Boosting Deformable Part Model by Sample Sharing and Outlier Ablation

Feng Liu[1], Yongzhen Huang[2], Liang Wang[2], and Wankou Yang[1]

[1] School of Automation, Southeast University, Nanjing, 210096, China
[2] National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing, 100190, China

Abstract. The deformable part model (DPM) achieves the best performance on some well known datasets in terms of object detection. Literature springs up to study the success of such a model and hence various methods are proposed to improve it. Yet one import issue, the sensitivity to outliers of the hinge loss,[1] has not been fully studied. In this paper, we take two initiatives to handle this problem: 1) we propose to share samples of one component to others by similarity; 2) we give samples different weights according to their costs. The model is better trained with our proposed method, and we boost the performance of the newly released voc-release 5 [6] model on the challenging VOC 2007 dataset.

Keywords: Object Detection, Deformable Part Model, Sample Sharing, Outlier Ablation.

1 Introduction

Object detection aims to find the object bounding boxes of the target class in a picture, and the deformable part model [3] is one of the most well-known models in this field. The DPM consists of several components with each component be a star model which is made of a root filter and several part filters. Usually, each component has a different aspect ratio[2] in order to catch objects of changing postures, viewpoints and deformations.

Divvala et al. [2] have made a comprehensive evaluation of the DPM and claim that, rather than the usage of deformation parts, the utilization of multiple components seems to contribute the most to achieve a high performance. The reason is that a single linear hyperplane is not capable of separating the sophisticated objects from the backgrounds, as the positive samples tend to scatter in the feature space. Using multiple components is to separate the feature space with several hyperplanes, each of which in charge of a subset of samples with some common properties. For example, in the DPM, samples of the same component have similar aspect ratios, and in Divvala's work, such samples are close in the feature space by Euclidian distance. That is to say, it is better for the samples

[1] DPM is typically trained by the latent SVM, the loss of which is hinge loss.

[2] *aspect ratio = height / width*

C. Sun et al. (Eds.): IScIDE 2013, LNCS 8261, pp. 418–426, 2013.

to have small variations in the same component. Zhu et al. [12] also demonstrate that 'clean'[3] data can help to improve the performance of a mixture model, and the 'clean' data is obtained by clustering. This viewpoint is also supported by the poor performed classes of the DPM on the VOC 2007 dataset, e.g., cat. These classes tend to have more deformations, which lead to a large variation in each component, thus causing an ill-trained model.

Why does large intra-component variation deteriorate the performance, especially for the SVM? It is because hinge loss is more sensitive to outliers compared with the 0-1 loss [11]. The loss suffered is proportional to the distance of a point to the classification hyperplane. In this condition, the hyperplane is more likely to turn towards the outliers to compensate the loss caused by them. In this paper, we consider samples which are dissimilar with the majority of the samples of this component as outliers, and take two initiatives to alleviate the effects of intra-component impurity caused by them:

1. Sample sharing: Instead of restricting one positive sample to be assigned exactly to one component, we relax this constraint by allowing a sample to be used by multiple components according to its similarity with them. Thus each component has more samples to choose.
2. Outlier ablation: A positive sample is given a different weight for each component. The weights of all samples as well as the SVM parameters are jointly learned by minimizing an augmented loss function. In this way, the samples which cause large losses will be given small weights (or be abandoned). The model can thus gain robustness to outliers.

The rest of the paper is organized as follows: we firstly revisit the related work in section 2. Then we propose our method in section 3. Our experimental results are reported in section 4 before we draw a conclusion in section 5.

2 Related Work

The deformable part model [3] is the state of art object detector on many datasets. Papers are published to study its success and lots of improvements are made. Among them, some [7,10] propose to combine the DPM with other cues to boost the overall detection results. Some [1,2] propose to improve the model with carefully initialized components or a learned parts relationship by using finer annotations.

Specially, Divvala et al. [5] attribute the DPM's success largely to the usage of multiple components. They split components by clustering the HOG features of all samples, replacing the aspect ratio heuristic used by the original DPM. With a more reasonable component initialization, the samples assigned to each component become 'cleaner', thus the model gets better trained. Zhu et al. [12] evaluate the influence of a variety of factors to the object detector's performance and advocate that we need 'clean' data to train our model. They also show that

[3] The data is called 'clean' when they have small variations.

clustering all the samples (by K-means) and using samples of the same cluster to train a separate component will help. They ascribe the bad performance of the model trained using unpurified data to the hinge loss, which is vulnerable to outliers. Our approach differs from theirs because we try to start directly from the defects of the hinge loss by modifying the loss function and generate virtual data by sample sharing. So our model does not depend on specific component initialization strategies. Instead, our method is easy to combine with them.

Our work is also related with [8] which tries to share samples between datasets and [9] which shows the possibility of training an object model with only one positive sample (yet many negative samples). However, both the final goal and the means between their methods and ours are different.

3 The Proposed Method

In spite of great achievements the DPM has made, it still performs not so well on some classes whose instances vary greatly in appearance, e.g., cat, cow. Keeping the same number of components will make these classes have a larger intra-class variation than others. However, it will be more complex and slow to inference if we use a model with more components. Moreover, the total number of instances hinders us from doing this. So we seek to improve the model without changing the number of components used. Fortunately, we find in [12] that the sensitivity of hinge loss might partly responsible for the degraded performance by impure data. We also find that better trained root filter will benefit the part initialization process, which is crucial to achieve a good performance by latent SVM. Before proposing the improving strategies, we firstly review the training process of the DPM.

3.1 Deformable Part Model Revisited

The Deformable part model is composed of several components where each component is a star model which in charge of a specific subset of objects for a class. Typically, one component includes a root filter and multiple part filters. During training only the bounding boxes of an object is observed, so the component index and part location must be inferred by the model. Thus the model is trained by the latent SVM in a stagewise way which behaves like the EM algorithm.

Stage 1. The positive samples (objects annotated by bounding boxes) are split into m groups according to their aspect ratios. For each group of positive samples, we train a separate SVM with randomly chosen negative samples. The learned weights are used to initialize the root filters of the mixture model.

Stage 2. The mixture model with only root filters is refined by letting each component choosing positive samples which fit it best. In this step, each positive sample can only be and must be assigned to one component for retraining.

Stage 3. The deformation parts are initialized from the refined root filter by an energy coverage rule. Then the new model is retrained on the full dataset with latent detection and some data-mining techniques for the negative samples.

3.2 Sample Sharing

In object detection, each object is annotated with a bounding box in an image, and we call it an object level sample x. In general, x can be used by only one component M_c, and we denote the sample for the c-th component as x^c. However, the strategies of assigning a sample to a specific component is usually heuristic or not precise, e.g., aspect ratio, especially in the first stage. What's more, for some object classes, samples of different components are similar after an appropriate transform, e.g., resizing a bus from this aspect ratio to another. If a sample fits several component models well and we just use it to train one of them, it is a waste of samples. So we draw samples for all component models from a single bounding box and this is called *sample sharing.*

The benefit of sample sharing is that we can increase the number of samples a component can choose, and it is implemented as follows. For the first stage, we just resize the sample of this component to another, since all positive samples are warped. As to the stages afterward, we allow each component to extract a sample x^c from a particular bounding box x. Note that to be selected, x^c has to be the highest scoring hypothesis for this component and satisfies the overlapping rule. The usage of a sample is determined by the strategy illustrated in the next subsection.

3.3 Outlier Ablation

For a class of large intra-class variation, it is difficult for a model M_c to account for all positive samples assigned to it, because hinge loss is rather sensitive to outliers.[4] Putting too much efforts to fit points away from the majority of the data points will deflect the classification hyperplane to the outliers, thus more low cost data points are misclassified. Though the total loss decreases, the error rate, however, increases. Intuitively, those points causing great losses should be abandoned or be given small weights to prevent them from dominating the total loss. So instead of trying to fit each positive sample well, we might as well focus on those representative ones. We now select samples from the candidate samples set by optimizing the following objective function:

$$\min_{\beta,\alpha} \sum_{c=1}^{m} \sum_{i=1}^{n_p} \alpha_{c,i} \ell(\beta_c, x_i^c, +1) + \sum_{c=1}^{m} \sum_{j=1}^{n_n} \ell(\beta_c, x_{j,c}, -1) + \lambda_1 \|\beta_c\|_2^2 + \lambda_2^c \mathcal{R}(1-\alpha), \quad (1)$$

where $\alpha_{c,i} \in [0,1]$ is the weight of x_i^c,[5] and it is initialized to one if x_i originally belongs to component c, otherwise it is set to zero. β_c is the parameter for the c-th component, and m, n_p, n_n are the number of components, positive instances and negative samples respectively. $\mathcal{R}(\cdot)$ is a regularization term which can either be the l_1 norm or the l_2 norm. $\ell = \max(1 - y_i \beta_c^T x_i^c, 0)$ is the hinge loss. λ_1, λ_2^c are hyperparameters which control the weights of the regularization terms.

[4] In this paper an outlier is a data point which is far away from the majority of the data points, not limited to those mislabeled data.

[5] x_i^c is the shared sample of x_i for component c and x_i is the i-th object sample. $x_{j,c}$ is the j-th negative sample for component c.

In this way, a positive sample will be given a small weight if its cost is very large. The hyperparameter λ_2^c can control the total members of samples used for each component, which means the bigger the λ_2^c, the larger amount of samples will be chosen. Compared with a standard SVM, the slope of the above objective is much smaller when the weight multiplied is less than one, which makes the model robust to outliers. One can verify the above objective is an upper bound of the 0-1 loss when l_1 norm is used and λ_2 is larger than one. So we opt to take l_1 norm as our regularizer.

The objective is biconvex, which is convex when optimizing one parameter while fixing the other. Suppose we have trained β and obtained the cost for each sample. Let us denote the cost as $L = (\ell_1^1, ..., \ell_1^m, ..., \ell_n^1, ..., \ell_n^m)$, then the problem becomes: $\min_\alpha L^T\alpha + \sum_{c=1}^m \lambda_2^c(1-\alpha^c)$ which is a linear programming problem, and we can get a analytical solution by coordinate descent. The answer is : $\alpha_i^c = 0$, if $l_i^c > \lambda^c$ and $\alpha_i^c = 1$, otherwise. In practice we take the top scoring p percentage samples to train one component and regard the rest as outliers. The theoretical support for being able to adopt this alternative measure is that for the positive sample, the lower the score the higher the loss. If the loss of a sample x_i^c is above λ_2^c, it will be dropped. It is equal to take the top scoring ones. This is how *outlier ablation* takes place. Similar measures have also been taken by Gaidonet al. [4], however, no theoretical analysis is given in his work.

3.4 DPM Equipped with the Proposed Strategies

We now integrate the two strategies developed above into the DPM. Specifically, we employ both methods on the first stage. We firstly initialize the root filter according to the aspect ratio heuristic and score the samples of the candidate samples set by the trained filters. The top scoring $p_1\%$ samples are chosen to retrained the model. On the second and third stage, we just use the outlier ablation strategy, because in practice, we find that the shared samples always score lower than the original ones. The percentage of samples used in stage two and stage three are denoted as $p_2\%$ and $p_3\%$ respectively. Usually, we take $p_1 < p_2 < p_3$.

4 Experiments

In this section, we firstly give an introduction of the dataset and experimental settings in 4.1, then we show some primacy results on the cat class of the VOC 2007 dataset[6] in 4.2. At last, the results of all classes are reported in 4.3.

4.1 Dataset and Experimental Settings

The PASCAL VOC 2007 dataset is chosen to evaluate our proposed model. It is a challenging dataset which consists of thousands of images of real world scenes

[6] “http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html”

Table 1. Performance achieved by the 5 models on the VOC 2007 cat class

models	M1	M2	M3	M4	M5
AP	11.8	14.2	23.0	24.9	25.5

over 20 classes. The unbalanced distribution of images over classes and large intra-class variations both make it a tough task.

Our model is composed of 3 components with each component of 8 parts. In the first stage, the number of samples of a component c to retrain the model is set to $0.3n_c$, where n_c is the amount of samples assigned to this component initially. We retrain the model of the first stage twice. The retraining of the second stage loops for two times with $0.6n_c$ samples. For the third stage, we firstly use $0.6n_c$ samples for six rounds and use all samples in the next three rounds. The purpose is to firstly train the appearance filters well with 'clean' data and then use all the samples to learn the parameters of the deformation features. The hyperparameters p_1, p_2, p_3 are tuned on the validation set. Though setting different values of these parameters for each class may get better performances, we use a unified term of value to ensure the fairness of comparison.

4.2 Some Primary Results

Before testing it on all classes, we firstly test several models on the cat class of the VOC 2007 dataset. The reason for using this class is that cats are flexible objects and this class is of large intra-class variation, thus it is perfect for testing our proposed method. We compare the performance of the following methods:

M1: A mixture model with 3 components, and each component just contains a root filter. It is trained using the framework of voc-release 5 [6], however, no parts are added. This corresponds to training a DPM which just uses the first two stages.
M2: A mixture model similar to M1. However, we use the proposed two strategies in the first stage and outlier ablation in the second stage.
M3: The original DPM implemented in voc-release 5 which contains 3 components with each component be of 8 parts.
M4: The DPM following the same settings as M3, yet with the proposed two methods used in the first stage and outlier ablation used in the second stage.
M5: The DPM as described in section 4.1.

We list the average precision (AP) achieved by each model in Table 1, and visualize the models trained by the first two methods in Fig. 1. From the table, we can see that our method significantly improve the mixture model, especially the one without parts. Possible reason is that a simple model doesn't have the ability to handle large variations, so the model misclassifies many easy samples in order to catch the outliers. Our model just focuses on the representative samples,

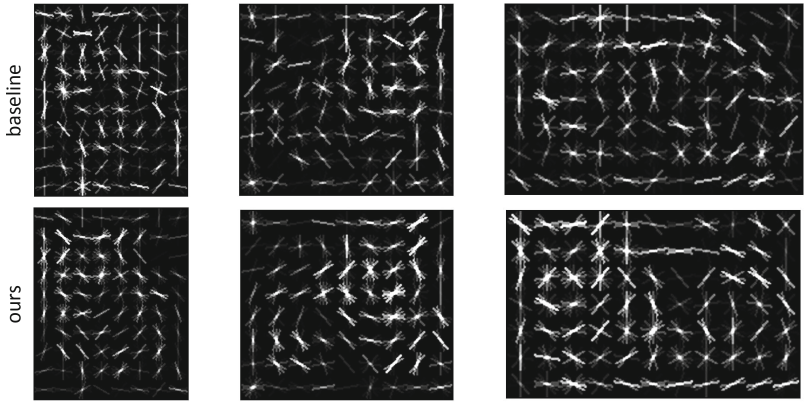

Fig. 1. Visualization of the original mixture model (M1, top row) and the one adopted our improvements (M2, bottom row) on the cat class. Our model has a more clear contour and has large weights on some specific locations. One can see that each component corresponds to a particular posture.

Table 2. Comparison of the original DPM with the improved model on the full VOC 2007 dataset (better performing ones are in bold). Our model beat the original DPM on most classes, especially those of large intra-class variation.

class	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
M3 [6]	**33.2**	**60.3**	10.2	16.1	**27.3**	54.3	**58.2**	23.0	20.0	24.1	26.7	12.7	58.1	**48.2**	**43.2**	12.0	21.1	**36.1**	**46.0**	**43.5**	33.7
M5	33.1	59.0	**12.5**	**17.5**	26.2	**55.3**	57.7	**25.5**	**21.7**	**27.1**	**32.6**	**13.2**	**59.7**	46.5	42.3	**14.0**	**23.8**	35.2	44.3	39.0	**34.3**

so the performance becomes better. After adding parts, a model can handle more variations, so the margin becomes smaller. The boosted performance of M4 over M3 proves that the initialization of part filters given by our model is better than the original one. With a good initialization, the latent SVM gets better trained. This can also be illustrated in Fig. 1 as our model learns a much clearer contour and is quite confident about specific gradient orientations at a fixed location.

4.3 Results on the PASCAL VOC 2007 Dataset

To show the effectiveness of our proposed method, we evaluate it on all classes of the VOC 2007 dataset, and the results are reported in Table 2. From the table we can see that our method outperforms the original DPM on 11 classes, and the margin is quite large on classes which have large intra-class variations. However, there still exist some classes on which we fail to beat the original model. The reason is that training a model is a tradeoff between data purity and the number of training samples [12], the performance will decrease if we use only a subset of the 'clean' data. Consequently, the mean average precision over all classes improves by 0.6. This improvement is impressive since the newly released voc-release 5 is the best object detector when only HOG feature are used.

5 Conclusion

In this paper, we give a theoretical analysis on how data impurity affects the DPM and propose two strategies, sample sharing and outlier ablation, to alleviate the harms caused by it. The sample sharing strategy seeks to enlarge the candidate samples set size for each component, while the outlier ablation scheme tries to fit only the representative samples by dropping the outliers. With the above strategies, models of most classes get better trained, especially those with large intra-class variation. Currently, however, there are no effective ways to choose the hyperparameters for sample selection, and we can only determine them by validation. So in future, we want to develop an explicit purity measure and find a way to automatically learn the hyperparameters.

Acknowledgement. This work is jointly supported by National Natural Science Foundation of China (61175003, 61135002, 61203252, 61005008, 61273023), Hundred Talents Program of CAS, and Tsinghua National Laboratory for Information Science and Technology Cross-discipline Foundation.

References

1. Azizpour, H., Laptev, I.: Object detection using strongly-supervised deformable part models. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 836–849. Springer, Heidelberg (2012)
2. Divvala, S.K., Efros, A.A., Hebert, M.: How important are "Deformable parts" in the deformable parts model? In: Fusiello, A., Murino, V., Cucchiara, R. (eds.) ECCV 2012 Ws/Demos, Part III. LNCS, vol. 7585, pp. 31–40. Springer, Heidelberg (2012)
3. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. TPAMI 32(9), 1627–1645
4. Gaidon, A., Marszalek, M., Schmid, C., et al.: Mining visual actions from movies. In: BMVC 2009 (2009)
5. Gao, T., Stark, M., Koller, D.: What makes a good detector? – structured priors for learning from few examples. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part V. LNCS, vol. 7576, pp. 354–367. Springer, Heidelberg (2012)
6. Girshick, R.B., Felzenszwalb, P.F., McAllester, D.: Discriminatively trained deformable part models, release 5, `http://people.cs.uchicago.edu/~rbg/latent-release5/`

7. Gu, C., Arbeláez, P., Lin, Y., Yu, K., Malik, J.: Multi-component models for object detection. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part IV. LNCS, vol. 7575, pp. 445–458. Springer, Heidelberg (2012)
8. Lim, J.J., Salakhutdinov, R., Torralba, A.: Transfer learning by borrowing examples for multiclass object detection. In: NIPS 2011 (2011)
9. Malisiewicz, T., Gupta, A., Efros, A.A.: Ensemble of exemplar-svms for object detection and beyond. In: ICCV 2011, pp. 89–96 (2011)
10. Mottaghi, R.: Augmenting deformable part models with irregular-shaped object patches. In: CVPR 2012, pp. 3116–3123 (2012)
11. Xu, L., Crammer, K., Schuurmans, D.: Robust support vector machine training via convex outlier ablation. In: Proceedings of the National Conference on Artificial Intelligence, vol. 21, p. 536
12. Zhu, X., Vondrick, C., Ramanan, D., Fowlkes, C.: Do we need more training data or better models for object detection? In: BMVC 2012 (2012)