Revisiting knowledge transfer for training object class detectors

Jasper R. R. Uijlings
jrru@google.com

S. Popov
spopov@google.com

V. Ferrari
vittoferrari@google.com

Google AI Perception

Abstract

We propose to revisit knowledge transfer for training object detectors on target classes from weakly supervised training images, helped by a set of source classes with bounding-box annotations. We present a unified knowledge transfer framework based on training a single neural network multi-class object detector over all source classes, organized in a semantic hierarchy. This generates proposals with scores at multiple levels in the hierarchy, which we use to explore knowledge transfer over a broad range of generality, ranging from class-specific (bycicle to motorbike) to class-generic (objectness to any class). Experiments on the 200 object classes in the ILSVRC 2013 detection dataset show that our technique (1) leads to much better performance on the target classes (70.3% CorLoc, 36.9% mAP) than a weakly supervised baseline which uses manually engineered objectness [11] (50.5% CorLoc, 25.4% mAP). (2) delivers target object detectors reaching 80% of the mAP of their fully supervised counterparts. (3) outperforms the best reported transfer learning results on this dataset (+41% CorLoc and +3% mAP over [18, 46], +16.2% mAP over [32]). Moreover, we also carry out several across-dataset knowledge transfer experiments [27, 24, 35] and find that (4) our technique outperforms the weakly supervised baseline in all dataset pairs by $1.5\times - 1.9\times$*, establishing its general applicability.*

1. Introduction

Recent advances such as [17, 28, 33, 50] have resulted in reliable object class detectors, which predict both the class label and the location of objects in an image. Typically, detectors are trained under full supervision, which requires manually drawing object bounding-boxes in a large number of training images. This is tedious and very time-consuming. Therefore, several research efforts have been devoted to training object detectors under weak supervision, i.e. using only image-level labels [2, 4, 7, 8, 21, 29, 36, 40, 41, 42, 48]. While this is substantially cheaper, the resulting detectors typically perform considerably worse than their fully supervised counterparts.

In recent years a few large datasets such as ImageNet [35] and COCO [27] have appeared, which provide many bounding-box annotations for a wide variety of classes. Since many classes share visual characteristics, we can leverage these annotations when learning a new class. In this paper we propose a technique for training object detectors in a knowledge transfer setting [15, 18, 32, 34, 38, 46]: we want to train object detectors for a set of *target classes* with only image-level labels, helped by a set of *source classes* with bounding-box annotations. We build on Multiple Instance Learning (MIL), a popular framework for weakly supervised object localization [29, 8, 2, 7, 10, 40, 42, 36], and extend it to incorporate knowledge from the source classes. In standard MIL, images are decomposed into object proposals [1, 47, 11] and the process iteratively alternates between re-localizing objects given the current detector, and re-training the detector given the current object locations. During re-localization, typically the highest-scoring proposal for an object class is selected in each image containing it.

Several weakly supervised object localization techniques [8, 7, 31, 37, 40, 38, 45, 49, 4] incorporate a *class-generic* objectness measure [1, 11] during the re-localization stage, to steer the selection towards objects and away from backgrounds. These works use a manually engineered objectness measure and report an improvement of around 5% correct localizations. As [9] argued, using objectness can be seen as a (weak) form of knowledge transfer, from a generic object appearance prior to the particular target class at hand.

On the opposite end of the spectrum, several works perform *class-specific* transfer [15, 34, 18, 46], For each target class, they determine a few most related source classes to transfer from. Some works [15, 34] use the appearance models of the source classes to guide the localization of the target class by scoring proposals with it, similar to the way objectness is used above. Other works [18, 46] instead perform transfer directly on the parameters of a neural network. They first train a neural network for whole-image classification on all source and target classes, then fine-tune the

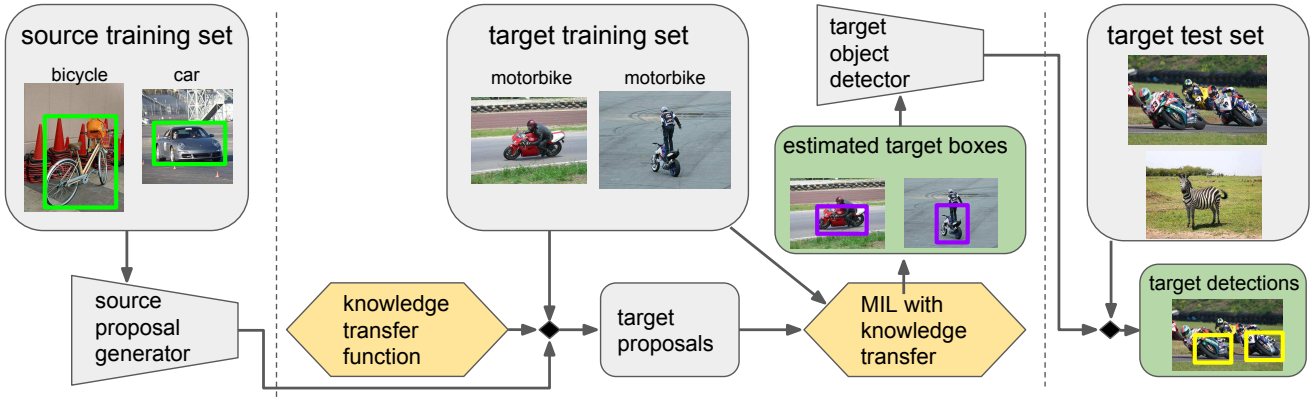

Figure 1. Illustration of our settings and framework. The source training set is annotated with bounding-boxes. We use this set to train a proposal generator, and then apply it to the target training set, where it produces proposals with scores at different levels of generality, ranging from class-specific to class-generic. We use these to perform MIL with knowledge transfer (Sec. 2.2) on the target training set, using only image-level labels (no bounding-boxes). MIL produces boxes for the target classes, which we use to train an object detector. Finally, we apply the object detector to the target test set (using no labels at all). The performance of our framework is measured both on the target training set (Sec. 3.1) and the target test set (Sec. 3.2).

source classifiers into object detectors, and finally transfer the parameter transformation between whole-image classifiers and object detectors from the source to related target classes, effectively turning them into detectors too.

Finally, YOLOv2 [32] jointly trains the source and target class detectors by combining a standard fully supervised loss with a weakly supervised loss (i.e. the highest scored box is considered to be the target class). During training they use hierarchical classification [5, 23, 39], which implicitly achieves knowledge transfer somewhere in-between class-generic and class-specific.

In this paper we propose a unified knowledge transfer framework for weakly supervised object localization which enables us to explore the complete range of semantic specificity (Fig. 1). We train a single neural network multi-class object detector [28] over all source classes, organized in a semantic hierarchy [35]. This naturally provides high-quality proposals and proposal scoring functions at multiple levels in the hierarchy, which we use during MIL on the target classes. The top-level scoring function for 'entity' conceptually corresponds to the objectness measure [1, 11], but it is stronger, as provided by a neural network properly trained on thousands of images. Compared to previous works using objectness [8, 7, 31, 37, 40, 38, 45, 49, 4], this leads to much larger performance improvements on the target classes. Compared to other transfer works [15, 32, 34, 18, 46], our framework enables to explore a broad range of generality of transfer: from the class-generic 'entity' class, from intermediate-level categories such as 'animal' and 'vehicle', and from specific classes such as 'tiger' and 'car'. We achieve this in a simple unified framework, using a single SSD model as source knowledge, where we select the proposal scoring function to be used depending on the target class and the desired level of generality of transfer.

Through experiments on the 200 object classes in the ILSVRC 2013 detection dataset, we demonstrate that: (1) knowledge transfer at any level of generality substantially improve results, with class-generic transfer working best. This is excellent news for practitioners, as they can get strong improvements with a relatively simple modification to standard MIL pipelines. (2) our class-generic knowledge transfer leads to large improvements over a weakly supervised object localization baseline using manually engineered objectness [11]: 70.3% CorLoc vs 50.5% CorLoc on the target training set, 36.9% mAP vs 25.4% mAP on the target test set. (3) our method delivers detectors for the target classes which reach 80% of the mAP of their fully supervised counterparts, trained from manually drawn bounding-boxes. (4) we outperform the best reported transfer learning results on this dataset: +41% CorLoc and +3% mAP over [18, 46], +16.2% mAP over [32]. Moreover, we also carry out several across-dataset [27, 24, 35] knowledge transfer experiments and find that (5) our technique outperforms the weakly supervised baseline in all dataset pairs by a factor $1.5\times - 1.9\times$, establishing its general applicability.

2. Method

We now present our technique for training object detectors in a knowledge transfer setting (Fig. 1). In this setting we have a training set $\mathcal{T}$ of *target classes* with only image-level labels, and a training set $\mathcal{S}$ of *source classes* with bounding-box annotations. The goal is to train object detectors for the target classes, helped by knowledge from the source classes.

We start in Section 2.1 by introducing a reference Multiple Instance Learning (MIL) framework, typically used in weakly supervised object localization (WSOL), i.e. when given only the target set $\mathcal{T}$. In Section 2.2 we then explain how we incorporate knowledge from the source classes $\mathcal{S}$ into this framework. Finally, Section 2.3 discusses the broad range of levels of transfer that we explore.

2.1. Reference Multiple Instance Learning (MIL)

General scheme. For simplicity, we explain here MIL for one target class $t \in \mathcal{T}$. The process can be repeated for each target class in turn. The input is a training set $\mathcal{I}$ with positive images, which contain the target class, and negative images, which do not. Each image is represented as a bag of object proposals $\mathcal{B}$ extracted by a generator such as [1, 11, 47]. A negative image contains only negative proposals, while a positive image contains at least one positive proposal, mixed in with a majority of negative ones. The goals are to find the true positive proposals and to learn an appearance model A_t for class t (the object detector). This is solved in an iterative fashion, by alternating between two steps until convergence:

1. **Re-localization:** in each positive image I, select the proposal b^* with the highest score given by the current appearance model A_t:

$$b^* \equiv \underset{b \in \mathcal{B}}{\operatorname{argmax}} A_t(b, I) \tag{1}$$

2. **Re-training:** re-train A_t using the current selection of proposals from the positive images, and all proposals from negative images.

Features and appearance model. Typical MIL implementations use as appearance model a linear SVM trained on CNN-features extracted from the object proposals [14, 7, 2, 3, 41, 48].

Initialization In the first iteration many works train the appearance model by using complete images minus a small boundary as positive training examples [6, 7, 30, 36, 29, 22].

Multi-folding. In a high dimensional feature space the SVM separates positive and negative training examples well, placing most positive samples far from the decision hyperplane. Hence, during re-localization the detector is likely to score the highest on the object proposals which were used as positive training samples in the previous iteration. This leads MIL to get stuck on some incorrect selection in early iterations. To prevent this, [7] proposed multi-folding: the training set is split into 10 subsets, and then the re-localization on each subset is done using detectors trained on the union of all other subsets.

Objectness. Objectness was proposed in [1] to measure how likely it is that a proposal tightly encloses an object of any class (e.g. bird, car, sheep), as opposed to background (e.g. sky, water, grass). Since [8], many WSOL techniques [4, 7, 31, 37, 40, 38, 45, 49] have used an objectness measure [1, 11] to steer the re-localization process towards objects and away from background. Following standard practice, incorporating objectness into Eq (1) leads to:

$$b^* \equiv \underset{b \in \mathcal{B}}{\operatorname{argmax}}\, \lambda \cdot A_t(b, I) + (1 - \lambda) \cdot O(b, I) \tag{2}$$

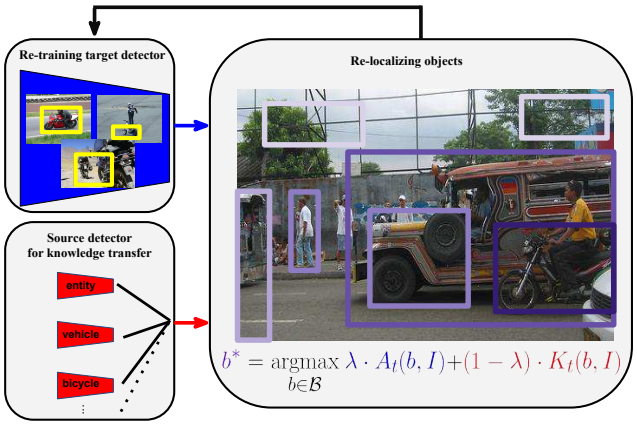

Figure 2. Illustration of MIL + knowledge transfer for the target class 'motorbike'. Standard MIL consists of a re-training stage and a re-localization stage (Sec. 2.1). We add knowledge transfer to this scheme by training SSD [28] on the hierarchy $\mathcal{H}$ defined by the source set $\mathcal{S}$. We use its proposals and a knowledge transfer function (Sec. 2.3) in the re-localization stage.

where λ is a weight controlling the trade-off between the class-generic objectness score O and the appearance model A_t of the target class t being learned during MIL. Using the objectness score in this manner, previous works typically report an improvement around 5% in correct localizations of the target objects [4, 8, 7, 31, 37, 40, 38, 45, 49].

2.2. MIL with knowledge transfer

Overview. In the standard WSOL setting, MIL is applied only on the training set $\mathcal{T}$ of target classes with image-level labels. In our setting we also have a training set of source classes $\mathcal{S}$ with bounding-box annotations. Therefore we incorporate knowledge from the source classes into MIL and help learning detectors for the target classes (Fig. 1). We train a multi-class object detector over all source classes $\mathcal{S}$ organized in a semantic hierarchy, and then apply it to $\mathcal{T}$ as a proposal generator. This naturally provides high-quality proposals, as well as proposal scoring functions at multiple levels in the hierarchy. We use these scoring functions during the re-localization stage of MIL on $\mathcal{T}$ (Fig. 2), which greatly helps localizing target objects correctly (Sec. 3).

Our scheme improves over previous usage of manually engineered object proposals and their associated objectness score [1, 11] in WSOL in two ways: (1) by using *trained* proposals and objectness scoring function trained on source classes; (2) generalizing the common use of a single class-agnostic objectness score to a *family of proposal scoring functions* at multiple levels of semantic specificity. This enables exploring using scoring functions tailored to particular target classes, and at various degrees of relatedness between source and target classes (Sec. 2.3). Below we explain the elements of our approach in more detail.

Training a proposal generator on the source set. We use the Single Shot Detection (SSD) network [28]. SSD

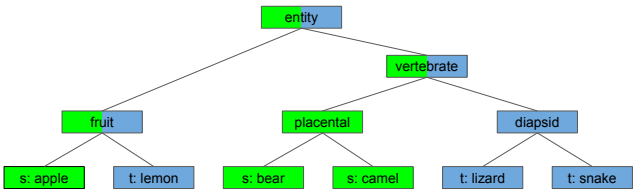

Figure 3. Illustration of part of the ImageNet hierarchy, with our source and target classes inside it. Source classes in $\mathcal{S}$ are the leaf nodes in green. Target classes in $\mathcal{T}$ are leaf nodes in blue. For other nodes the color shows whether it has only source classes as leaves under it, only target classes, or a mixture of both.

starts from a dense grid of 'anchor boxes' covering the image, and then adjusts their coordinates to match objects using regression. This in turn enables substituting Region-of-Interest pooling [16, 13, 33] with convolutions, yielding considerable speed-ups at a small loss of performance [19]. The SSD implementation we use has Inception-V3 [44] as base network and outputs 1296 boxes per image.

We train SSD on the source set $\mathcal{S}$. For each anchor box, SSD regresses to a single output box, along with one confidence score for each source class. Therefore, the proposal set $\mathcal{B}$ generated for an image is class-generic. Before training SSD, we first position the source classes $\mathcal{S}$ into the ImageNet semantic hierarchy $\mathcal{H}$ [35] and expand the label space to include all ancestor classes up to the top-level class 'entity' (Fig. 3). After this expansion, each object bounding-box has multiple class labels, including its original label from $\mathcal{S}$ (e.g. 'bear') and all its ancestors up to 'entity' (e.g. 'placental', 'vertebrate', 'entity'). Hence, we train SSD in a multi-label setting, and we use a sigmoid cross entropy loss for each class separately, instead of the common log softmax loss across classes (which is suited for standard 1-of-K classification, e.g. [13, 28, 33]). Note how this entails that ancestor classes use as training samples the set union of all samples over their descendants in $\mathcal{S}$.

We stress that we use SSD instead of Fast- or Faster-RCNN detectors [13, 33] because these detectors perform class-specific bounding-box regression. That leads to different sets of boxes for each source class, which complicates experiments in our knowledge transfer setting. SSD instead delivers a single set of class-generic boxes, and attaches to each box multiple scores (one per source class).

Knowledge transfer during re-localization. After training SSD on $\mathcal{S}$, we apply it to each image I in the target set $\mathcal{T}$. It produces a set of proposals $\mathcal{B}$ and assigns to each proposal $b \in \mathcal{B}$ scores $F_s(b, I)$ at all levels of the hierarchy. More precisely, it assigns a score for each class $s \in \mathcal{H}$, including scores for the original leaf classes $\mathcal{S}$, the intermediate-level classes, and the top-level 'entity' class. This top-level score corresponds conceptually to traditional objectness [1, 11], but is now properly trained.

We use this family of scoring functions F_s to compose one particular knowledge transfer function $K_t(b, I)$ tailored to each target class $t \in \mathcal{T}$. We discuss in Sec. 2.3 four strategies for composing K_t. As illustrated in Fig. 2, we use K_t inside the re-localization stage of MIL by generalizing Eq (2) to become:

$$b^* \equiv \underset{b \in \mathcal{B}}{\operatorname{argmax}}\, \lambda \cdot A_t(b, I) + (1 - \lambda) \cdot K_t(b, I) \qquad (3)$$

Note how the special case of $K_t(b, I) = O$ (using a standard objectness score [11, 1]) and $\mathcal{B}$ coming from a standard object proposal generator corresponds to WSOL with the reference MIL algorithm (sec. 2.1).

2.3. Knowledge transfer functions K_t

In this paper we explore knowledge transfer at different levels of semantic generality. For a given target class t, we use the proposal scoring functions F_s to compose a knowledge transfer function K_t at a desired levels of generality, out of four possible options: class-generic, closest source classes, closest common ancestor, and closest common ancestor with at least n sources.

Class-generic objectness. The most generic way of transferring knowledge is to use the scoring function $F_{\text{entity}}(b, I)$ from the top-level class 'entity' in the hierarchy. The idea is that such a generic measure generalizes beyond the source classes it was trained on, and it helps steer the re-localization process towards objects and away from background in the target dataset. This corresponds to the traditional use of objectness in WSOL [8, 7, 31, 37, 40, 38, 45, 49, 4], but done with a stronger objectness measure trained in a neural network:

$$K_t(b, I) \equiv F_{\text{entity}}(b, I) \qquad (4)$$

In our main experiments, this scoring function is trained on the set union over the training samples of all 100 source classes in the dataset we use (Sec. 3).

Closest source classes. On the other end of the spectrum, we can transfer knowledge from the most similar source classes to the target t, the most common scenario in a knowledge transfer setting [15, 18, 34, 38, 46]. To find these source classes, we consider the position of t in the semantic hierarchy $\mathcal{H}$. We find the closest ancestor a_1 of t whose descendants include at least one source class from $\mathcal{S}$, and then take all leaf-most source classes among its descendants. Often these closest sources are the siblings of t (e.g. lemon for apple in Fig. 3), but if t has no siblings in $\mathcal{S}$ the procedure backtracks to a higher-level ancestor and takes its descendants instead (e.g. bear and camel for lizard in Fig. 3). In practice, a target class has a median number of 3 closest source classes in our main experiments (Sec. 3).

We combine the scoring functions of the closest source classes $\mathcal{N}_t$ into the knowledge transfer function as:

$$K_t(b, I) \equiv \frac{1}{|\mathcal{N}_t|} \sum_{s \in \mathcal{N}_t} F_s(b, I) \quad (5)$$

Closest common ancestor. A different way to use the training data from the closest source classes is to directly use the scoring function F_{a_1} of the closest ancestor a_1 of t who has descendants in $\mathcal{S}$:

$$K_t(b, I) \equiv F_{a_1}(b, I) \quad (6)$$

The scoring function F_{a_1} is trained from the set union of the training data over all closest source classes $\mathcal{N}_t$ (instead of averaging the scoring function outputs in Eq. (5)).

Closest common ancestor with at least n sources. The extreme cases above present a trade-off. The 'entity' class is trained from a lot of samples, but is very generic. In contrast, the closest source classes are more specific to the target, but have less training data to form strong detectors.

Here we propose an intermediate approach: we control the degree of generality of transfer by setting a minimum to the number of source classes an ancestor should have. We define a_n as the closest ancestor of t who has at least n source classes as descendants. This generalizes Eq. (6) to:

$$K_t(b, I) \equiv F_{a_n}(b, I) \quad (7)$$

Note that setting $n = |\mathcal{S}|$ leads to to selecting the *entity* class as ancestor, matching Eq. (4). In our experiments in Sec. 3 we set $n = 5$, resulting in a median of 10 source classes under the ancestor selected for each target class.

3. Results on ILSVRC 2013

Dataset. We use ILSVRC 2013 [35], following exactly the same settings as [18, 46] to enable direct comparisons. We split the ILSVRC 2013 validation set into two subsets val1 and val2, and augment val1 with images from the ILSVRC 2013 training set such that each class has 1000 annotated bounding-boxes in total [14]. ILSVRC 2013 has 200 object classes: we use the first 100 as sources $\mathcal{S}$ and second 100 as targets $\mathcal{T}$ (classes are alphabetically sorted).

As our *source training set* we use all images of the augmented val1 set which have bounding-box annotations for 100 source classes, resulting in 63k images with 81k bounding-boxes. As *target training set* we use all images of the augmented val1 set which contain the 100 target classes and remove all bounding-box annotations, resulting in 65k images with 93k image-level labels. In Sec. 3.1 we report results for MIL applied to the target training set. As *target test set* we use all 10k images of val2 and remove all annotations. In Sec. 3.2 we train a detector from the output of MIL on the target training set, and evaluate it on the target test set. Finally, in Sec. 3.3 we compare to three previous works [18, 46, 32] on knowledge transfer on ILSVRC 2013.

3.1. Knowledge transfer to the target training set

We first explore the effects of knowledge transfer for localizing objects in the target training set.

Settings for MIL with knowledge transfer. We train SSD [28] on the source training set and apply it to the target training set to produce object proposals and corresponding scores (Sec. 2.2). Then we apply MIL on the target training set (Sec. 2.2) while varying the knowledge transfer function K_t (Sec. 2.3) during re-localization (Eq. (3)).

Following [2, 3, 7, 41, 48] during MIL we describe each object proposals with a 4096-dimensional feature vector using the Caffe implementation [20] of the AlexNet CNN [25]. As customary, we use weights from [20] resulting from training on ILSVRC classification [35] using only image-level labels (no bounding-box annotations). As appearance model we use a linear SVM on these features.

For each knowledge transfer function, we optimize λ in Eq.(3) on the source training set in a cross-validation manner: we subdivide this set in 80 source classes and 20 target classes, run our knowledge transfer framework, and choose the λ which leads to the highest localization performance (CorLoc, see below).

Evaluation measure. We quantify localization performance with Correct Localization (CorLoc) [8] averaged over the target classes $\mathcal{T}$. CorLoc is the percentage of images of class t where the method correctly localizes one of its instances. We consider two Intersection-over-Union (IoU) [12] thresholds: we report CorLoc at IoU > 0.5 (commonly used in the WSOL literature [4, 7, 8, 45]) and IoU > 0.7 (stricter criterion requiring tight localizations).

Quantitative results. The first two rows of Table 1 report CorLoc on the target training set. As a baseline we use no knowledge transfer function at all. This leads to 41.4% CorLoc at IoU > 0.5.

All the forms of knowledge transfer we explore yield massive improvements over the baseline: 27-29% CorLoc increase. Interestingly, simply transferring from the top-level 'entity' class works best and yields 70.3% CorLoc. This shows that the trade-off between semantic generality and number of source training samples is skewed towards the former. We believe this is excellent news for the practitioner: our experiments show that a simple modification to standard MIL pipelines can lead to dramatic improvements in localization performance (i.e. just change the scoring function used during re-localization to include a strong objectness function trained on the source set).

	Knowledge transfer results					EdgeBoxes baseline		Full supervision
Knowledge transfer function K_t	none	closest sources	closest ancestor	ancestor a_5	class-generic	none	objectness	-
CorLoc IoU > 0.5	41.4	68.5	68.4	68.6	**70.3**	39.4	50.5	-
CorLoc IoU > 0.7	20.0	56.4	56.4	56.6	**58.8**	18.4	28.5	-
mAP IoU > 0.5	23.2	35.3	34.8	35.9	**36.9**	20.7	25.4	46.2
mAP IoU > 0.7	7.0	25.7	25.5	25.9	**27.2**	6.7	11.0	31.7

Table 1. Results for various knowledge transfer functions and full supervision on the target training set (CorLoc) and target test set (mAP). Knowledge transfer results significantly outperform the MIL baseline (EdgeBoxes). Class-generic knowledge transfer works best.

When measuring CorLoc at the stricter IoU> 0.7 threshold, the benefits of knowledge transfer are even more pronounced. The baseline without knowledge transfer only brings 20% CorLoc, while class-generic transfer achieves 58.8% CorLoc, almost $3\times$ higher. This suggests that knowledge transfer enables localizing objects with tighter bounding-boxes.

Reference MIL with manually engineered proposals. In the previous experiments we transferred knowledge from the source classes not only via the knowledge transfer functions, but also by using trained object proposals: the locations of the proposals produced by SSD on the target training set are influenced by the locations of the objects in the source training set.

To eliminate all forms of knowledge transfer, we perform here experiments using the same MIL framework as before, but now using untrained, manually engineered EdgeBox proposals [11]. Without using any objectness function during re-localization (Eq. (1)), this baseline obtains 39.4% CorLoc at IoU > 0.5. In contrast, our SSD proposals without any knowledge transfer function yields 41.4% CorLoc. This shows that trained object proposal locations helps only a little. Furthermore, using also the untrained, class-generic objectness of Edgeboxes during re-localization (Eq. (2)) results in 50.5% CorLoc, an improvement of 11%. In contrast, our trained class-generic knowledge transfer yields 70.3% CorLoc, a much higher improvement of 29%. This system (MIL with EdgeBoxes and Objectness) is the reference MIL of Sec. 2.1, which represents a standard WSOL method without learned knowledge transfer functions.

The above experiments demonstrate that the major reason for the performance improvement brought by our knowledge transfer scheme is the knowledge transfer functions, not the trained proposals.

A closer look at closest sources. The previous section showed that class-generic transfer outperforms class-specific transfer in our experiments. As this may seem counter-intuitive, we investigate here whether our closest source strategy could be improved.

Above we used distance in the WordNet hierarchy to find the closest source classes to a target, which reflects semantic similarity rather than visual similarity. Here we perform an additional experiment using visual similarity instead. We extract whole-image features on the source and target training sets using an AlexNet [25] classification network pretrained on ILSVRC classification [35]. For each class, we average the features of all its training images. For each target class, the closest source class is the one with the most similar averaged features, measured in Euclidean distance.

closest source strategy	CorLoc IoU > 0.5
WordNet hierarchy	68.5
Visual similarity	68.0
Best source upper-bound	69.6
class-generic	70.3

Table 2. Knowledge Transfer using different ways to determine the closest source class. Even the upper-bound does not outperform class-generic knowledge transfer.

We also compute an upper-bound performance by selecting for each target class the source class that leads to the highest CorLoc on the target training set. This is the best possible source. Note how this experiment needs ground-truth bounding-boxes on the target training set to select a source class, and so it is not a valid strategy in practice. It is only intended to reveal the upper-bound that any way of selecting a source specific to a target cannot exceed.

The results in Tab. 2 show that using either semantic similarity or visual similarity yields similar results: 68.5% and 68.0% CorLoc respectively. Using the best source class improves moderately over both automatic ways to select a source class (69.6% CorLoc). Interestingly, even the best source class does not outperform class-generic knowledge transfer (70.3% CorLoc). This is likely due to the fact that individual source classes have too little training data to form strong detectors, whereas the class-generic objectness model benefits from a very large training set (effectively the set union of all sources).

Correlation between semantic similarity and improvement. We now investigate whether there is a relation between the improvements brought by our class-generic knowledge transfer on a particular target class, and its semantic similarity to the source classes. We measure semantic similarity by the widely used Lin similarity [26] on WordNet (same hierarchy as ImageNet). For each target class in $\mathcal{T}$, the horizontal axis in Fig. 4 reports the similarity of the most similar source class in $\mathcal{S}$. The vertical axis reports the absolute CorLoc improvement on the target training set, over the no-transfer baseline.

Interestingly, we observe no significant correlation between CorLoc improvement and semantic similarity. This suggests that this knowledge transfer function, trained on a large set of 100 diverse source classes, is truly class generic.

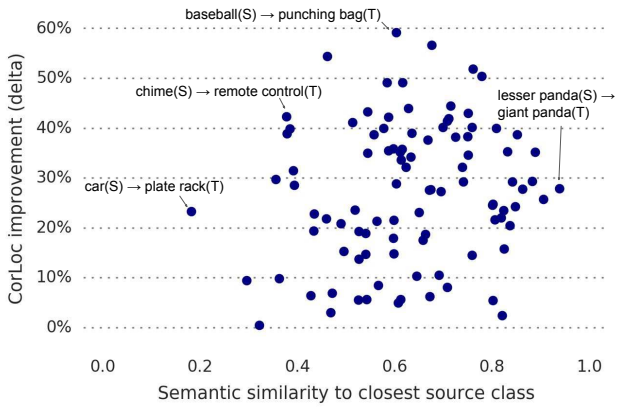

Figure 4. Absolute CorLoc improvement brought by our class-generic knowledge transfer, as a function of semantic similarity [26] between a target class and the most similar source class. Each point represents one target class. For several points we show which source class (S) it represents, and its most similar class (T).

3.2. Object detection on the target test set

We now train an object detector from the bounding-boxes produced on the target training set by MIL. We train a Faster-RCNN detector [33] with Inception-ResNet [43] as base network. We apply it to the target test set and report mean Average Precision (mAP) [12, 35].

As Tab. 1 shows, mAP on the test set correlates very well with CorLoc on the training set. At IoU$>$ 0.5, the best results are brought by our class-generic transfer method (36.9% mAP), strongly improving over the no-transfer baseline (23.2%) and the EdgeBoxes + objectness baseline (25.4%). Results at IoU$>$ 0.7 reveal an interesting phenomenon: both baselines fail to train an object detector that localizes objects accurately enough (7.0% mAP for no-transfer, 11.0% mAP for EdgeBoxes + objectness). Instead, our class-generic knowledge transfer scheme succeeds even at this strict threshold. Its mAP (27.2%) is around $4\times$ and $2.5\times$ higher than the baselines. To put our results in context, we also report mAP when training on the target training set with ground-truth bounding-boxes, which acts as an upper-bound ('full supervision' in Tab. 1). At IoU $>$ 0.5 and IoI $>$ 0.7, our scheme reaches 80% and 86% of this upper-bound, respectively.

These experiments consolidates our findings and shows that our simple class-generic transfer strategy is effective in improving the performance of object detectors for target classes for which only image-level labels are available.

3.3. Comparison to [18, 46, 32]

We now compare our technique to two transfer learning works [18, 46] using the exact same dataset with the same source and target training splits as in [18, 46] (see Sec. 3).

In terms of CorLoc on the target training set, LSDA [18] reports 28.8% at IoU $>$ 0.5 while our method delivers 70.3%, more than twice higher (Tab. 3). Note that [18, 46] and our MIL method all use the same base network (AlexNet [25]) to produce feature descriptors for proposals. Hence, they are directly comparable.

	CorLoc IoU $>$ 0.5	mAP IoU $>$ 0.5
LSDA [18]	28.8	18.1
Tang et al. [46]	-	20.0
our method	70.3	23.3

Table 3. Comparison of our results to [18, 46] at IoU $>$ 0.5. All numbers presented in this table use AlexNet [25] as base network. Tang et al. [46] does not report CorLoc.

In terms of performance on the target test set, in order to make a fair comparison to [18, 46], we train a Fast-RCNN detector model [13] using the same base network: AlexNet [25] (as opposed to the results in Tab. 1, which use a stronger detector). Our method leads to detectors performing at 23.3 mAP on the target test set, improving over the 20.0 by [46] and 18.1 by [18]. Moreover, our method is also much simpler: just insert a properly trained class-generic objectness scoring function into standard MIL pipelines.

We also compare to YOLOv2 following their settings (Section 4 in [32]): COCO train as the source training set, ImageNet-classification as the target training set, and ILSVRC-detection validation as the target test set. In our setup we subsample ImageNet-classification by randomly selecting up to 1K images for each of the 200 target classes.

In the spirit of knowledge transfer, [32] report results over the 156 target classes that are not present in COCO. On those classes, our method yields 32.2 mAP, substantially better than the 16.0 mAP of [32].

We note that the object detection model that we use seems approximately comparable to YOLOv2 (Table 3 of [32]). This shows that our improvement is due to better transfer learning.

4. Generalization across datasets

The experiments in Sec. 3 suggest a relatively easy recipe for knowledge transfer for WSOL: train a strong class-generic proposal generator on a source training set with object bounding-boxes, and use its proposals and scores inside MIL on a target set with only image-level labels. We demonstrate here this recipe in several cross-dataset experiments, going beyond within-dataset transfer typically shown in previous works [15, 18, 34, 38, 46].

For these experiments, we switch to a stronger object proposal generator than SSD with Inception-V3: Faster-RCNN [33] with Inception-ResNet[43]. Note that considering a single class-generic 'entity' class avoids the technical problem raised in Sec. 2.2, as Faster-RCNN will now output a single set of proposals, along with a single score (for objectness). The rest of our framework remains unaltered.

As source training sets we use: (1) the ILSVRC 2013 source training set as defined in Sec. 3 (100 classes, 63k images), (2) the COCO 2014 training set [27] (80 classes, 83k images), and (3) the PASCAL VOC 2007 trainval set [12]

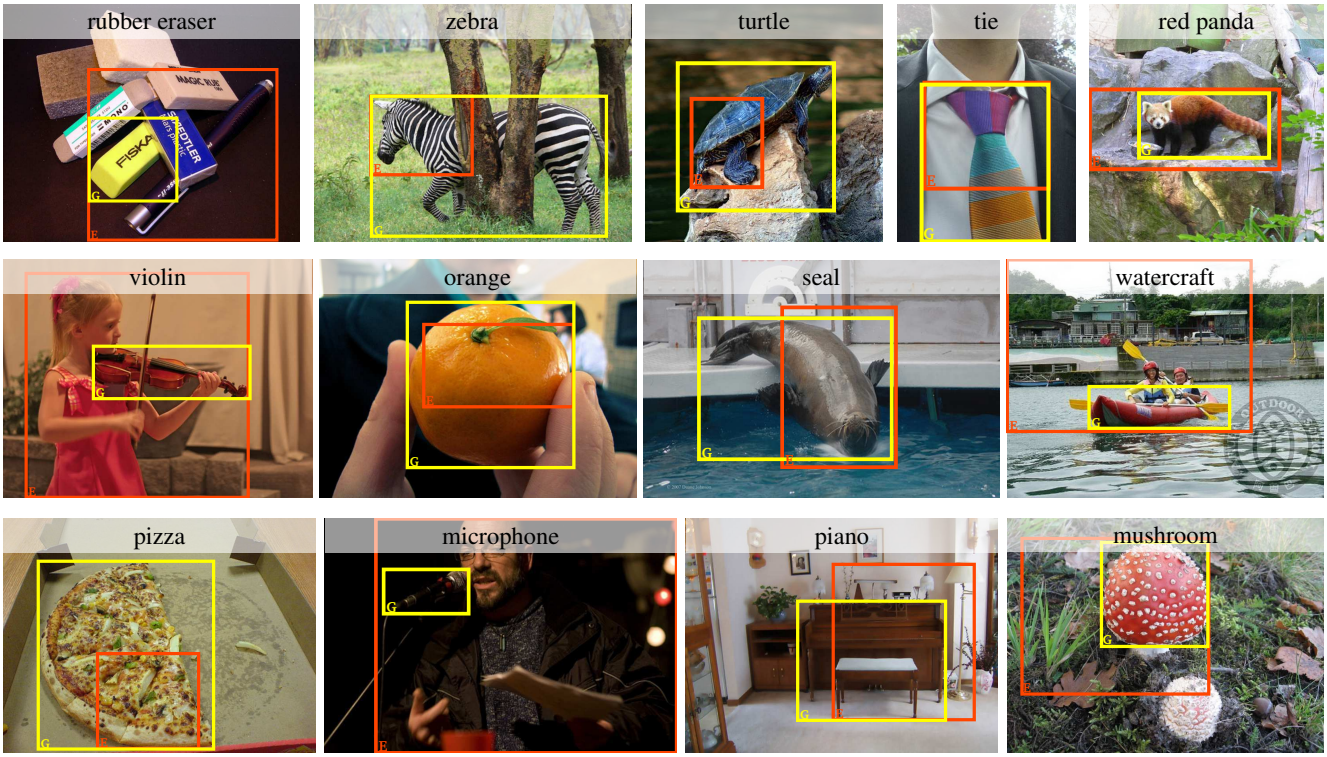

Figure 5. Example localizations produced by our class-generic knowledge transfer scheme (yellow) and by the EdgeBox+objectness baseline (red) on the target training set. Our technique steers localization towards complete objects and away from backgrounds. Labels are shown on the images.

target set / source set	ILSVRC target		COCO 2014 train		OID V2 val	
	> 0.5	> 0.7	> 0.5	> 0.7	> 0.5	> 0.7
ILSVRC source	74.2	61.7	34.5	26.8	62.0	51.8
COCO 2014 train	67.7	58.5	-	-	59.5	49.8
PASCAL 2007 trainval	59.5	47.2	26.2	20.8	55.3	42.2
EdgeBox + objectness	50.5	28.5	20.6	10.2	32.4	16.3

Table 4. MIL + Knowledge transfer across datasets: CorLoc results for IoU > 0.5 and > 0.7 on target datasets. Even knowledge transfer from the small PASCAL 2007 trainval works better than the baseline of EdgeBoxes with objectness. Generally, transfer works better when the source training set contains more classes. Note that CorLoc when transferring from ILSVRC 2013 source train to ILSVRC 2013 target train is higher than in Tab. 1 due to using a stronger proposal generator.

(20 classes, 5011 images). As target training sets we use (1) the ILSVRC 2013 target training set as defined in Sec. 3 (100 classes, 65k images), (2) the COCO 2014 training set [27] (80 classes, 83k images), and (3) the Open Images V2 dataset [24], combining the validation and test set [24] (600 classes, 167k images). In this experiment we do not try to remove source classes from the target training sets.

Tab. 4 presents our across-dataset results and the MIL baseline using EdgeBoxes [11] with objectness (Sec. 2.1). We observe that the knowledge transfer method considerably outperform the baseline for all dataset pairs. This is especially true at IoU > 0.7, where even using the small PASCAL VOC 2007 dataset as source yields 1.6-2.6$\times$ higher CorLoc than the baseline. Furthermore, using more source classes is consistently better for all target datasets: ILSVRC 2013 (100 classes) is the best source, followed by COCO 2014 (80 classes), and then by PASCAL VOC 2007 (20 classes). This is despite COCO 2014 train having many more object instances (605k boxes) than the ILSVRC 2013 source train set (81k boxes). We conclude that our knowledge transfer strategy works much better than the weakly supervised baseline and generalizes well across datasets.

5. Conclusions

We proposed a unified knowledge transfer framework for weakly supervised object localisation, which enabled exploring knowledge transfer functions ranging from class-specific to class-generic. Our experiments on ILSVRC [35] demonstrate: (1) knowledge transfer at any level of generality substantially improve results, with class-generic knowledge transfer working best. (2) class-generic knowledge transfer leads to large improvements over a weakly supervised baseline using manually engineering objectness [11]: +19.8% CorLoc and +11.5% mAP. (3) our method delivers target class detectors reaching 80% of the accuracy of their fully supervised counterparts. (4) we outperform the best reported transfer learning results on this dataset: +41% CorLoc and +3% mAP over [18, 46], +16.2% mAP over [32]. Moreover, across-dataset [27, 24, 35] experiments demonstrate (5) the general applicability of our technique.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, 2010. 1, 2, 3, 4

[2] H. Bilen, M. Pedersoli, and T. Tuytelaars. Weakly supervised object detection with posterior regularization. In *BMVC*, 2014. 1, 3, 5

[3] H. Bilen, M. Pedersoli, and T. Tuytelaars. Weakly supervised object detection with convex clustering. In *CVPR*, 2015. 3, 5

[4] H. Bilen and A. Vedaldi. Weakly supervised deep detection networks. In *CVPR*, 2016. 1, 2, 3, 4, 5

[5] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni. Hierarchical classification: combining bayes with svm. In *ICML*, 2006. 2

[6] R. Cinbis, J. Verbeek, and C. Schmid. Multi-fold mil training for weakly supervised object localization. In *CVPR*, 2014. 3

[7] R. Cinbis, J. Verbeek, and C. Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *IEEE Trans. on PAMI*, 2016. 1, 2, 3, 4, 5

[8] T. Deselaers, B. Alexe, and V. Ferrari. Localizing objects while learning their appearance. In *ECCV*, 2010. 1, 2, 3, 4, 5

[9] T. Deselaers, B. Alexe, and V. Ferrari. Weakly supervised localization and learning with generic knowledge. *IJCV*, 2012. 1

[10] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997. 1

[11] P. Dollar and C. Zitnick. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 1, 2, 3, 4, 6, 8

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV*, 2010. 5, 7

[13] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 4, 7

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 3, 5

[15] M. Guillaumin and V. Ferrari. Large-scale knowledge transfer for object localization in imagenet. In *CVPR*, 2012. 1, 2, 4, 7

[16] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014. 4

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR*, 2016. 1

[18] J. Hoffman, D. Pathak, E. Tzeng, J. Long, S. Guadarrama, and T. Darrell. Large scale visual recognition through adaptation using joint representation and multiple instance learning. *JMLR*, 2016. 1, 2, 4, 5, 7, 8

[19] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. In *CVPR*, 2017. 4

[20] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. http://caffe.berkeleyvision.org/, 2013. 5

[21] V. Kantorov, M. Oquab, M. Cho, and I. Laptev. Contextlocnet: Context-aware deep network models for weakly supervised localization. In *ECCV*, 2010. 1

[22] G. Kim and A. Torralba. Unsupervised detection of regions of interest using iterative link analysis. In *NIPS*, 2009. 3

[23] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In *ICML*, 1997. 2

[24] I. Krasin, T. Duerig, N. Alldrin, V. Ferrari, S. Abu-El-Haija, A. Kuznetsova, H. Rom, J. Uijlings, S. Popov, A. Veit, S. Belongie, V. Gomes, A. Gupta, C. Sun, G. Chechik, D. Cai, Z. Feng, D. Narayanan, and K. Murphy. Open-images: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2017. 1, 2, 8

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 5, 6, 7

[26] D. Lin. An information-theoretic definition of similarity. In *ICML*, 1998. 6, 7

[27] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 1, 2, 7, 8

[28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. 1, 2, 3, 4, 5

[29] M. Nguyen, L. Torresani, F. de la Torre, and C. Rother. Weakly supervised discriminative localization and classification: a joint learning process. In *ICCV*, 2009. 1, 3

[30] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *ICCV*, 2011. 3

[31] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. Learning object class detectors from weakly annotated video. In *CVPR*, 2012. 1, 2, 3, 4

[32] J. Redmon and A. Farhadi. YOLO9000: Better, faster, stronger. In *CVPR*, 2017. 1, 2, 5, 7, 8

[33] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1, 4, 7

[34] M. Rochan and Y. Wang. Weakly supervised localization of novel objects using appearance transfer. In *CVPR*, 2015. 1, 2, 4, 7

[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. *IJCV*, 2015. 1, 2, 4, 5, 6, 7, 8

[36] O. Russakovsky, Y. Lin, K. Yu, and L. Fei-Fei. Object-centric spatial pooling for image classification. In *ECCV*, 2012. 1, 3

[37] N. Shapovalova, A. Vahdat, K. Cannons, T. Lan, and G. Mori. Similarity constrained latent support vector machine: An application to weakly supervised action classification. In *ECCV*, 2012. 1, 2, 3, 4

[38] Z. Shi, P. Siva, and T. Xiang. Transfer learning by ranking for weakly supervised object annotation. In *BMVC*, 2012. 1, 2, 3, 4, 7

[39] C. Silla and A. Freitas. A survey of hierarchical classification across different application domains. In *Data Mining and Knowledge Discovery*, 2011. 2

[40] P. Siva and T. Xiang. Weakly supervised object detector learning with model drift detection. In *ICCV*, 2011. 1, 2, 3, 4

[41] H. Song, R. Girshick, S. Jegelka, J. Mairal, Z. Harchaoui, and T. Darell. On learning to localize objects with minimal supervision. In *ICML*, 2014. 1, 3, 5

[42] H. Song, Y. Lee, S. Jegelka, and T. Darell. Weakly-supervised discovery of visual pattern configurations. In *NIPS*, 2014. 1

[43] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *AAAI*, 2017. 7

[44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 4

[45] K. Tang, A. Joulin, L.-J. Li, and L. Fei-Fei. Co-localization in real-world images. In *CVPR*, 2014. 1, 2, 3, 4, 5

[46] Y. Tang, J. Wang, B. Gao, E. Dellandréa, R. Gaizauskas, and L. Chen. Large scale semi-supervised object detection using visual and semantic knowledge transfer. In *CVPR*, 2016. 1, 2, 4, 5, 7, 8

[47] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 2013. 1, 3

[48] C. Wang, W. Ren, J. Zhang, K. Huang, and S. Maybank. Large-scale weakly supervised object localization via latent category learning. *IEEE Transactions on Image Processing*, 24(4):1371–1385, 2015. 1, 3, 5

[49] L. Wang, G. Hua, R. Sukthankar, J. Xue, and J. Zheng. Video object discovery and co-segmentation with extremely weak supervision. In *ECCV*, 2014. 1, 2, 3, 4

[50] X. Zeng, W. Ouyang, B. Yang, J. Yan, and X. Wang. Gated bi-directional cnn for object detection. In *ECCV*, 2016. 1