Much of the impressive progress in object detection is built on the methodologies of statistical machine learning, which makes use of large training datasets. Consider the benchmark results of the well-known PASCAL VOC object challenge (Fig. 1). We see a clear trend in increased performance over the years as methods have gotten better and training datasets have become larger. In this work, we ask a meta-level question about the field: will continued progress be driven faster by increasing amounts of training data or the development of better object detection models?

To answer this question, we collected a massive training set that is an order of magnitude larger than existing collections such as PASCAL [6]. We follow the dominant paradigm of

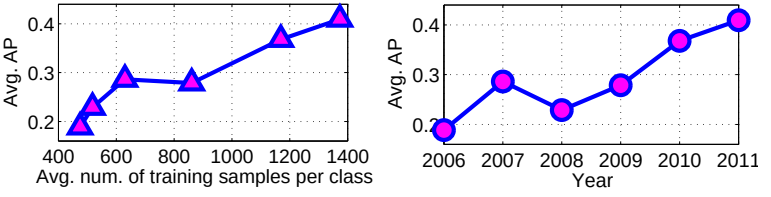


Figure 1: The best reported performance on PASCAL VOC challenge has shown marked increases since 2006. However, it is not clear whether this is due to more sophisticated models or simply the availability of larger training sets.

scanning-window templates trained with linear SVMs on HOG features [1, 3, 7, 10], and evaluate detection performance as a function of the amount of training data and the model complexity.

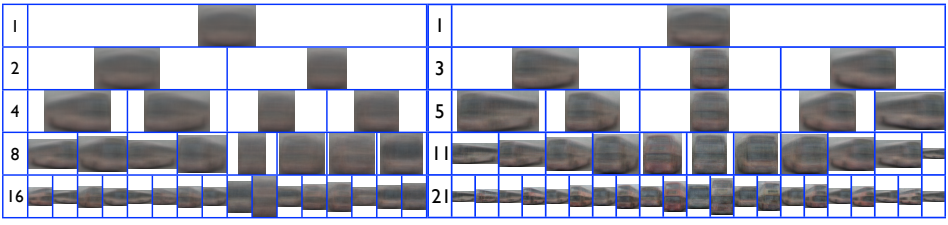
We found there is a surprising amount of subtlety in scaling up training data sets in current systems. For a given model, one would expect performance to generally increase with the amount of data, but eventually saturate. Empirically, we found the bizarre result that off-the-shelf implementations often decrease in performance with additional data! One would also expect that to take advantage of additional training data, it is necessary grow the model complexity, e.g., by adding mixture components to model object sub-categories, 3D viewpoint, etc. However, we often found scenarios in which performance was relatively static even as model complexity and training data grew.

In this paper, we offer explanations and solutions for these phenomena. First, we found it crucial to set model regularization as a function of training dataset using cross-validation, a standard technique not typically deployed in current object detection systems. Second, existing strategies for discovering subcategory structure, such as clustering aspect ratios [7] and appearance features [5] may not suffice. We found this was related to the inability of classifiers to deal with "polluted" data when mixture labels were improperly assigned. Increasing model complexity is thus only useful when mixture components capture the "right" subcategory structure. Finally, we found that is was easier to capture the "right" structure with compositional representations; we show that one can implicitly encode an exponentially-large set of mixtures by composing parts together, yielding substantial performance gains over explicit mixture models. We conclude that there is currently little benefit to simply increasing training dataset sizes. But there may be significant room to improve current representations and learning algorithms, even when restricted to existing feature descriptors.

2 Datasets

We performed experiments using two training datasets: a newly collected dataset containing annotated objects from 11 PASCAL categories and the CMU MultiPIE dataset containing faces from multiple viewpoints.

PASCAL-10X: In order to study detection with large datasets, we built an order of magnitude larger dataset than PASCAL for 11 categories. We collected images from Flickr and annotated them on Mechanical Turk. We took care to ensure that we obtained high-quality bounding box annotations and we manually verified that our larger dataset matches the distribution of the original PASCAL set. Our dataset is unique for its massive number of positive



(a) Unsupervised (b) Supervised

Figure 2: We compare supervised versus automatic (k-means) approaches for clustering images of PASCAL buses. Supervised clustering produces more clear clusters, e.g. the 21 supervised clusters correspond to viewpoints and object type (single vs double-decker). Supervised clusters perform better in practice, as we show in Fig. 6.

examples per category and we have made it available for further research. [1].

CMU MultiPIE: We also use 900 faces across 13 view points from CMU's MultiPIE dataset [8]. Each viewpoint was spaced 15° apart spanning 180°. 300 of the faces are frontal, while the remaining 600 are evenly distributed among the remaining viewpoints. For negatives, we use 1218 images from the INRIAPerson database [3]. Our testset is the annotated face in-the-wild (AFW) [15], which contains images from real-world environments and tend to have cluttered backgrounds with large variations in both face viewpoint and appearance.

3 Mixture models

To take full advantage of additional training data, it is important to be able to grow model complexity. We accomplish this by adding a mixture model that capture "sub-category" structure. If the number of mixtures grows with the number of training examples, this is similar to the non-parametric "exemplar SVM" model of [10]. We now briefly describe how we build these mixtures throughout our experiments.

Unsupervised clustering: For our unsupervised baseline, we cluster the positive training images of each category into 16 clusters using hierarchical K-means, recursively splitting each cluster into $k = 2$ subclusters. To capture both appearance and shape when clustering, we apply PCA to reduce the dimensionality of HOG and append the aspect ratio to the feature. However, a difficulty in evaluating mixture models is that clustering has a non-convex objective and frequently settles in local optima, which may mask variations in performance that we wish to measure across samples of training data.

Partitioned sampling: Given a fixed training set of N_{max} positive images, we want a clustering for $K = 8$ to respect the cluster partitions for $K = 4$; similarly, we would like to construct a smaller sampled subsets, say of $N = \frac{N_{max}}{2}$ images, whose cluster partitions respect those in the full dataset. To do this, we first hierarchically-partition the full set of N_{max} images by recursively applying K-means, as described. We then subsample half the images from each level of the hierarchy to generate a partitioning of $\frac{N_{max}}{2}$ images. We recursively repeat this procedure, generating a dataset of $\frac{N_{max}}{4}$ images by further sampling half the data. This procedure yields a (K,N) partitioning of the data with two properties: (1) for a fixed K, each smaller set is a subset of the larger ones, and (2) given a fixed N, new clusters are constructed by further splitting existing ones. We compute confidence intervals

[1] The dataset can be downloaded from http://vision.ics.uci.edu/datasets/

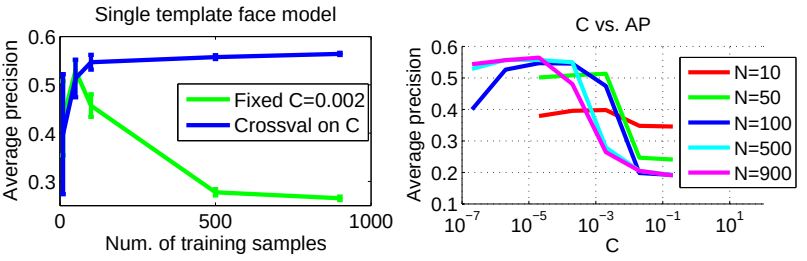


Figure 3: [left] More training data could hurt if we did not pick a proper C. We need to cross-validate on the correct C to produce good results. [right] shows that test performance can change drastically with C. Moreover, the optimal C depends on the amount of positive training examples N.

by repeating this procedure multiple times to resample the dataset and produce multiple sets of (K,N)-consistent partitions.

Supervised clustering: To examine the effect of supervision, we cluster the training data by manually grouping visually similar samples. For CMU MultiPIE, we define clusters using viewpoint annotations provided with the dataset. We generate a hierarchical clustering by having a human operator merge similar viewpoints, following the partitioned sampling scheme above. Since PASCAL-10X does not have viewpoint labels, we generate an "over-clustering" with K-means with a large K, and have a human operator manually merge clusters. Fig. 2 shows an example of the average images for each cluster.

4 Experiments

We performed experiments on 11 PASCAL categories along with experiments on multiview face detection using the CMU MultiPIE training set and the Annotated Faces in the Wild test set [15]. For each category, we train the model with varying number of samples (N) and mixtures (K). We train rigid HOG template models [3] for each cluster independently with linear SVMs [2]. We calibrated SVM scores using Platt scaling [17]. To show the uncertainty of the performance with respect to different sets of training samples, we randomly re-sample the training data 5 times for each N and K following the partitioned sampling scheme of Sec. 3. The best regularization parameter C for the SVM is selected by cross validation. We adopt the PASCAL VOC precision-recall protocol for object detection (requiring 50% overlap), and report the average precision for each run. For diagnostic analysis, we first focus on faces and buses.

4.1 Does more data help performance?

We begin with a rather simple experiment: how does a rigid HOG template tuned for faces perform when we give it more training data N? Fig. 3 shows the surprising result that additional training data can *decrease* performance! For imbalanced object detection datasets with many more negatives than positives, the hinge loss appears to grow linearly with the amount of positive training data; if one doubles the number of positives, the total hinge loss also doubles. This leads to overfitting. To address this problem, we found it *crucial to cross-validate C* across different N. By doing so, we do see better performance with more data.

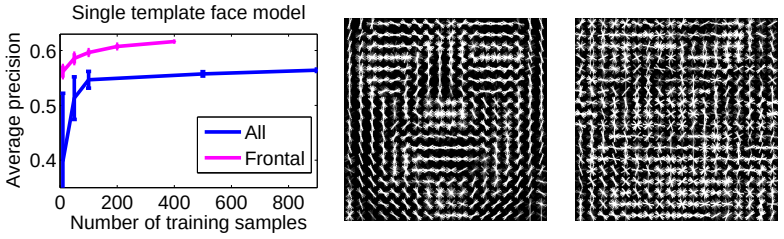


Figure 4: [left] We compare the performance of a single HOG template trained with N multiview face examples, versus a template trained with a subset of those N examples corresponding to frontal faces. The frontal-face template [center] looks "cleaner" and makes fewer mistakes on both testing and training data. The fully-trained template [right] looks noisy and performs worse, even though it produces a lower SVM objective. This suggests that SVMs are sensitive to noise and that we must train models with "clean" data.

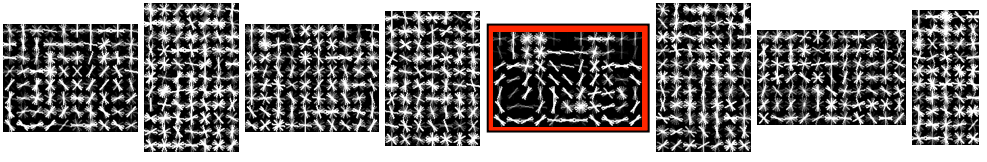


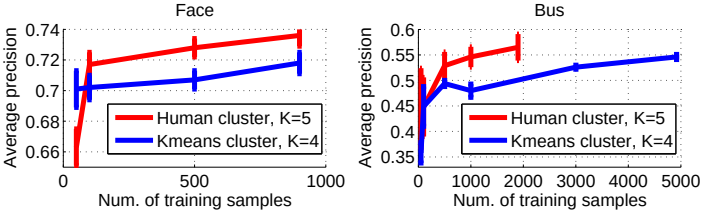
Figure 5: The single clean bicycle template (marked with red) alone achieves ap=29.4%, which is almost equivalent to the performance of using all 8 mixtures (ap=29.7%). Both models strongly outperform a single-mixture model trained on the full training set; this suggests that these additional mixtures to useful to prevent "noisy" data from polluting a single template.

While cross-validation is a somewhat standard procedure, most off-the-shelf detectors are trained using a fixed C across object categories with large variations in the number of positives. We suspect other systems may also be suffering from suboptimal regularization [3, 7] and might show an improvement by proper cross-validation.

4.2 Does clean training data help?

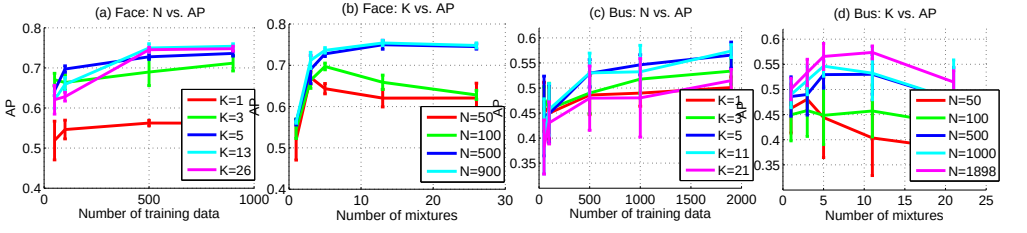
Although proper regularization parameters proved to be crucial, we still discovered scenarios where additional training data hurt performance. Fig. 4 shows an experiment with a fixed set of N training examples where we train two detectors: (1) *All* is trained with with all N examples, while (2) *Frontal* is trained with a "clean" subset of N only containing frontal faces. We cross-validate C for each model for each N. Surprisingly, *Frontal* outperforms *All* even though it is trained with less data.

This outcome cannot be explained by a failure of the model to generalize from training to test data. We examined the training loss for both models, evaluated on the full training set. As expected, *All* has a lower SVM objective function than *Frontal* (1.29 vs 3.48). But in terms of 0-1 loss, *All* makes nearly twice as many classification errors on the same training images (900 vs 470). This observation suggests that *the hinge loss is a poor surrogate to the 0-1 loss* because "noisy" hard examples can wildly distort the decision boundary as they incur a large, unbounded hinge penalty. Interestingly, latent mixture models can mimic the behavior of non-convex bounded loss functions [14] by placing noisy examples into junk



(a) (b)

Figure 6: We compare the human clustering and automatic k-means clustering at near-identical K. We find that supervised clustering provides a small but noticeable improvement of 2-5%.



(a) Face vary N (b) Face vary K (c) Bus vary N (d) Bus vary K

Figure 7: (a)(c) show the monotonic non-decreasing curves when we add more training data. The performance saturates quickly at a few hundreds training samples. (b)(d) show how the performance changes with more mixtures K. Given a fixed number of training samples N, the performance increases at the beginning, and decreases when we split the training data too much so that each mixture only has few samples.

clusters that simply serve to explain outliers in the training set. In some cases, a single "clean" mixture component by itself explains most of the test performance (Fig. 5).

The importance of "clean" training data suggests it could be fruitful to correctly cluster training data into mixture components where each component is "clean". We evaluated the effectiveness of providing fully supervised clustering in producing clean mixtures. In Fig. 6, we see a small 2% to 5% increase for manual clustering. In general, we find that unsupervised clustering can work reasonably well but depends strongly on the category and features used. For example, the DPM implementation of [7] initializes mixtures based on aspect ratios. Since faces in different viewpoint share similar aspect ratios, this tends to produce "unclean" mixtures compared to our latent clustering.

4.3 Does performance asymptote as both K and N grow?

Given the right regularization and clean mixtures, we now evaluate whether performance asymptotes as the amount of training data and the model complexity increase. Fig. 7 shows performance as we vary K and N after cross-validating C and using supervised clustering. Fig. 7a demonstrates that increasing the amount of training yields a clear improvement in performance. Larger models with more mixtures tend to perform worse with fewer examples due to over fitting, but eventually win with more data. Surprisingly, improvement tends to saturate at 100 training examples per mixture and with 13 mixtures. Fig. 7b shows perfor-

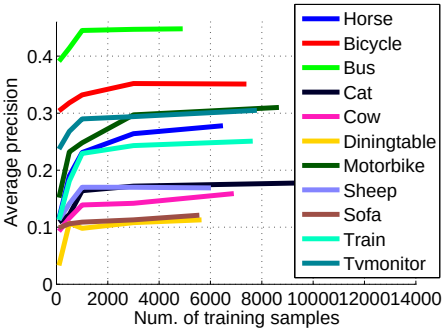


Figure 8: We plot the best performance at varying amount of training data for 11 PASCAL categories. All the curves saturate with a relatively small amount of training data.

mance as we vary model complexity for a fixed amount of training data. Particularly at small data regimes, we see the critical point one would expect: a more complex model performs better up to a point, after which it overfits. We found similar behavior for the Buses category which we manually clustered by viewpoint.

We performed similar experiments for all 11 PASCAL object categories in our PASCAL-10X dataset shown in Fig. 8. We evaluate performance on the PASCAL 2010 trainset since the evaluation set annotations are not public. We cluster the training data into $K = [1,2,4,8,16]$ mixture components, and $N = [50,\ 100,\ 500,\ 1000,\ 3000,\ N_{max}]$ training samples, where N_{max} is the number of training samples collected for the given category. For each N, we select the best C and K through cross-validation. Results across all categories indicate that performance saturates as we increase the amount of training data.

4.4 Is performance saturated for HOG?

Since increasing model complexity does not improve performance, one hypothesis is that we have saturated HOG and are now operating within the Bayes risk. To evaluate this, we compare our model to other representations trained with the same data and feature space. We train and test the latent deformable parts model (LDPM) of [7] as a baseline, which obtains an AP of 80% compared to our 76%. We also compare to the recent DPM of [15] which adds additional supervision, shared parts, and multi-view trees to define spatial constraints. This model further improves performance to 91%, which is clearly a lower bound on the optimal performance. These results demonstrate that *mixture models have not hit the fundamental limits of HOG*. So why did additional training data not allow us to approach 91% AP?

DPMs as large-mixture models: To explain this phenomenon, it will be useful to model a DPM as an exponentially-large mixture of rigid templates. We can then analyze under what conditions a classic mixture model will approach the behavior of a DPM. Let the location of part i be (x_i, y_i). Given an image I, we score a configuration of P parts $(x,y) = \{(x_i,y_i) : i = 1..P\}$ as:

$$S(I,x,y) = \Big(\sum_{i=1}^{P} \sum_{(u,v)\in W_i} \alpha_i[u,v] \cdot \phi(I, x_i+u, y_i+v)\Big) + \sum_{ij\in E} \psi_{ij}(x_i - x_j, y_j - y_j) \qquad (1)$$

where W_i defines the spatial extent (length and width) of part i. The first term defines a local appearance score, where α_i is the appearance template for part i and $\phi(I,x_i,y_i)$ is the feature vector extracted from location (x_i,y_i). The second term defines a pairwise deformation model that scores the relative placement of a pair of parts. When the associated relational graph $G = (V,E)$ is tree-structured, one can compute the best-scoring part configuration

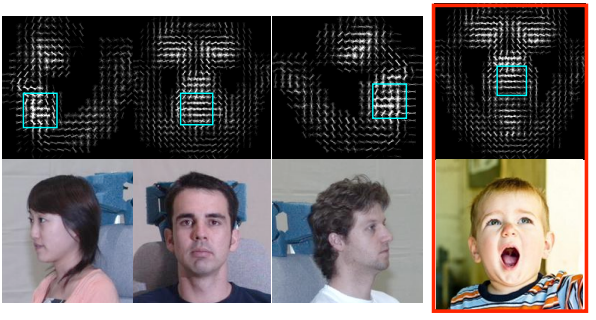


Figure 9: We show three rigid mixtures from a RMP model on the [left], along with their corresponding training images. RMPs share spatially-localized regions (or "parts") between mixtures. Each rigid mixture is a superposition of overlapping parts. A single part is drawn in blue. On the [right], we show a mixture component which is implicitly synthesized by a DPM for a novel test image on-the-fly. In Fig. 10, we show that both sharing of parameters between mixture components and implicit generation of mixture components corresponding to unseen part configurations contribute to the strong performance of a DPM.

$\max_{(x,y)\in\Omega} S(I,x,y)$ with dynamic programming, where Ω is the space of possible part placements. Given that each of P parts can be placed at one of L locations, $|\Omega| = L^P \approx 10^{20}$ for our models.

By defining index variables in image coordinates $u' = x_i + u$ and $v' = y_i + v$, we can rewrite Eqn.1 as:

$$S(I,x,y) = \Big(\sum_{(u',v')\in I} \sum_{i=1}^{P} \alpha_i[u'-x_i, v'-y_i] \cdot \phi(I,u',v') \Big) + \sum_{ij\in E} \psi_{ij}(x_i - x_j, y_j - y_j) \quad (2)$$

$$= \Big(\sum_{(u',v')\in I} w_{xy}[u',v'] \cdot \phi(I,u',v') \Big) + b_{xy} \quad (3)$$

$$= w_{xy} \cdot \phi(I) + b_{xy} \quad (4)$$

where $w_{xy}[u',v'] = \sum_{i=1}^{P} \alpha_i[u'-x_i, v'-y_i]$. For notational convenience, we assume parts templates are padded with zeros outside of their default spatial extent. Therefore, a DPM is equivalent to exponentially-large mixture model where each mixture component is indexed by a particular configuration of parts (x,y). The template corresponding to each mixture component w_{xy} is constructed by adding together parts at shifted locations. The bias corresponding to each mixture component b_{xy} is equivalent to the spatial deformation score for that configuration of parts. DPMs differ from classic mixture models in that they (1) share parameters across a large number of mixtures or rigid templates, (2) "synthesize" new templates not encountered during training, and finally, (3) efficiently search over a large number of templates (making use of dynamic programming).

RMPs: To consider the impact of (1) vs (2), we define a part model that limits the possible configurations of parts to those seen in the N training images, written as $\max_{(x,y)\in\Omega_N} S(I,x,y)$ where $\Omega_N \subseteq \Omega$. We call such a model a Rigid Mixture of Parts (RMP), since it can also be interpreted as set of N rigid templates with shared parameters. Indeed, RMPs are optimized with a discrete enumeration over N templates rather than dynamic programming. RMPs have the benefit of sharing, but cannot synthesize new templates that were not present in the

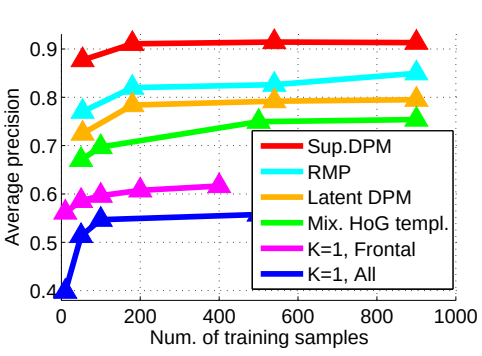


Figure 10: We compare the performance of mixtures models with RMPs and latent/supervised DPMs. A single rigid template ($K = 1$) tuned for frontal faces outperforms the one tuned for all faces (as shown in Fig. 4). Mixture models boost performance to 76%, approaching the performance of a latent DPM (79%). The RMP shares supervised part parameters across rigid templates, boosting performance to 85%. The supervised DPM (91%) shares parameters but also implicitly scores additional templates not seen during training.

training data. We visualize example RMP mixtures in Fig. 9. In Fig. 10, RMPs outperform our cross-validated subcategory mixture model (the green curve), increasing performance from 76% to 85%. Subcategory mixture models learn parameters independently for each distinct viewpoint and deformation, and so may require additional data to achieve the same performance as the RMP model.

While RMPs outperform the latent DPM of [7], they underperform the supervised DPM of [15], even though they are trained with the same supervised part labels of the latter. RMP performance grows slowly with additional training data. In the limit that every configuration is seen in training ($\Omega_N = \Omega$), an RMP must behave identically to a DPM. However, this would require a training set of at least $|\Omega| = L^P \approx 10^{20}$ examples, far more than the number of images we can realistically collect. This suggests *the performance gap (85% vs 91%) stems from the ability of deformable models to synthesize configurations that are not seen during training*.

In summary, part models can be seen as a mechanism for performing intelligent parameter sharing across observed mixture components and extrapolation to implicit, unseen mixture components. Both these aspects contribute to the strong performance of DPMs. Once the representation for sharing and extrapolation is accurately specified, fairly little training data is needed. Indeed, our analysis shows that one can train a state-of-the-art face detector [15] with 50 face images.

5 Discussion and Conclusion

We have performed an extensive analysis of the current dominant paradigm for object detection using HOG feature templates. We view this study as complementary to other meta-analysis of the object recognition problem such as studies of the dependence of performance on the number of object categories [2], dataset collection bias [13], and component-specific analysis of recognition pipelines [11].

An emerging idea in our community is that object detection might be solved with simple models backed with massive training sets, reminiscent of the "big-data" hypothesis [9]. Our experiments suggest an alternative view. Given the size of existing datasets, it appears none of the models we tested will benefit greatly from more data. Instead, the largest gains were in enforcing richer representational structure and constraints within the model.

Another regular hypothesis is that we should focus on developing better features, not better learning algorithms. While HOG is certainly limited, we still see substantial performance

gains without any change in the features themselves or the class of discriminant functions. Instead, the strategic issues appear to be parameter sharing and compositionality. Establishing and using accurate, clean correspondence among training examples (e.g., that specifies that certain examples belong to the same sub-category, or that certain spatial regions correspond to the same part) and developing compositional approaches that implicitly make use of augmented training sets appear the most promising directions.

Acknowledgements: Funding for this research was provided by NSF IIS-0954083, NSF DBI-1053036, ONR-MURI N00014-10-1-0933, a Microsoft Research gift to DR, and a Google research award to CF.

References

[1] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *International Conference on Computer Vision*, 2009.

[2] C. Chang and C. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

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

[4] J. Deng, A. Berg, K. Li, and L. Fei-Fei. What Does Classifying More Than 10,000 Image Categories Tell Us? In *International Conference on Computer Vision*, 2010.

[5] S. Divvala. Visual subcategories. Technical report, PhD thesis proposal, Carnegie Mellon University, 2011. URL http://www.cs.cmu.edu/~santosh/papers/thesis_proposal.pdf.

[6] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

[7] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE TPAMI*, 2010.

[8] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. *Image and Vision Computing*, 2010.

[9] A. Halevy, P. Norvig, and F. Pereira. The unreasonable effectiveness of data. *Intelligent Systems, IEEE*, 24(2):8–12, 2009.

[10] T. Malisiewicz, A. Gupta, and A.A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *International Conference on Computer Vision*, pages 89–96. IEEE, 2011.

[11] D. Parikh and C.L. Zitnick. Finding the weakest link in person detectors. In *Computer Vision and Pattern Recognition*, pages 1425–1432. IEEE, 2011.

[12] J. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999.

[13] A. Torralba and A.A. Efros. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition*, pages 1521–1528. IEEE, 2011.

[14] Y. Wu and Y. Liu. Robust truncated hinge loss support vector machines. *Journal of the American Statistical Association*, 102(479):974–983, 2007.

[15] X. Zhu and D. Ramanan. Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition*, 2012.