Expanding Object Detector's HORIZON: Incremental Learning Framework for Object Detection in Videos

Alina Kuznetsova[1,3], Sung Ju Hwang[2], Bodo Rosenhahn[1], and Leonid Sigal[3]

[1]Leibniz University Hannover , [2]UNIST , [3]Disney Research Pittsburgh

Abstract

Over the last several years it has been shown that image-based object detectors are sensitive to the training data and often fail to generalize to examples that fall outside the original training sample domain (e.g., videos). A number of domain adaptation (DA) techniques have been proposed to address this problem. DA approaches are designed to adapt a fixed complexity model to the new (e.g., video) domain. We posit that unlabeled data should not only allow adaptation, but also improve (or at least maintain) performance on the original and other domains by dynamically adjusting model complexity and parameters. We call this notion domain expansion. *To this end, we develop a new scalable and accurate incremental object detection algorithm, based on several extensions of large-margin embedding (LME). Our detection model consists of an embedding space and multiple class prototypes in that embedding space, that represent object classes; distance to those prototypes allows us to reason about multi-class detection. By incrementally detecting object instances in video and adding confident detections into the model, we are able to dynamically adjust the complexity of the detector over time by instantiating new prototypes to span all domains the model has seen. We test performance of our approach by expanding an object detector trained on ImageNet to detect objects in egocentric videos of Activity Daily Living (ADL) dataset and challenging videos from YouTube Objects (YTO) dataset.*

1. Introduction

Over the past several years its been shown that there are significant biases among object detection datasets [19, 34], as well as between such datasets and the real world imagery. As a result, supervised classifiers/detectors trained on one dataset, often fail to work adequately on another, or real world images, statistics of which may have not been well captured in the original (labeled) training dataset. To handle such biases a number of supervised [1, 20, 27], semi-supervised and unsupervised [9, 12, 13, 16] domain adaptation methods have been proposed, for both classification [20, 27] and real-valued regression [39] tasks.

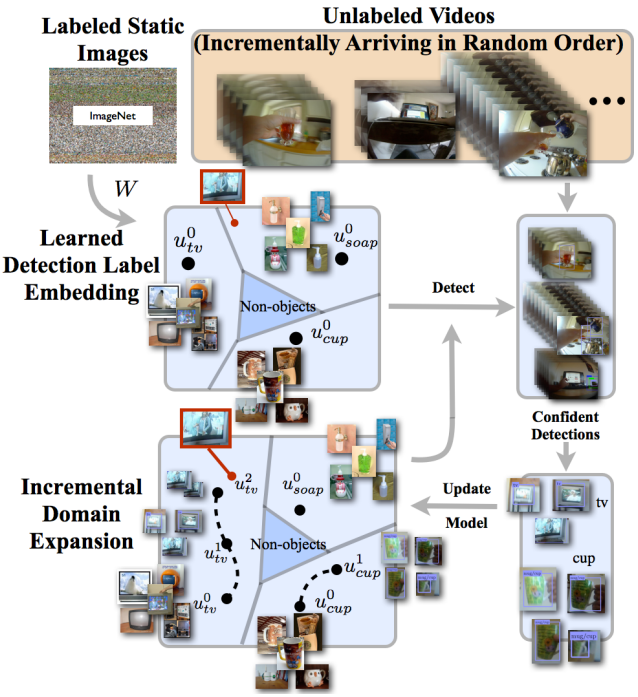

Figure 1. **Incremental Domain Expansion:** Illustration of the the overall proposed learning framework. First a large margin embedding (LME) detector is built based on labeled static images from ImageNet. As unlabeled videos arrive, detected objects are ranked based on detection confidence. Top ranked detections are expanded into tracks and used for new class prototype learning. Note that while a *TV* test sample (in red) may be too far in appearance from the original ImageNet trained model and hence misclassified, new prototypes, added based on tracks from videos, help to bridge the gap leading to correct classification.

While most domain adaptation techniques focus on applications where both training and test instances are images [20, 27], taken with conventional cameras, a few address

the problem in the context of image-to-video object detector adaptation [6, 11, 28, 29, 31]. Image-to-video scenario, is both compelling and challenging. It is highly desirable to utilize image datasets for training detectors to be used in videos, because images are easier to label and plenty of richly labeled datasets already exist. Obtaining a video equivalent of the ImageNet [5], in terms of scope, would be an insurmountable task. However, there are often significant appearance differences between images (*e.g.*, obtained on the web) and videos (*e.g.*, obtained on Youtube or using egocentric cameras). Web images tend to be of high resolution and are object-centric [5]. Videos, on the other hand, often come at lower resolution, are not object centric and, at least in egocentric setting, have a widely different appearance due to the quality of the sensor and motion artifacts. Hence domain shift between images and videos is often severe (*e.g.*, see results in [25]).

Nearly all domain adaptation techniques assume that data is separated into well defined discrete domains, most often a source (training) domain the the target (test) domain, and the task is to effectively transfer learned information (or labeled samples) from source to the target domain. This notion of discrete domains and focus on performance in only the target domain is somewhat of an oversimplification. In practice, as noted in [15], the target domain is often continuously evolving. Further, one can argue that as object instance, appearance, lighting and view are changing, the resulting evolution is actually an *expansion* of the original domain of this object, not formation of a new or evolution of the old domain. The difference is subtle. In continuous domain adaptation [15] (and incremental learning [28, 29]) the goal is to continuously adapt (or learn) a fixed complexity model to perform as accurately as possible on the arriving target batch of data. We argue for continuously adapting the complexity of the model itself. This should allow the adapted model to not only improve with respect to the arriving data, but also to at least retain its performance on the prior and future domains. We also do not assume that data arrives in a continuously evolving stream [15].

To this end we propose an incremental, self-paced inspired, approach to expanding the domain from images to unlabeled videos. We start from a large-margin embedding (LME) model [37], which we adapt to a detection task. Using this detection model, objects in the arriving unlabeled videos are found, and tracks associated with most confident instances are extracted (Figure 1 (right)). If instances from these tracks form a cluster, they are further used to adjust the complexity of the model by adding new class prototypes (Figure 1 (bottom left)). This process of extracting confident instances and learning expanded domain model, continues as additional videos arrive.

Our method is inspired by the overarching goals of lifelong learning [4, 30]. We note that our approach is related to sub-categorization, but unlike sub-categorization, which assumes fully labeled [14] or weakly-labeled instances [4], we work in an entirely unsupervised scenario. Further, while sub-categories, in general, do not form any sort of coherent structure in appearance space, our model ensures that object class prototypes form a coherent manifold, through regularization, limiting drift in learning.

Contributions: Our main contribution is the framework for *incremental domain expansion*, where complexity of an image-based object detection model is continuously adjusted to newly arriving unlabeled videos in a way that, over time, improves the performance on the evolving video domain but at the same time maintains (or improves) accuracy on the original image domain. Effectively, domain expansion, is about building a better overall detector using unsupervised video data. As part of this larger goal we formulate a new object detection model, inspired by the large-margin embedding (LME). We show how to extend the LME from multi-class object categorization to multi-class object detection problem, by introducing novel detection constraints to deal with the negative instances. We also propose a probabilistic formulation for LME, which allows the model to perform intuitive confidence evaluation for test instances and a novel multi-prototype LME formulation, that supports incremental learning.We show incremental domain expansion is effective in applying object detectors, trained with only ImageNet, to videos, improving performance by 48% (13% through expansion) with respect to original LME on ADL dataset[25] and by 15% on the YTO dataset [26].

2. Related Work

Domain adaptation from image to image domain: Our domain expansion method is closely related to domain adaptation (DA), which is a statistical method that focuses on the adaptation of an existing model in one domain (source) to a new data domain (target). The domain adaptation can be categorized into supervised methods [1, 20, 27], where labels are available for the samples in the target domain, and unsupervised methods, where no label is provided [9, 12, 13, 16, 31]. Our method relates to the latter case, as we aim to expand a model learned on labeled images to encompass unlabeled video data. The main difference between our method from the existing method is that, while the existing methods assume that there exist multiple discrete domains, we view all domains as related, as in [15], which models the source and target data on a single continuous manifold without clear distinction between the two. Based on this assumption, our model aims to improve on both source and target domains, while most method care only about the performance on a given target domain.

Adapting object detectors trained on images to videos: Among many DA tasks, the task of adapting detectors

trained on images to unlabeled video data is a topic of particular interest, largely due to the difficulty of video data annotation. Many models resort to a strategy that selects negative and positive samples from the test data based on their confidence with respect to the existing detector [36, 29, 31]. In [36], the baseline detector with low threshold generates positive/negative samples for the new vocabulary tree-based classifier that then decides on the label. Our model also leverages an existing model to select test samples, but in our case, the model is not fixed, but is allowed to expanded in complexity. We also aim to not only improve on the video domain, but also maintain, or improve, performance on the image domain. When deciding which detections to add to the training pool, many works further exploit the temporal continuity of frames in the video data [29, 31, 6], such as [29] which utilizes tracks, and leverage the matches between tracks and confident detections from a baseline detector as additional positive samples.

Perhaps closest works to ours are [31] and [6]. Tang *et al*. [31] proposed a self-paced method that incrementally adds positive samples, in the order of increase classification performance. Our self-paced learning algorithm is similar, but we expand the model instead of retraining it. Donahue *et al*. [6] proposed a method that incorporates an instance similarity graph to regularize the model, and applied it to the case of video data, where the distance of the instances within a track were utilized as the auxiliary instance similarity. Our method also leverages such similarities between entities, but it models group(video)-to-category similarity rather than instance-to-instance similarity, and the similarity graph is not given but is implicitly built from the order the videos arrive. Some works attempt the opposite of using weakly supervised YouTube videos to train image object detectors [26].

Self-paced learning: Our ranking of the unlabeled video samples based on their classification confidence, is related to self-paced learning [21], where the data points are presented in a meaningful order, which is often determined by the difficulty of classifying a given sample. In the original work of [21], self-paced learning was used to learn latent variables, and in [22], it was used to discover object categories from clustered image patches. Self-paced learning was also used in Tang *et al*. [31] to incrementally add in unlabeled samples into the labeled pool. Our work leverages a similar selection method, but our model considers multi-class case while [31] considers the single-class model.

Lifelong learning: The idea of lifelong learning, which is a continuos learning that transfers the knowledge learned at earlier learning stages to later stages, was first conceived in [32], and has become an active topic of research following the success of Never Ending Language Learner (NELL) [3]. NELL is an incremental model that learns about new concepts and rules by continuously observing textual input. Similar work has been proposed for the case of image data in [4]. Our work can be also viewed as an instance of lifelong learning, since the model is incrementally improved leveraging continuous stream of inputs, and the new subcategory prototypes are learned in the context of existing category prototypes.

Large-margin manifold embedding models for recognition: Our model builds on the large-margin (class) embedding (LME) [37, 2, 38], which aims to learn a low-dimensional space that is optimized for class discrimination. LME recently gained popularity, largely due to it's ability to scale to many class labels, which is becoming increasingly important with image classification becoming more focused on large-scale datasets. While there are many variants of LME, the one that is particularly relevant is [24], which presents a probabilistic multi-centroid model, that bares similarity to ours. However, the k-centroids for each class in [24] are obtained from k-means clustering on the original labeled samples, while in our model the multiple centroids are incrementally added as the model expands with new videos.

3. Incremental Learning Framework

We consider the general problem of applying an object detector, trained on images, for detecting objects in videos in completely unsupervised manner.

To formally state the problem, given a training image set $\mathcal{D}_{\mathcal{I}} = \{\mathbf{x}_i, y_i\}_{i=1}^{N_{\mathcal{I}}}$, such as ImageNet [5], that has $N_{\mathcal{I}}$ labeled instances, where $\mathbf{x}_i \in \mathbb{R}^D$ is a D-dimensional feature descriptor of an image patch containing an object and $y_i \in \{1, \ldots, C\}$ is the object label, we propose to first learn a large-margin embedding (LME)-based object detection model. The choice of proposing an embedding-based detection paradigm, over the more traditional SVMs or latent SVM, stems from flexibility and scalability of such models, their ability to generalize with little to no data [24], as well as their state-of-the-art performance on large-scale categorization tasks [10].

Once initial LME detection model is trained (Sections 3.1 and 4.1), we want to utilize unlabeled data from a sequence of arriving videos to incrementally improve the learned model. We propose an incremental learning framework that iteratively refines and adds complexity to the model as it is needed and consists of the following steps:

1. From each video we extract object proposals $\{\mathbf{b}_i\}_{i=1}^{N_v}$, using [35], and corresponding feature vectors $\{\mathbf{x}_i\}_{i=1}^{N_v}$.

2. We evaluate each $\mathbf{x}_i$ using proposed probabilistic multi-center LME model to obtain a set of detections, labels and corresponding confidences $\mathcal{D}_v = \{\mathbf{x}_i, c_i, p(y = c_i, d = 1|\mathbf{x}_i)\}_{i=1}^{N_v}$ (see Section 4.5)

3. We extend the set of detections by exploiting temporal consistency (see Section 4.5).

4. Finally, we update the model using selected samples, as described in Section 4.4.

This process continues while videos arrive. The framework is illustrated in Figure 1.

3.1. Background: Large Margin Embedding

Large-margin embedding [37] is a method for classification that projects samples into the low-dimensional space in a way that achieves separation among instances belonging to different classes, with respect to Euclidean metric.

As above, we denote the labeled training data[1] as $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$. The goal of LME is to learn a linear low-dimensional embedding defined by a projection matrix $\mathbf{W} \in \mathbb{R}^{d \times D}$ ($d \ll D$), together with class prototypes $\mathbf{u}_c \in \mathbb{R}^d, c = \{1 \dots C\}$, in the embedding space, such that a sample projected into this low dimensional space is closer to the correct class prototype than to all other prototypes.

Let us denote $d(\mathbf{z}_i, \mathbf{u}_c)$ as a similarity measure between a projected feature vector $\mathbf{z}_i = \mathbf{W}\mathbf{x}_i$ and a prototype $\mathbf{u}_c$. LME objective described above can be encoded by a positive margin between similarity of $\mathbf{z}_i$ and its true prototype and all the other prototypes:

$$d(\mathbf{z}_i, \mathbf{u}_{y_i}) + \xi_{ic} \geq d(\mathbf{z}_i, \mathbf{u}_c) + 1, \quad (1)$$
$$i = \{1 \dots N\}, c = \{1 \dots C\}, c \neq y_i,$$

where ξ_{ic} play the role of slack variables that we want to minimize. The learning of the optimal $\mathbf{W}$ and $\{\mathbf{u}_1, \dots, \mathbf{u}_c\}$ can be formulated as minimization of:

$$\sum_{i,c:c \neq y_i} \xi_{ic}^+ + \lambda \|\mathbf{W}\|_{FRO}^2 + \gamma \|\mathbf{U}\|_{FRO}^2, \quad (2)$$

where $\mathbf{U}$ us the columnwise concatenation of prototypes $\mathbf{u}_c$, ξ^+ is defined as $\max(\xi, 0)$ and λ and γ are weights of the regularizers. The label of a new sample $\mathbf{x}^*$ at the test time can then be determined by comparing the similarity of this new sample to prototypes in the embedding space:

$$y^* = \underset{c}{\operatorname{argmax}}\, d(\mathbf{z}^*, \mathbf{u_c}) = \underset{c}{\operatorname{argmax}}\, d(\mathbf{W}\mathbf{x}^*, \mathbf{u_c}). \quad (3)$$

In the initial formulation [37], L2-based similarity measure was used, however, we employ scalar product to measure similarity in the embedding space

$$d(\mathbf{z}_i, \mathbf{u}_c) = d_{\mathbf{W}}(\mathbf{x}_i, \mathbf{u}_c) = \langle \mathbf{W}\mathbf{x}_i, \mathbf{u}_c \rangle. \quad (4)$$

[1] We drop $\mathcal{I}$ subscript to avoid clutter.

4. Multi-prototype LME for object detection

The initial LME model is designed for *object classification* task. We extend the LME formulation to be applicable for *object detection* and provide corresponding probabilistic interpretation. We also derive multi-prototype formulation and present an algorithm for incremental learning.

4.1. LME model for object detection

The trivial way to extend the LME model for object detection is to assume existence of a *non-object* class. However, this would lead to modeling of this *non-object* class in LME using a *non-object* prototype. Since the variability in the appearance within the *non-object* class is much higher then within any other class, this may not be ideal.

Hence, instead, we define a patch as not containing an object of interest if it is sufficiently *dissimilar* to all known object class prototypes. This can be expressed as a set of additional large-margin constraints in the optimization:

$$d_{\mathbf{W}}(\mathbf{x}_j^0, \mathbf{u}_c) \leq 1 + \xi_{j0}, \quad c = \{1, \dots, C\}, \xi_{j0} \geq 0, , \quad (5)$$

that require the similarity to be low (distance to object prototypes high) for the non-object samples. Here $\mathbf{x}_j^0, \quad j = \{1, \dots, N_0\}$ are patches, that do not contain any object of the target classes, and ξ_{j0} are positive slack variables.

We note that for our specific similarity measure this actually pushes negative samples towards the center of the embedding space and effectively amounts to feature selection (or suppression) between all positive and a negative class; for other metrics, *e.g.*, a Euclidian metric, the geometric interpretation would be different.

The training objective is changed respectively to:

$$\sum_{i,c:c \neq y_i} \xi_{ic}^+ + \sum_j \xi_{j0}^+ + \lambda \|\mathbf{W}\|_{FRO}^2 + \gamma \|\mathbf{U}\|_{FRO}^2. \quad (6)$$

The prediction for a new feature vector $\mathbf{x}^*$ is formulated as follows:

$$y^* = \begin{cases} \operatorname{argmax}_c d_{\mathbf{W}}(\mathbf{x}^*, \mathbf{u_c}), & d_{\mathbf{W}}(\mathbf{x}^*, \mathbf{u_c}) \geq \tau, \\ c_0, \quad \forall c = 1, \dots, C: & d_{\mathbf{W}}(\mathbf{x}^*, \mathbf{u_c}) < \tau, \end{cases} \quad (7)$$

where τ is chosen based on the precision-recall trade-off, and c_0 denotes a non-object class.

Numerical optimization: Optimization in Eq. (6), with the corresponding constraints, is bi-convex in $\mathbf{W}$ and $\mathbf{U}$. We optimize Eq. (6) using alternating optimization, where we alternate between solving for $\mathbf{U}$ and $\mathbf{W}$ while keeping the other variable fixed, using stochastic gradient descent. The alternation process is repeated until the convergence criterion is met[2].

[2] $\|\mathbf{U} - \mathbf{U}_{prev}\|_2 + \|\mathbf{W} - \mathbf{W}_{prev}\|_2 \leq \epsilon$

4.2. Probabilistic LME interpretation

Estimation of confidence of the detector will be critical in ordering and selecting samples for domain expansion. In [31] authors use the value of the loss (or margin) as confidence. We, however, are dealing with a multi-class problem, so instead we derived the following probabilistic interpretation of the LME. We define the posterior probability of a sample that is considered to be a detection to belong to class c, by mapping the similarity between the projected instance and a class embedding to the range between 0 and 1 as follows:

$$p(y = c|d = 1, \mathbf{x}) = \frac{e^{d_{\mathbf{W}}(\mathbf{x},\mathbf{u}_c)/2\sigma^2}}{\sum_{i=1}^{C} e^{d_{\mathbf{W}}(\mathbf{x},\mathbf{u}_i)/2\sigma^2}}. \quad (8)$$

where $d = 1$ indicates that a sample is considered to be a detection.

In this setting, the probability of $\mathbf{x}$ being a detection can be formulated as follows:

$$p(d|\mathbf{x}) = \frac{1}{1 + e^{ad^m_{\mathbf{W}}(\mathbf{x})+b}}, \quad (9)$$

where $d^m_{\mathbf{W}}(\mathbf{x}) = \max_c d_{\mathbf{W}}(\mathbf{x}, \mathbf{u}_c)$, and a, b are parameters, serving the same purpose as τ in (7). Therefore, given a sample $\mathbf{x}^*$, the probability of detection of an instance of a class c is defined as:

$$p(y^* = c, d|\mathbf{x}^*) = p(y^* = c|d, \mathbf{x}^*)p(d|\mathbf{x}^*), \quad (10)$$

that is interpreted as a detection confidence for a class c.

4.3. Multi-prototype LME

Domain shift is accompanied by change in feature distribution in the original space and consequently, in the low-dimensional embedding space. This shift causes the performance decrease of a detector and to cope with such domain shift, we need a more flexible class representation in the embedding space. Following the work of [24], we learn several, K_c, prototypes for each class c to represent multimodal feature distribution across domains: $\mathbf{U}_c = [\mathbf{u}_c^1, \ldots, \mathbf{u}_c^{K_c}]$. Then, the similarity score between an instance and a class can be computed using the similarities to the different prototypes of the same class:

$$\tilde{d}_{\mathbf{W}}(\mathbf{x}_i, \mathbf{U}_c) = f(d_{\mathbf{W}}(\mathbf{x}_i, \mathbf{u}_c^1), \ldots, d_{\mathbf{W}}(\mathbf{x}_i, \mathbf{u}_c^{K_c})). \quad (11)$$

Different choices exist for the function $f(\cdot)$. However, in spirit of LME, $f(\cdot) = \max_k d_{\mathbf{W}}(\mathbf{x}_i, \mathbf{u}_c^k)$, seems like an appropriate choice.

We further replace $\max(\cdot)$ function by its smooth approximation $S^\alpha_{\mathbf{W}}(\mathbf{x}_i, \mathbf{U}_c)$ to simplify the numerical optimization of Eq. (5)-(6):

$$S^\alpha_{\mathbf{W}}(\mathbf{x}_i, \mathbf{U}_c) = \frac{\sum_{k=1}^{K_c} d_{\mathbf{W}}(\mathbf{x}_i, \mathbf{u}_c^k) e^{\alpha d_{\mathbf{W}}(\mathbf{x}_i,\mathbf{u}_c^k)}}{\sum_{j=1}^{K_c} e^{\alpha d_{\mathbf{W}}(\mathbf{x}_i,\mathbf{u}_c^j)}}, \quad (12)$$

where the greater the parameter α, the better the function approximates $\max(\cdot)$. The optimization problem for multi-prototype model can be formulated in the same manner as the LME model with detection constraints in Eq.(5)-(6) by replacing $d_{\mathbf{W}}(\mathbf{x}, \mathbf{u}_c)$ with $S_\alpha(\mathbf{x}_i, \mathbf{U}_c)$.

4.4. Incremental multi-prototype LME model expansion

The multi-prototype LME model is naturally suitable for domain expansion. As model encounters new data, which is not well approximated by the current prototypes, we can add new prototypes incrementally to more precisely model the feature distribution in the embedding space. The problem of learning a new prototype then can be formulated within LME framework as the following incremental learning procedure.

Suppose we want to expand the prototype-based representation for the class c_n. When adding a new prototype $\mathbf{u}_{c_n}$ to the model it should satisfy two properties: (i) the new prototype should be representative and discriminative for its class; (ii) it should not cause misclassification of samples from other classes, *i.e.*, it should be sufficiently far from existing category prototypes for other classes. More formally, the optimization problem can be formulated as follows:

minimize:

$$\sum_{\substack{i,c: y_i = c_n, \\ c \neq c_n}} \xi^+_{ic} + \sum_{i: y_i \neq c_n} \zeta^+_i + \sum_j \xi^+_{j0} + \nu\|\mathbf{u}_{c_n} - \mathbf{u}_0\|^2 + \eta\|\mathbf{W} - \mathbf{W}_0\|^2, \quad (13)$$

subject to:

$$S^\alpha_{\mathbf{W}}(\mathbf{x}_i, \tilde{\mathbf{U}}_{c_n}) + \xi_{ic} \geq S^\alpha_{\mathbf{W}}(\mathbf{x}_i, \mathbf{U}_c) + 1, y_i = c_n \quad (14)$$

$$S^\alpha_{\mathbf{W}}(\mathbf{x}_i, \mathbf{U}_{y_i}) + \zeta_i \geq S^\alpha_{\mathbf{W}}(\mathbf{x}_i, \tilde{\mathbf{U}}_{c_n}) + 1, y_i \neq c_n \quad (15)$$

$$S^\alpha_{\mathbf{W}}(\mathbf{x}^0_j, \tilde{\mathbf{U}}_{c_n}) \leq 1 + \xi_{j0}, \quad (16)$$

where $\mathbf{W}$ is a newly learned data embedding, $\mathbf{W}_0$ is the existing data embedding, $\mathbf{u}_0$ is the original prototype for the given category, and $\tilde{\mathbf{U}}_{c_n} = [\mathbf{U}_{c_n}, \mathbf{u}_{c_n}]$. Eq. (14) is a sofmax LME constraint between the new category and the existing categories, Eq. (15) is the same constraint between each of the existing categories to the new category embedding, and Eq. (16) is the detection constraints. The parameters ν and η are the regularization weights[3] which determine how similar the newly learned embeddings should be to the original category and data embeddings. Optimization problem in Eq (13)-(16) is non-convex, but provided with a good initialization stochastic gradient descent allows to obtain reasonable local minima.

The incremental update is especially beneficial, when not all data is available and the newly arriving data has different, or evolving, feature distribution. However, to apply

[3] In practice, we set η to a high number to prevent model drift.

the derived model, the remaining core question is how to select the samples from unlabeled videos for the incremental model update; we address this in the next section.

4.5. Discovering objects from unlabeled video

Initial detection set extraction: Given an unlabeled video, we first extract the initial set of detections by computing object proposals $\{\mathbf{b}_i\}_{i=1}^{N_v}$ and their features $\{\mathbf{x}_i\}_{i=1}^{N_v}$, using off-the-shelf proposal method [35]. Then we extract visual feature $\mathbf{x}_i$ for each object proposal i and evaluate them using the multi-prototype model to obtain probability score for each proposal. Then a set of detected objects $\mathcal{D}_v = \{\mathbf{x}_i, c_i, p(y = c_i, d = 1|\mathbf{x}_i)\}_{i=1}^{D_v}$ could be formed by selecting the object proposal i, s.t. $p(y = c_i, d = 1|\mathbf{x}_i) > \nu$, where ν is some threshold; $\nu = 0.6$ allowed us to obtain fairly good results in our experiments. The obtained set of detection $\mathcal{D}_v$ can be then used as new positive training samples to train the new category prototype.

Tracks formation: To obtain more samples, we further exploit the temporal consistency, *i.e.*, if object is detected in one frame, it is likely to persist for a number of frames at relatively similar position and scale.

Specifically, we employ idea proposed in [31] and extract tracks from a video using the KLT tracker [33, 23]. After computing a set of confident object proposals $\mathcal{D}_v$ with the corresponding bounding boxes $\{b_i\}_{i=1}^{|D_v|}$, for each object proposal bounding box b_i, we select the longest track t_i that intersects it. We then compute the relative positions of the object proposals that intersect this track t_i across frames, and at each frame select the proposal that has the highest PASCAL overlap[4] with b_i swept across the track. In this way we obtain a set of object proposals for each b_i, which constitute a track. To obtain track score, we evaluate them and accept track if more then half of the detections on the track have $p(y^* = c, d = 1|\mathbf{x}) > \nu$. If a track is accepted, we add all the samples from the track to $\mathcal{D}_v$.

5. Experiments

We validate our method on real-world image and video datasets. For image dataset, to train the base detector, we use the subsets of the ImageNet [5] dataset (ca. 600 images per class) with the corresponding classes. We use a disjoint subset of ImageNet [5] (ca. 400 images per class) to report detector performance on images before and after incremental domain expansion where appropriate.

We test our method on the Activities of Daily Living (ADL) [25] and YouTube Objects (YTO) [26] datasets[5]:

ADL Dataset: The ADL [25] dataset contains 20 first-person videos, recorded by different subjects. This is a challenging dataset and straightforward application of the detector learned on static images does not work well [25], since the objects suffer from large viewpoint/scale variations and occlusions due to interactions. Each video in the ADL dataset has bounding box annotations for objects from 48 classes. We select a subset of 8 most frequently encountered classes, namely *bottle*, *fridge*, *microwave*, *mug*, *oven*, *soap liquid*, *tap*, and *tv*, to test our model.

YTO Dataset: The YTO [26] dataset consists of the collection of internet videos, each video containing a single object out of 10 classes: *aeroplane*, *bird*, *boat*, *car*, *cat*, *cow*, *dog*, *horse*, *motorbike*, *train*. The dataset is divided into train and test parts, where the test portion contains a single frame with bounding box annotation of the target object per video. For the evaluation we used the test part of the dataset only; it contains $15 - 60$ videos for each class.

We use the following methods as baselines:

DPM: Performance reported in the papers [25, 18] obtained using Deformable Part Model [8].

GK: An approach of [12] for unsupervised domain adaptation. We first select samples from $\mathcal{D}_v$ and use them for learning feature transformation between the source and the target domains. We then use the learned mapping to reproject all features from the target (video) domain to the source (image) domain and perform detection in the source domain.

LME: A baseline LME model that formulates a prediction using Eq. (7).

LME-A: A baseline doman adaptation (DA) approach that adapts to video domain in a batch (without increasing model complexity): we select confident samples, as described in Section 4.5, and re-train our model using baseline LME detector as initialization.

To show the performance gain obtainable by each step of our algorithm, we implement the following variants:

LME-D: LME with the detection constraints in Eq. (5).

LME-DT: LME with the detection constraints and exploiting temporal consistency by using tracks.

IDE-LME: Our full probabilistic multi-centroid LME model with detection constraints, that is incrementally expanded with the unlabeled data.

We use Caffe features [17], which are deep image representations obtained at layer *fc7* of a convolutional neural network, for all LME baselines and our variants.

5.1. Quantitative Evaluation

We evaluate object detection performance of the baselines and our models using mean average precision (mAP) [7] on ADL (Table 1) and YTO (Table 2) datasets. For both datasets we observe that while the baseline LME model

[4] $overlap(b_1, b_2) = \frac{area(b_1 \cap b_2)}{area(b_1 \cup b_2)}$

[5] Note that our domain expansion method is entirely unsupervised and the annotations on the target datasets are only used for evaluation.

	bottle	fridge	microwave	mug/cup	oven/stove	soap liquid	tap	tv	av. ADL	ImageNet
DPM[25]	**9.8**	0.4	20.2	14.8	0.1	**2.5**	0.1	26.9	9.35	–
GK [12]	2.11	1.77	41.19	14.70	19.57	0.20	1.62	**60.67**	17.73	–
LME	0.00	0.28	3.07	0.00	0.52	0.03	0.55	3.73	1.02	–
LME-A	1.93	**3.42**	40.30	18.34	27.84	0.37	1.46	53.26	18.36	76.96
LME-D	1.69	1.63	39.87	13.06	19.33	0.35	1.67	40.64	14.78	78.91
LME-DT	1.85	1.76	52.37	15.91	24.54	0.42	**2.41**	56.16	19.43	–
IDE-LME	2.04	2.73	**56.69**	**21.86**	**29.94**	0.25	2.26	59.53	**21.91**	**79.23**

Table 1. Detection performance for each class and all categories averaged by mAP on the ADL dataset, for the baselines and our method's variants. We also report the detection results on the ImageNet subset containing the 8 classes from the ADL dataset

Figure 2. Illustration of data and detection results. Top row: example of images for ImageNet, used for training of the initial model. Bottom row: instances of detections that were correctly detected using IDE-LME. Notice the significant differences in how objects appear in ImageNet and ADL dataset.

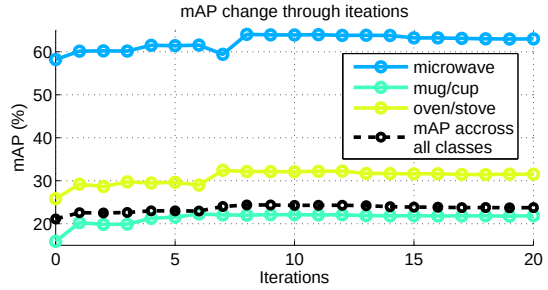

Figure 3. mAP as a function of videos seen (x-axis) for subset of classes in ADL dataset [25]; the mAP is average across videos used for expansion and the rest of the videos in the ADL dataset; the increase of mAP illustrates, that as the model gains complexity, the performance improves also on unseen videos.

trained on the images without detection constraints performs very poorly, adding detection constraints results in performance on par with DPM baselines. Incorporating temporal consistency using tracks (LME-DT) improves the performance by over 31% with respect to LME-D for ADL dataset (5% for YTO dataset). Incrementally updating the model (IDE-LME) using our approach brings further significant performance improvement of 13% (9% for YTO dataset) in comparison with LME-DT, leading to overall 48% and 15% improvement over LME-D on ADL and YTO dataset respectively. The smaller performance gain on YTO dataset can be attributed to the fact that for YTO dataset, feature distribution is similar to that of images as each video contains one or few objects in typical viewpoints; another reason is sparse annotations of the YTO dataset, that limit the ability to estimate the performance improvement.

Note that the other baseline, GK, outperforms IDE-LME on the classes with high initial precision (e.g. *tv* and *microwave* for ADL dataset), while performs significantly worse on the other classes. We believe that such classes effectively determine GK transformation, while the change in the distribution of other classes is not taken into account. Another trend seen on both datasets is that the initial model should have enough precision to be able to select samples from the videos for the update to work effectively, otherwise a slight performance drop can occur (*soap liquid* class in ADL dataset or *cat* class in YTO dataset).

Above experiments suggest that increased complexity of the model captures previously unseen variations in the object class appearance. To support this claim and to show that our model also improve on the original image domain, we report classification results on the test split of ImageNet dataset. In Table 1 and 2 we observe small but positive gains on the ImageNet, over LME-D. This suggests that newly added samples do not only improve the detection performance for the test video data, but also improve the classification performance on the source image data. Note that our domain adaptation (DA) baseline LME-A improves on videos but degrades on source image domain (a typical behavior for DA), on both datasets.

The performance of the model generally increases with more observed videos, but asymptotes after first 10 iterations for ADL dataset (see Figure 3). This early performance saturation might be due to high appearance similarity among objects in the target domain (egocentric videos). YTO dataset shows a similar trend. However, if target domain constantly changes or evolves over time, the performance might continue to increase.

	aero	bird	boat	car	cat	cow	dog	horse	mbike	train	av. YTO	ImageNet
DPM[18]	30.79	10.46	0.97	48.62	18.30	33.69	13.67	**26.78**	35.85	**23.98**	24.31	–
GK[12]	**40.05**	23.16	24.44	32.62	**24.26**	38.26	24.23	17.75	**36.27**	10.69	27.17	–
LME	35.00	24.13	16.08	27.41	4.30	31.18	2.12	0.23	6.83	10.30	15.75	–
LME-A	39.78	**35.18**	35.20	**48.67**	15.02	37.90	**30.70**	25.86	28.93	10.82	**30.80**	79.91
LME-D	29.61	22.91	32.39	25.53	18.63	38.94	15.55	9.22	31.47	12.09	23.63	83.16
LME-DT	31.67	21.83	40.13	25.94	17.59	41.44	15.47	11.74	30.56	13.67	25.00	–
IDE-LME	33.07	21.40	**42.26**	34.49	18.33	**46.92**	17.24	11.83	34.73	12.50	**27.28**	**83.20**

Table 2. Detection performance (mAP) for each class and mean mAP across all classes on the YouTube Objects (YTO) dataset.

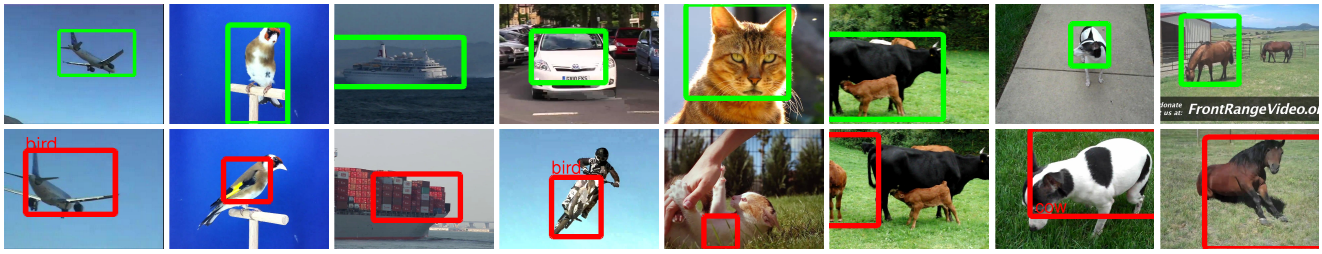

Figure 4. Top row: examples of the correctly detected objects. Bottom row: examples of the incorrect detections (from left to right), due to incorrect classification, inaccurate bounding boxes, or incorrect labels.

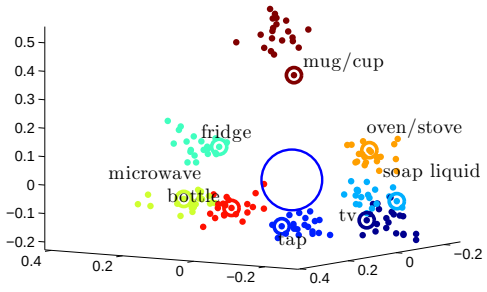

Figure 5. The visualization of the learned (expanded) multi-center LME, on the ADL dataset [25], projected into the 3D space; a group of prototypes of the same color represents a class; the initial prototypes are marked with additional circle around them.

5.2. Qualitative Analysis

Figure 2 show examples detections on the ADL dataset. Notice the significant difference between the source domain and the target domain. Figure 5 is the 3D visualization of the learned 8-dimensional embedding, where each category is represented as a set (manifold) of category prototypes which were expanded over the learning process. We observe that for some object classes, such as *mug*, the later added prototypes are placed far from the original center, that represents feature distribution change between the *mug* class in the ImageNet dataset and in the ADL dataset.

Figure 4 shows the detection examples on the YTO dataset, obtained using IDE-LME. Note that object is often identified correctly, but the bounding box is either too small or too large. We attribute this to the fact that background comprises large portion of ImageNet images, which might rank loose detections higher than tight ones.

6. Conclusion

In this paper, we have tackled the problem of domain expansion, where the scope of the object detector learned on the initial image labeled training set is incrementally expanded to cover incoming unlabeled videos. To this end, we have developed a novel online probabilistic multi-center large margin embedding model with detection constraints, where each object category is represented with multiple prototypes, which incrementally increase in number as self-paced learning algorithm selects confident samples from the incoming unlabeled data to add. Experimental validations on the ADL and YTO public datasets shows that the proposed model significantly improves the detection performance not only on the target unlabeled videos, but also on the source image domain. Our incremental domain expansion model could serve as a lifelong learning system for object detection—as the model expands to encompass continuous stream of unlabeled new video data. One potential problem that might arise is *model drift*. We have not seen this in our experiments and our regularization is designed to prevent this, but it is possible that such drift may arise with much larger scale datasets. As future work, we plan to explore a human-in-the loop system with active learning to prevent such drift, such that model can essentially self-train itself, with infrequent human intervention only triggered by the model's request.

References

[1] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *ICCV*, 2011.

[2] S. Bengio, J. Weston, and D. Grangier. Label embedding trees for large multi-class task. In *NIPS*, 2010.

[3] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell. Toward an architecture for never-ending language learning. In *AAAI*, 2010.

[4] X. Chen, A. Shrivastava, and A. Gupta. Neil: Extracting visual knowledge from web data. In *ICCV*, 2013.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. F.-F. ei. Imagenet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[6] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell. Semi-supervised domain adaptation with instance constraints. In *CVPR*, 2013.

[7] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, June 2010.

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

[9] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, 2013.

[10] Y. Fu, T. Hospedales, T. Xiang, Z. Fu, and S. Gong. Transductive multi-view embedding for zero-shot recognition and annotation. In *ECCV*, 2014.

[11] A. Gaidon, G. Zen, and J. A. Rodriguez-Serrano. Self-learning camera: Autonomous adaptation of object detectors to unlabeled video streams. In *Arxiv*, 2014.

[12] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, 2012.

[13] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, 2011.

[14] M. Hoai and A. Zisserman. Discriminative sub-categorization. In *CVPR*, 2013.

[15] J. Hoffman, T. Darrell, and K. Saenko. Continuous manifold based adaptation for evolving visual domains. In *CVPR*, 2014.

[16] J. Hoffman, B. Kulis, T. Darrell, and K. Saenko. Discovering latent domains for multisource domain adaptation. In *ECCV*, 2012.

[17] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *Arxiv*, 2014.

[18] V. Kalogeiton, V. Ferrari, and C. Schmid. Analysing domain shift factors between videos and images for object detection. *Arxiv*, 2015.

[19] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012.

[20] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, 2011.

[21] M. P. Kumar, B. Packer, and D. Koller. Self-paced learning for latent variable models. In *NIPS*, 2010.

[22] Y. J. Lee and K. Grauman. Learning the easy things first: Self-paced visual category discovery. In *CVPR*, 2011.

[23] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1981.

[24] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *TPAMI*, 2013.

[25] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *CVPR*, 2012.

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

[27] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

[28] P. Sharma, C. Huang, and R. Nevatia. Unsupervised incremental learning for improved object detection in a video. In *CVPR*, 2012.

[29] P. Sharma and R. Nevatia. Efficient detector adaptation for object detection in video. In *CVPR*, 2013.

[30] D. L. Silver, Q. Yang, and L. Li. Lifelong machine learning systems: Beyond learning algorithms. In *AAAI*, 2013.

[31] K. Tang, V. Ramanathan, L. Fei-Fei, and D. Koller. Shifting weights: Adapting object detectors from image to video. In *NIPS*, 2012.

[32] S. Thrun. A lifelong learning perspective for mobile robot control. In *Intelligent Robots and Systems*. 1995.

[33] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical report, 1991.

[34] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011.

[35] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *IJCV*, 2013.

[36] X. Wang, G. Hua, and T. Han. Detection by detections: Non-parametric detector adaptation for a video. In *CVPR*, June 2012.

[37] K. Q. Weinberger and O. Chapelle. Large margin taxonomy embedding for document categorization. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *NIPS*. 2009.

[38] J. Weston, S. Bengio, and N. Usunier. Wsabie: Scaling up to large vocabulary image annotation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.

[39] M. Yamada, Y. Chang, and L. Sigal. Domain adaptation for structured regression,. In *IJCV*, 2014.