CLUSTERED EXEMPLAR-SVM: DISCOVERING SUB-CATEGORIES FOR VISUAL RECOGNITION

Nataliya Shapovalova *Greg Mori*

School of Computing Science, Simon Fraser University, Canada

ABSTRACT

We present a novel algorithm for image classification that is targeted to capture class variability. A single model is often not sufficient to represent a category since categories can vary from large semantic classes to fine-grained sub-categories. Instead, we develop a representation based on discovering visually similar sub-categories within a given class. We introduce a novel Clustered Exemplar SVM classifier which incorporates data-driven and exemplar focused discovery. Semi-supervised learning is employed for training each C-eSVM classifier. We evaluate our approach on two datasets and demonstrate the efficacy of our method over standard Exemplar SVM.

Index Terms— visual recognition, sub-categories

1. INTRODUCTION

Classification of images is one of the main problems in computer vision. Real-life categories are complex, and often contain large variations. General classifiers are broad and often cannot capture all variabilities in the data. Sub-categorization partially solves this problem. It helps in several cases: to distinguish between different viewpoints (e.g. [1, 2]), and to distinguish fine-grained sub-categories (e.g. [3]). An extreme case of sub-categorization is presented by Malisiewicz *et al.* [4], where each exemplar forms a sub-category of its own, and a general classifier is replaced by an Ensemble of Exemplar SVMs. However, since they are trained only with one positive example, the resulting eSVMs have difficulty in capturing the details of a class.

Another drawback of general classifiers is a lack of interpretability. Using sub-categories enriches the interpretation of the output. In addition, eSVM allows metadata transfer (e.g., 3D layout, attributes, etc.) from training to test examples. This, however, is more challenging with large sub-categories.

In our work, we extend the eSVM approach and introduce a *Clustered Exemplar SVM (C-eSVM)* classifier. We believe that each category contains different small groups (a "cluster") of examples representing either a fine-grained sub-category or a particular viewpoint. These clusters are not as tiny as one exemplar, but also not as all-encompassing as broad category models. The goal is to discover such clusters automatically and learn a separate C-eSVM classifier for each of them. In other words, instead of using just one positive example as in [4], a cluster of positive examples is used for learning each C-eSVM classifier. However, C-eSVM is still exemplar oriented: each cluster originates from one exemplar in the dataset. We design a novel algorithm based on semi-supervised learning to discover such clusters given an initial exemplar. Finally, C-eSVMs are used to form a mid-level representation of images: each image in the dataset is represented by a vector containing the response scores from all C-eSVM classifiers. The new features are used to learn a final classifier for each category in the dataset. During the test we perform not only classification, but also attribute transfer from training examples to a new unseen example.

The paper is organized as follows. In Sec. 2 we review in detail related work. In Sec. 3 we introduce C-eSVMs and demonstrate how we apply semi-supervised techniques to automatically discover clusters. Sec. 4 contains experimental results and analysis. The paper is concluded in Sec. 5.

2. RELATED WORK

Our work contains aspects of sub-categorization, mid-level image representations, and semi-supervised learning. Below, we briefly review each of these areas.

Sub-categorization: The goal is to automatically create a mixture of models that can capture variation of the data. One of the common approaches to this problem is based on Latent SVM [1], where sub-category assignment is modeled by latent variables. Initialization of latent variables is a critical step, and usually clustering with some heuristics is used. Felzenszwalb *et al.* [1] used bounding box aspect ratio as a criterion for sub-category assignment. Yang and Toderici [3] clustered examples based on co-watch data during learning YouTube category models. Similarly, Gu and Ren [2] used a normalized cut clustering to determine initial viewpoint categories. Another strategy was presented by Hoai and Zisserman [5] where clustering and the learning criteria are combined into one objective function. Finally, as previously mentioned, an extreme case for sub-category learning is exemplar SVM [4], where each sub-category has just one example.

Mid-level features: Mid-level features represent meaningful visual concepts that can describe an object or action.

Farhadi *et al*. [6] use attributes to represent images and videos. Li *et al*. [7] introduced an "Object bank", which consists of thousands of detectors for different object categories. Similarly, an "action bank" citeposelets contains thousands of detectors of primitive actions, and "poselets" [8] is based on detectors for pose parts. As an alternative to supervised learning mid-level features, unsupervised approaches discover mid-level features automatically [9, 10, 11]. However, these approaches work on a patch-level, resulting in complex procedures for patch selection and pruning.

Semi-supervised learning: A comprehensive overview of Semi-Supervised Learning (SSL) from the machine learning perspective is presented in [12] and [13]. The broad goal of SSL is to include unlabeled data into the training process. Here we focus on work that uses SSL for the problem of sub-categorization and attributes in computer vision.

Gu and Ren [2] combine semi-supervised learning with mixture models for viewpoint categorization. Since only partial labeling of viewpoints is known, self-training is applied to infer all the labels initially. Parikh and Grauman [14] use SSL to discover a set of discriminative and semantically meaningful attributes. Active learning is employed to get the necessary labels for the discovered attributes. Choi *et al*. [15] select unlabeled examples based on the attributes. Unlabeled images that are likely to have the same attributes as images in the training data are added to the training set. Similarly, Chen and Grauman [16] augments training data by selecting frames from unlabeled videos based on the poses of actions in training images.

3. CLUSTERED EXEMPLAR SVM

Our goal is to learn a classifier for each exemplar in the dataset, such that these classifiers hold two properties: (i) they represent a cluster of examples from one category (in other words, a sub-category) and (ii) be exemplar oriented. Further, these classifiers are used to form a new mid-level representation of images based on their response scores. Given a new representation of training examples, a final binary classifier is learnt for each category. An overview of our approach is presented in Fig. 1.

We build our C-eSVMs by augmenting eSVMs with extra positive examples, which are selected from the dataset in a semi-supervised manner. In the end of this section we demonstrate how to apply C-eSVMs to new images. Due to the lack of space we omit description of the Exemplar SVM, please refer to Malisiewichz *et al*. [4] instead.

3.1. Collection of Clustered Exemplar SVMs

The intuition behind the Clustered eSVM is the following. We believe that one example is not enough to form a solid representation of a classifier; instead using a cluster of exemplars that are visually similar to each other will help to build a stronger classifier.

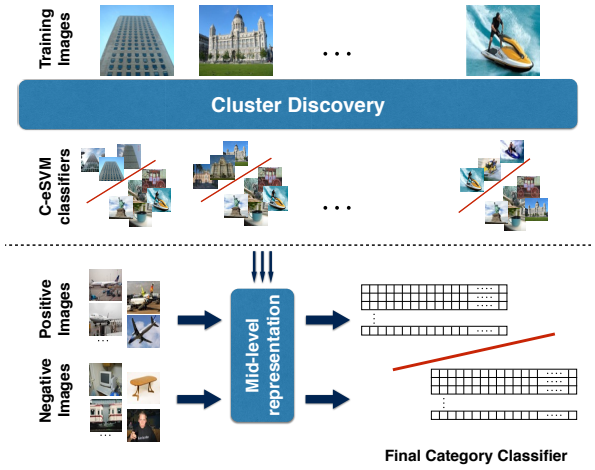


Fig. 1. **Overview:** First, we train a set of Clustered Exemplar SVMs for each exemplar in the dataset; second, we use the C-eSVMs to form a mid-level representation for each image and train a final binary classifier for each category.

Learning C-eSVMs: In contrast to eSVM, where a classifier is learnt only from one positive exemplar, we use *a cluster of examples* for each classifier. In other words, we use each exemplar as a seed to form a cluster of examples from the the same category, and then train a *separate classifier per each cluster*. This approach mimics sub-category detection while staying focused on the specific example from the training set.

Given a set of training images and their labels $\{(x_i, y_i)\}_{i=1}^{n}$ for each exemplar x_i we form a negative set N_i by taking all images from categories that are different from x_i, and a cluster P_i containing examples from the same category as x_i. We learn a SVM classifier for each positive cluster P_i:

$$\begin{aligned} &\min_{w_i, b_i} \|w_i\|^2 + C^+ \sum_{x_i \in P_i} \xi_i^+ + C^- \sum_{x_j \in N_i} \xi_j^- \\ &\text{s.t.} \quad w_i^T \phi(x_i) + b_i \geq 1 - \xi_i^+, \quad \forall x_i \in P_i, \ \ \xi_i^+ \geq 0 \\ &\quad -(w_i^T \phi(x_j) + b_i) \geq 1 - \xi_j^-, \quad \forall x_j \in N_i, \ \ \xi_j^- \geq 0 \end{aligned} \tag{1}$$

We call resulting classifier a *C-eSVM classifier*.

Note that even though we reuse negative data for training classifiers for clusters from the same category ($N_k = N_l$ iff $y_k = y_l$), the impact of negative data is different since different negative examples can be selected as support vectors.

Selecting positive clusters: We pose the problem of cluster discovery as the problem of semi-supervised learning. For each example (x_i, y_i) we form two sets: N_i and Q_i. As we previously mentioned, N_i is a set of negative examples. Meanwhile, Q_i is a set of so called "semipositive examples": examples of Q_i has the same category as (x_i, y_i), but we do not know which examples from Q_i belong to the same cluster as (x_i, y_i). This is a classical problem of SSL: we have a set of labeled examples (original exemplar (x_i, y_i) and negative examples N_i), and a set of unlabeled examples Q_i. The goal is to estimate labels of examples in Q_i. In our approach

Algorithm 1 Training Clustered Exemplar SVMs

Input : examples $\{(x_i, y_i)\}_{i=1}^n$, # of steps τ,
  # of confident examples M
Output : parameters $\mathbf{w} = \{w_1, \ldots, w_n\}$
for $i \leftarrow 1$ **to** n **do**
  Initialize $P_i^0 = \{\emptyset, x_i\}$; $Q_i^0 = \{x_q\}, \forall q$ s.t. $y_q = y_i$
  Compute $w_i^0 = \text{learnSVM}(P_i, N_i)$ using Eq. 1
  for $t \leftarrow 1$ **to** τ **do**
    Compute scores $s_q = {w_i^{t-1}}^T \phi(x_q), \quad \forall x_q \in Q_i^{t-1}$
    Select M examples $\{x_m^*\}_{m=1}^M$ with highest scores s_m
    Update Q_i: $Q_i^t = Q_i^{t-1} \backslash \{x_1^*, ..., x_M^*\}$
    Update P_i: $P_i^t = P_i^{t-1} \cup \{x_1^*, ..., x_M^*\}$
    Compute $w_i^t = \text{learnSVM}(P_i, N_i)$ using Eq. 1
  end for
end for
return $\mathbf{w} = \{w_1^\tau, ..., w_n^\tau\}$

we do not aim to find the best global clustering of all examples in the same category. Instead we solve a local problem: given x_i, we want to find a subset in Q_i of the most confident examples that belong to the same cluster as (x_i, y_i).

We solve this problem by using the self-training technique. In a nutshell, the idea of self-training is to build a classifier from the available labeled data, and then use the classifier to estimate labels of unlabeled examples. Note that other approaches from SSL could be used instead.

We use τ iterations to build a cluster P_i; this results in a set of intermediate clusters $\{P_i^0, ..., P_i^\tau\}$. Initially, given $P_i^0 = \{\emptyset, x_i\}$ and N_i, we first train eSVM. Next, we apply the eSVM classifier to all the examples from Q_i. The most confident examples $\{x_1, ..., x_M\}$ from Q_i are selected to form a new cluster $P_i^{t+1} = P_i^t \cup \{x_1, ..., x_M\}$; selected examples are then removed from Q_i. We repeat the procedure: a new C-eSVM classifier is trained for cluster P_i^t, semipositive set Q_i is reevaluated based on the scores from a new classifier, and the next cluster is formed. Self-training is terminated when $t = \tau$ and a final cluster $P_i = P_i^\tau$ is returned.

The algorithm for learning C-eSVMs and inference of cluster are presented in Alg. 1. We argue that a collection of C-eSVMs has two advantages. First, it will result in a model that naturally clusters all the images of one category in subcategories. Second, similar to eSVM, given a new example we are able to correspond it with specific example(s) from the training data.

3.2. Classification of Test Examples

We use mid-level features to classify test examples. Given an image, we apply all C-eSVM classifiers to it. The outputs of classifiers are then transformed into a feature vector, which corresponds to a new representation.We use this representation for all images from the training data. Then, we learn a final binary SVM classifier for each category. On the test stage, first we apply C-eSVM classifiers to get a mid-level representation, and then the final classifier is applied to infer the label of a test example.

	Bag-of-Words	eSVM	Ours, $M = 0.2$	Ours, $M = 1$	Ours, $M = 2$	Ours, $M = 5$
aPascal	49.6	48.9	**52.1**	50.0	50.4	51.4
aYahoo	72.4	72.4	**74.0**	71.8	72.5	73.3

Table 1. Mean average precision for aPascal and aYahoo, $\tau = 1$.

4. EXPERIMENTAL RESULTS

We analyze how a collection of C-eSVMs performs on the task of classification and how unseen examples correspond to examples from the training dataset. We first present the experimental setup, and then discuss results.

4.1. Experimental Setup

Datasets and parameter selection: We test our approach on two different datasets: aPascal [6] and aYahoo [6]. We use the same train/test split for the aYahoo as in [17]. Each object in the image is represented with a 9751-dimensional feature vector that contains BoW histograms on color, texture, visual words, and edges [6]. In addition, on all the datasets we use approximated kernels [18] for all our experiments (baseline, C-eSVMs and the final classifier).

We fix $C^- = 1$ and set $C^+ = N_{neg}/N_{pos}$ for learning C-eSVMs, where N_{pos} is number of examples in a cluster and N_{neg} is a number of negative examples. For the final classifier, we use 3-fold cross-validation to choose a C parameter for each category. We vary τ and M in our experiments.

Baselines and experiments: We conduct two baseline experiments to better evaluate our model. First, we learn a general SVM classifier original feature vectors. Second, we learn a collection of eSVMs and use them to produce mid-level representation and then train a final classifier. As for our model, we learn a collection of C-eSVMs for fixed $\tau = 1$ and different values of $M = \{0.2,\ 1,\ 2,\ 5\}$ and show performance of our model for different M. A value of $M = 0.2$ indicates that we add 20% of total positive examples to form a cluster. If $M = 5$, we simply add 5 examples at each step. In addition, we also evaluate our model for fixed $M = 5$ and different values of $\tau = \{1,\ 2,\ 3\}$.

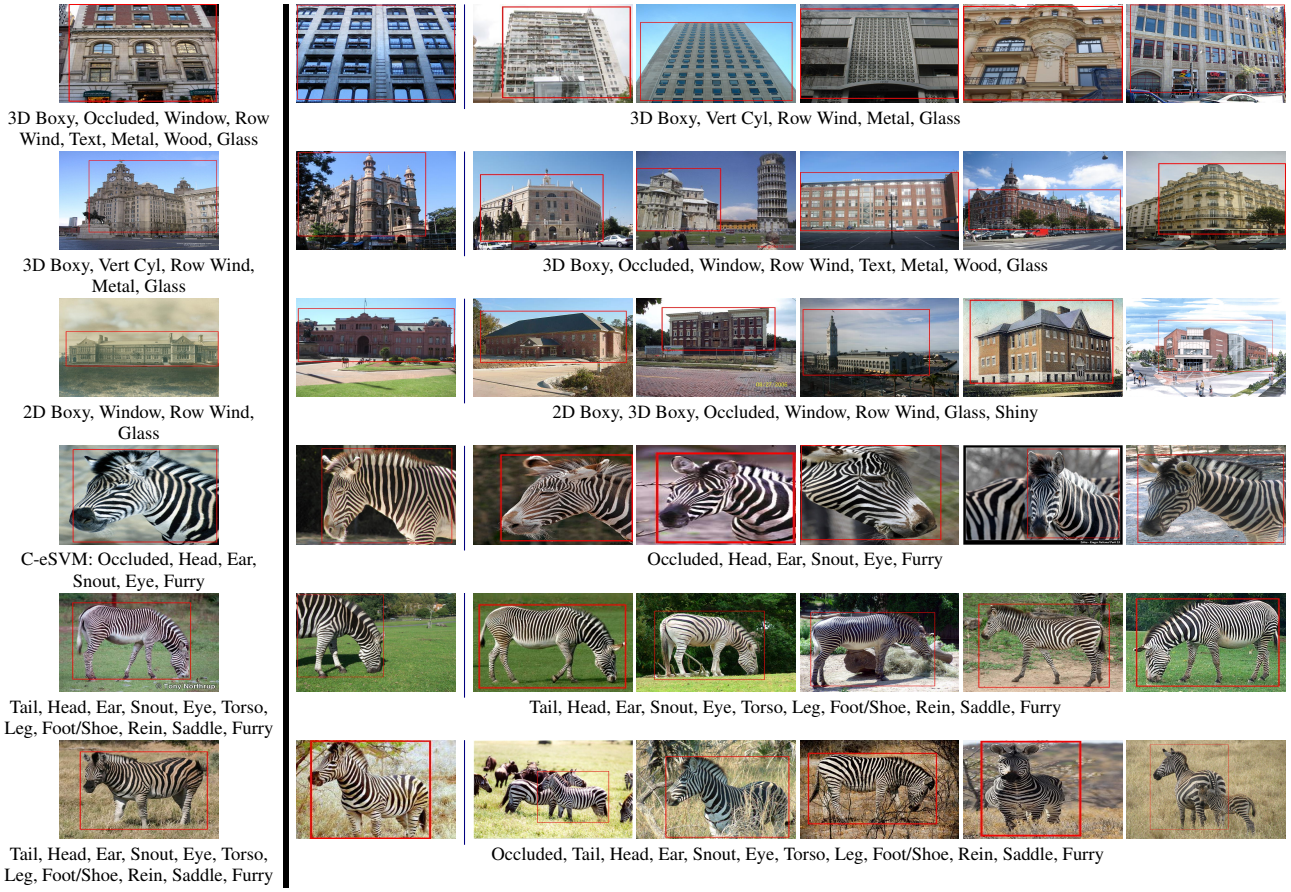
Data transfer: For each discovered cluster in the training data, we infer its attributes. When given a new test example with determined category label, we associate it with the cluster from the training set and transfer attributes.

4.2. Results Discussion

Classification performance: Classification results on the aPascal and aYahaoo datasets are presented in Table 1. For both datasets we can observe that creating larger clusters is beneficial, in particular adding 20% of positive examples helps improve classification performance for nearly all categories for all datasets. Furthermore, forming small clusters by adding only a few examples, e.g. 2 or 5 leads to a positive shift in performance. To better evaluate our approach, for the

Model	donkey	monkey	goat	wolf	jetski	zebra	centaur	mug	statue	building	bag	carriage	mAP
Ours, $\tau = 1$	52.8	62.2	**52.5**	**69.5**	**99.1**	**97.2**	24.5	92.9	63.5	**95.6**	**80.1**	90.0	73.3
Ours, $\tau = 2$	54.3	**63.9**	51.9	68.2	**99.1**	97.0	**24.7**	93.1	**64.0**	95.3	80.0	92.1	**73.6**
Ours, $\tau = 3$	**55.6**	63.5	50.1	66.4	99.0	96.6	23.0	**93.3**	63.8	95.0	78.7	**92.7**	73.1

Table 2. Classification results on the aYahoo dataset for different τ values, $M = 5$.



(a) Test query and groundtruth attributes

(b) Cluster from the training data and its attributes. Each cluster has two parts: the original examplar (on the left) and M added examples from the same category

Fig. 2. Test query and associated training examples on the aYahoo dataset. Rows 1-3: "building" category; rows 4-6: "zebra" category.

aYahoo dataset in Table 2 we also provide results for different τ while fixing $M = 5$.

Results demonstrate that different categories benefit from forming clusters of different sizes and using different numbers of iterations. We believe that careful selection of parameters M and τ will lead to a better performance of our model, which could be explored as future work.

Cluster Analysis: We use our model with $M = 5$ and $\tau = 1$ to analyze clusters and correspondence between test and training examples. Given a test example, we choose a C-eSVM and a corresponding cluster that leads to a maximum score. Visualization of clusters for the aYahoo is presented in Fig. 2. Each row correspond to a query test example, and associated cluster from the training data.

As we can observe, our model is capable to establish a meaningful correspondence between a test query and training examples and attributes from the cluster can be directly applied to the test image. In particular, in the aYahoo a zebra image was associated with other images based on pose and appearance; zebra clusters overall look coherent.

5. CONCLUSION

In this work we have introduced a novel collection of C-eSVM classifiers which incorporates automatic cluster discovery in a semi-supervised setting. The key feature of our method is that it is still exemplar oriented and allows direct association between test and train examples. The experimental results on two datasets demonstrate that Clustered Exemplar SVM creates meaningful clusters and has superior performance to the original Exemplar SVM.

6. REFERENCES

[1] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan, "Object detection with discriminatively trained part based models," *IEEE PAMI*, 2010.

[2] Chunhui Gu and Xiaofeng Ren, "Discriminative mixture-of-templates for viewpoint classification," in *ECCV*, 2010.

[3] Weilong Yang and George Toderici, "Discriminative tag learning on youtube videos with latent sub-tags," in *CVPR*, 2011.

[4] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros, "Ensemble of exemplar-SVMs for object detection and beyond," in *ICCV*, 2011.

[5] Minh Hoai and Andrew Zisserman, "Discriminative sub-categorization," in *CVPR*, 2013.

[6] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth, "Describing objects by their attributes," in *CVPR*, 2009.

[7] Li-Jia Li, Hao Su, Eric P. Xing, and Li Fei-Fei, "Object bank: A high-level image representation for scene classification and semantic feature sparsification," *NIPS*, 2010.

[8] Sreemanananth Sadanand and Jason J. Corso, "Action bank: A high-level representation of activity in video," in *CVPR*, 2012.

[9] Saurabh Singh, Abhinav Gupta, and Alexei A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *ECCV*, 2012.

[10] Xinggang Wang, Baoyuan Wang, Xiang Bai, Wenyu Liu, and Zhuowen Tu, "Max-margin multiple-instance dictionary learning," in *ICML*, 2013.

[11] Arpit Jain, Abhinav Gupta, Mikel Rodriguez, and Larry S. Davis, "Representing videos using mid-level discriminative patches," in *CVPR*, 2013.

[12] Xiaojin Zhu, "Semi-supervised learning literature survey," Tech. Rep., University of Wisconsin, Madison, 2008.

[13] Olivier Chapelle, Bernhard Schlkopf, and Alexander Zien, *Semi-Supervised Learning*, MIT Press, 2006.

[14] Devi Parikh and Kristen Grauman, "Interactively building a discriminative vocabulary of nameable attributes," in *CVPR*, 2011.

[15] Jonghyun Choi, Mohammad Rastegari, Ali Farhadi, and Larry S Davis, "Adding unlabeled samples to categories by learned attributes," in *CVPR*, 2013.

[16] Chao-Yeh Chen and Kristen Grauman, "Watching unlabeled video helps learn new human actions from very few labeled snapshots," in *CVPR*, 2013.

[17] Arash Vahdat and Greg Mori, "Handling uncertain tags in visual recognition," in *ICCV*, 2013.

[18] Andrea Vedaldi and Andrew Zisserman, "Efficient additive kernels via explicit feature maps," in *CVPR*, 2010.