International Journal of Computer Vision (2019) 127:1279–1302
https://doi.org/10.1007/s11263-019-01183-3

Unsupervised Learning of Foreground Object Segmentation

Ioana Croitoru[1] · **Simion-Vlad Bogolin**[1] · **Marius Leordeanu**[1,2]

Received: 23 June 2018 / Accepted: 29 April 2019 / Published online: 13 May 2019

Abstract
Unsupervised learning represents one of the most interesting challenges in computer vision today. The task has an immense practical value with many applications in artificial intelligence and emerging technologies, as large quantities of unlabeled images and videos can be collected at low cost. In this paper, we address the unsupervised learning problem in the context of segmenting the main foreground objects in single images. We propose an unsupervised learning system, which has two pathways, the teacher and the student, respectively. The system is designed to learn over several generations of teachers and students. At every generation the teacher performs unsupervised object discovery in videos or collections of images and an automatic selection module picks up good frame segmentations and passes them to the student pathway for training. At every generation multiple students are trained, with different deep network architectures to ensure a better diversity. The students at one iteration help in training a better selection module, forming together a more powerful teacher pathway at the next iteration. In experiments, we show that the improvement in the selection power, the training of multiple students and the increase in unlabeled data significantly improve segmentation accuracy from one generation to the next. Our method achieves top results on three current datasets for object discovery in video, unsupervised image segmentation and saliency detection. At test time, the proposed system is fast, being one to two orders of magnitude faster than published unsupervised methods. We also test the strength of our unsupervised features within a well known transfer learning setup and achieve competitive performance, proving that our unsupervised approach can be reliably used in a variety of computer vision tasks.

Keywords Unsupervised learning · Foreground object segmentation · Object discovery in video · Transfer learning

1 Introduction

Unsupervised learning is one of the most difficult and interesting problems in computer vision and machine learning today. Many researchers believe that learning from large collections of unlabeled videos could help decode hard questions regarding the nature of intelligence and learning. Moreover, as unlabeled images and videos are easy to collect at relatively low cost, unsupervised learning could be of real practical value in many computer vision and robotics applications. In this article, we propose a novel approach to unsupervised learning that successfully tackles many of the challenges associated with this task. We present a system that is composed of two main pathways, one that performs unsupervised object discovery in videos or large image collections—the teacher branch, and the other—the student branch, which learns from the teacher to segment foreground objects in single images. The unsupervised learning process could continue over several generations of students and teachers. In Algorithm 1, we present the high level description of our method. We will use throughout the paper the terms "generation" and "iteration" of Algorithm 1 interchangeably. The key aspects of our approach, which ensure improvement in performance from one generation to the next, are: (1) the existence of an unsupervised selection module that is able to pick up good quality masks generated by

Communicated by Bernt Schiele.

Ioana Croitoru and Simion-Vlad Bogolin have contributed equally to this work.

✉ Marius Leordeanu
marius.leordeanu@imar.ro

Ioana Croitoru
ioana.croi@gmail.com

Simion-Vlad Bogolin
vladbogolin@gmail.com

[1] Institute of Mathematics of the Romanian Academy, 21 Calea Grivitei, Bucharest, Romania

[2] University "Politehnica" of Bucharest, 313 Splaiul Independentei, Bucharest, Romania

Algorithm 1 Unsupervised learning of foreground object segmentation

Step 1: perform unsupervised object discovery in unlabeled videos (or image collections, at later iterations), along the teacher pathway (module B in Fig. 1).
Step 2: automatically filter out poor soft masks produced at the previous step (module C in Fig. 1).
Step 3: use the remaining masks as supervisory signal for training one or more student nets, along the student pathway (module A in Fig. 1).
Step 4: use as new teacher one or several student nets from the current generation (a new module B) and learn a more powerful soft-mask selector (a new module C), for the next iteration.
Step 5: extend the unlabeled video or image dataset and return to Step 1 to train the next generation (Note: from the first iteration forward, the training dataset can also be extended with collections of unlabeled images, not just videos).

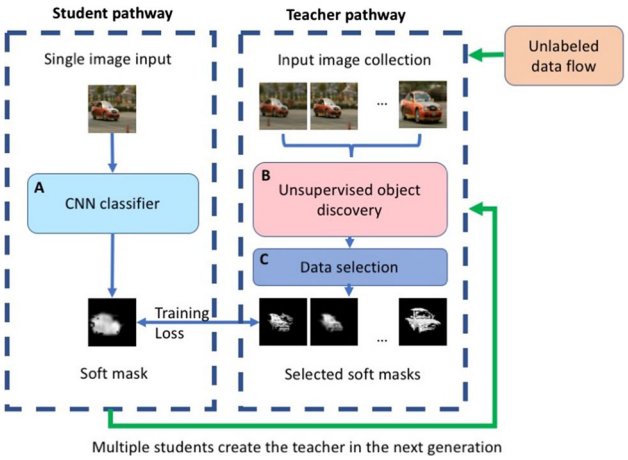

Fig. 1 The dual student-teacher system proposed for unsupervised learning to segment foreground objects in images, functioning as presented in Algorithm 1. It has two pathways: along the teacher branch, an object discoverer in videos or large image collections (module B) detects foreground objects. The resulting soft masks are then filtered based on an unsupervised data selection procedure (module C). The resulting final set of pairs—input image (or video frame) and soft mask for that particular frame (which acts as an unsupervised label)—are used to train the student pathway (module A). The whole process can be repeated over several generations. At each generation several student CNNs are trained, then they collectively contribute to train a more powerful selection module C (modeled by a deep neural network, Sect. 4.3) and form an overall more powerful teacher pathway at the next iteration of the overall algorithm

the teacher and pass them for training to the next generation students; (2) training of multiple students with different architectures, able through their diversity to help train a better selection module for the next iteration and form together with the selection a more powerful teacher pathway at the next iteration and (3) access to larger quantities of, and potentially more complex, unlabeled data, which becomes more useful as the generations become stronger.

Our approach is general in the sense that the student or teacher pathways do not depend on a specific neural network architecture or implementation. Through many experiments and comparisons to state of the art methods, we also show that it is applicable to different tasks in computer vision, such as object discovery in video, unsupervised image segmentation, saliency detection and transfer learning. A preliminary version of our work, presenting an algorithm without learning over several generations and without experiments on saliency detection and transfer learning, appeared at ICCV 2017 (Croitoru et al. 2017).

In Fig. 1 we present a graphic overview of our full system. In the unsupervised training stage the student network (module A) learns, frame by frame, from an unsupervised teacher pathway (modules B and C) to produce similar object masks in single images. Module B discovers objects in images or videos, while module C selects which masks produced by module B are sufficiently good to be passed to module A for training. Thus, the student branch tries to imitate the output of module B for the frames selected by module C, having as input only a single image—the current frame, while the teacher can have access to an entire video sequence.

The strength of the trained student (module A) depends on the performance of the module B. However, as we see in experiments, the power of the selection module C contributes to the fact that the newly student will outperform its initial teacher module B. Therefore, throughout the paper we refer to B as the initial "teacher" and to both B and C together, as the full "teacher pathway". The method presented in Algorithm 1 follows the main steps of the system as it learns from one iteration (generation) to the next. The steps are discussed in more detail in Sect. 3.

During the first iteration of Algorithm 1, the unsupervised teacher (module B) has access to information over time—a video. In contrast, the student is deeper in structure, but it has access only to a single image—the current video frame. Thus, the information discovered by the teacher in time is captured by the student in added depth, over neural layers of abstraction. Several student nets with different architectures are trained at the first iteration. In order to use as supervisory signal only good quality masks, an unsupervised mask selection procedure (very simple at Iteration 1) is applied (module C), as explained in Sect. 4. Once several student nets are trained, they can form (in various ways, as explained in Sects. 4.1 and 5.1) the teacher pathway at the next iteration, along with a stronger unsupervised selection module C, represented by a deep neural network, EvalSeg-Net, trained as explained in detail Sect. 4.3. In short, EvalSeg-Net learns to predict the output masks agreement among the generally diverse students, which statistically takes place when the masks are of good quality. Thus EvalSeg-Net could be used as an unsupervised mask evaluation procedure and a strong selection module. Then, we run, at the next generation, the

newly formed teacher pathway (modules B and C) on a larger set of unlabeled videos or collections of images, to produce supervisory signal for the next generation students. In experiments, we show that the improvement of both modules B and C at the next iterations, together with the increase in the amount of data, are all important, while not all necessary, for increasing accuracy at the next generation.

Note that, while at the first iteration the teacher pathway is required to receive video sequences as input, from the second generation on, it could receive as input large image collections, as well. Due to the very high computational and storage costs, required during training time, we limit our experiments to learning over two generations, but our algorithm is general and could run over many iterations. We show in extensive experiments that even two generations are sufficient to outperform the current state of the art on object discovery in videos and images. We also demonstrate experimentally a solid improvement from one generation to the next for each component involved: the individual students (module A), the teacher (module B), as well as the selection module C.

Now we enumerate the main contributions of our approach and also point out, where it is the case, the key contributions that were not published in our ICCV2017 conference paper (Croitoru et al. 2017):

(1) We introduce a novel approach to unsupervised learning to segment foreground objects in images. The overview of our system and algorithm are presented in Fig. 1 and Algorithm 1. The system has two main pathways—one that acts as a teacher (module B) and discovers objects in videos or large collections of images followed by an unsupervised selection module C that filters out low quality masks, and the other that acts as student and learns from the teacher pathway to detect the foreground objects in single input images. We provide a general algorithm for unsupervised learning over several generations of students and teachers. **In addition to our conference paper**, we show how to learn an unsupervised mask selection deep network (EvalSeg-Net, see Sect. 4.3), which is important in improving the teacher pathway at the next iteration, over all cases tested: when the teacher (module B) is formed by a single student network, by all students combined into an ensemble, or by all students taken separately. The whole unsupervised training at the second generation is a novelty over the conference work, with significantly improved experimental results (see Sect. 5).

(2) At the higher level, our proposed algorithm is sufficiently general to accommodate different implementations and neural network architectures. In this paper, we also provide a specific implementation which we describe in detail. We demonstrate its performance on three unsupervised learning tasks, namely video object discovery tested on YouTube Objects (Prest et al. 2012), unsupervised foreground segmentation in images tested on Object Discovery in Internet Images (Rubinstein et al. 2013) and saliency detection tested on Pascal-S (Li et al. 2014), on which we obtain state of the art results. We further apply our approach to a well-known transfer learning setup and obtain competitive results when compared to the top transfer learning methods in the field. We also compare experimentally our method to the work most related to ours Pathak et al. (2017), on both foreground segmentation and transfer learning tasks and show that our method obtains better results on foreground object segmentation, while theirs is more effective for transfer learning. To the best of our knowledge, we are one of the first two methods, along with the work of Pathak et al. (2017), to propose a system that learns to detect and segment foreground objects in images in unsupervised fashion, with no pre-trained features given or manual labeling, while requiring only a single image at test time. Our experiments on image saliency and transfer learning are completely new and **in addition to our conference paper**.

2 Scientific Context

The literature on unsupervised learning follows two main directions. (1) One is to learn powerful features in an unsupervised way and then use them for transfer learning, within a supervised scheme and in combination with different classifiers, such as SVMs or CNNs (Radenović et al. 2016; Misra et al. 2016; Li et al. 2016). (2) The second direction is to discover, at test time, common patterns in unlabeled data, using clustering, feature matching or data mining formulations (Jain et al. 1999; Cho et al. 2015; Sivic et al. 2005).

Belonging to the first category and closely related to our work, the approach in Pathak et al. (2017) proposes a system in which a deep neural network learns to produce soft object masks from an unsupervised module that uses optical flow cues in video. The deep features learned in this manner are then applied to several transfer learning tasks. Their work, together with ours, are probably the first two that show ways to learn in an unsupervised fashion to segment objects in single images. While the two approaches are clearly different at the technical and algorithmic level, we also perform some interesting comparisons in the experiments Sect. 5.3, on both tasks of transfer learning and foreground object segmentation. Our results reveal that while their approach is better on transfer learning tasks, ours is more effective on unsupervised segmentation as tested on several datasets.

Recently, researchers have started to use the natural, spatial and temporal structure in images and videos as supervisory signals in unsupervised learning approaches that are

considered to follow a *self-supervised learning* paradigm (Raina et al. 2007; Lee et al. 2017; Wang and Gupta 2015). Methods that fall into this category include those that learn to estimate the relative patch positions in images (Doersch et al. 2015), predict color channels (Larsson et al. 2016), solve jigsaw puzzles (Noroozi and Favaro 2016) and inpaint (Pathak et al. 2016). One trend is to use as supervisory signal, spatial and appearance information collected from raw single images. In such single-image cases the amount of information that can be learned is limited to a single moment in time, as opposed to the case of learning from video sequences. Using unlabeled videos as input is closer related to our work and includes learning to predict the temporal order of frames (Lee et al. 2017), generate the future frame (Finn et al. 2016; Xue et al. 2016; Goroshin et al. 2015) or learn from optical flow (Wang and Gupta 2015).

For most of these papers, the unsupervised learning scheme is only an intermediate step to train features that are eventually used on classic supervised learning tasks, such as object classification, object detection or action recognition. Such pre-trained features perform better than randomly initialized ones, as they contain valuable information implicit in the natural structure of the world used as supervisory signal. While the unsupervised features might not contain semantic, class-specific information (Bau et al. 2017), it is clear that they capture general objectness properties, useful for tasks such as segmenting the main objects in the scene or transfer-learning to specific supervised classification problems. In our work, we focus mostly on specific unsupervised tasks on which we perform extensive evaluations, but we also show some results on transfer learning experiments.

The second main approach to unsupervised learning includes methods for image co-segmentation (Joulin et al. 2010, 2012; Kim et al. 2011; Rubinstein et al. 2013; Kuettel et al. 2012; Vicente et al. 2011; Rubio et al. 2012; Leordeanu et al. 2012) and weakly supervised localization (Deselaers et al. 2012; Nguyen et al. 2009; Siva et al. 2013). Earlier methods are based on local features matching and detection of their co-occurrence patterns (Stretcu and Leordeanu 2015; Sivic et al. 2005; Leordeanu et al. 2005; Parikh and Chen 2007; Liu and Chen 2007), while more recent ones (Joulin et al. 2014; Rochan and Wang 2014; Prest et al. 2012) discover object tubes by linking candidate bounding boxes between frames with or without refining their location. Traditionally, the task of unsupervised learning from image sequences has been formulated as a feature matching or data clustering optimization problem, which is computationally very expensive due to its combinatorial nature.

There are also other papers (Lee et al. 2011; Cheng et al. 2017; Dutt Jain et al. 2017; Tokmakov et al. 2017) that tackle unsupervised learning tasks but are not fully unsupervised, using powerful features that are pre-trained in supervised fashion on large datasets, such as ImageNet (Russakovsky et al. 2015) or VOC2012 (Everingham et al. 2015). Such works take advantage of the rich source of supervised information learned from other datasets, through features trained to respond to general object properties over tens or hundreds of object categories. In another paper some amount of supervision is necessary, as in Tokmakov et al. (2016) where a system is proposed having a motion-CNN that learns from weakly annotated videos and optical flow cues to segment objects in video frames. One key difference from our work, is that their approach requires the class labels of the training video frames.

With respect to the end goal, our work is more related to the second research direction, on unsupervised discovery in video. However, unlike that research, we do not discover objects at test time, but during the unsupervised training process, when the student pathway learns to detect foreground objects. Therefore, from the learning perspective, our work is more related to the first research direction based on self-supervised training. While there are published methods that leverage spatiotemporal information in video, our method at the second iteration is able to learn even from collections of unrelated images. Related to our idea of improving segmentations over several iterations, there is the method proposed in Khoreva et al. (2017), which is not unsupervised as it requires the ground truth bounding box information in order to improve the segmentations over several iterations, in conjunction with a modified version of GrabCut (Rother et al. 2004).

3 Overall Approach

We propose a genuine unsupervised learning algorithm (see Algorithm 1) for foreground object segmentation that offers the possibility to improve over several iterations. Our method combines in complementary ways multiple modules that are well suited for this task.

It starts with a teacher (module B, Fig. 1) that discovers objects in unlabeled videos and produces a soft mask of the foreground object in each frame. There are several available methods for video discovery in the literature, with good performance (Borji et al. 2012; Cheng et al. 2015; Barnich and Van Droogenbroeck 2011). We chose the VideoPCA algorithm introduced as part of the system in Stretcu and Leordeanu (2015) because it is very fast (50–100 fps), uses very simple features (individual pixel colors) and it is completely unsupervised, with no usage of supervised pre-trained features. It learns how to separate the foreground from the background and it exploits the spatio-temporal consistency in appearance, shape, movement and location of objects, common in video shots, along with the contrasting properties, in size, shape, motion and location, between the main object and the background scene. Note that it would be much harder,

at this first stage, to discover objects in collections of unrelated images, where there is no smooth variation in shape, appearance and location over time. Only at the second iteration of the algorithm, the simpler VideoPCA is replaced by a more powerful teacher which is able to discover objects in collections of images as well.

The resulting soft-masks of lower quality are then filtered out automatically (module C, Fig. 1), using at the first iteration a very simple automatic procedure. Next, the remaining ones are passed to a student ConvNet, which learns to predict object masks in single images. When several student nets of different architectures are learned, they give the possibility of learning a stronger selection network (module C, Fig. 1) and form a more powerful teacher pathway (modules B and C, Fig. 1) for the next generation. Then, the whole process is repeated. As discussed in Sect. 1, three key aspects contribute to improvement at the second iteration: learning of a more powerful teacher (module B), which could be formed by a single student model or an entire ensemble, learning a stronger selection module C (modeled by a EvalSeg-Net, Sect. 4.3) and last, but not least, increasing the amount of unlabeled data. As shown in the experiments Sect. 5.1, bringing in more data helps only at the second generation when both the teacher and the selection module are improved. In Algorithm 1 we enumerate concisely the main steps of our approach.

4 System Architecture

We detail the architecture and training process of our system, module by module, as seen in Fig. 1. We first present the student pathway (module A in Fig. 1), which takes as input an individual image (e.g. current frame in the video) and learns to predict foreground soft-masks from an unsupervised teacher. The teacher (represented by module B) and the selection module C, are explained in the next Sects. 4.2 and 4.3.

4.1 Student Path (Module A): Single-Image Segmentation

The student pathway (module A in Fig. 1) consists of a deep convolutional network. We test different network architectures, some of which are commonly used in the recent literature on semantic image segmentation. We create a small pool of relatively diverse architectures, presented next.

The first convolutional network architecture for semantic segmentation that we test, is based on a more traditional CNN design. We term it LowRes-Net (see Fig. 2) due to its low resolution soft-mask output. It has ten layers (seven convolutional, two pooling and one fully connected) and skip connections. Skip connections have proved to offer a boost in performance, as shown in the literature (Raiko et al. 2012; Pinheiro et al. 2016). We also observed a similar improvement in our experiments when using skip connections. The LowRes-Net takes as input a 128×128 RGB image (along with its hue, saturation and derivatives w.r.t. x and y) and produces a 32×32 soft segmentation of the main objects present in the image. Because LowRes-Net has a fully connected layer at the top, we reduced the output resolution of the soft-segmentation mask, to limit memory cost. While the derivatives w.r.t x and y are in principle not needed (as they could be learned by appropriate filters during training), in our tests explicitly providing the derivatives along with HSV and by using skip-connections boosted the accuracy by over 1%. The LowRes-Net has a total of 78M parameters, most of them being in the last, fully connected layer.

The second CNN architecture tested, termed FConv-Net, is fully convolutional (Long et al. 2015), as also presented in Fig. 2. It has a higher resolution output of 128×128, with input size of 256×256. Its main structure is derived from the basic LowRes-Net model. Different from LowRes-Net, it is missing the fully connected layer at the end and has more parameters in the convolutional layers, for a total of 13M parameters.

We also tested three different nets based on the U-Net (Ronneberger et al. 2015) architecture, which proved very effective in the semantic segmentation literature. Our U-Net networks are: (1) BasicU-Net, (2) DilateU-Net—similar to BasicU-Net but using atrous (dilated) convolutions (Yu and Koltun 2015) in the *center* module, and (3) DenseU-Net—with dense connections in the *down* and *up* modules (Jégou et al. 2017).

The BasicU-Net has 5 *down* modules with 2 convolutional layers each, with 32, 64, 128, 256 and 512 features maps, respectively. In the *center* module the BasicU-Net has two convolutional layers with 1024 feature maps each. The *up* modules have 3 convolutional layers and the same number of features maps as the corresponding *down* modules. The only difference between BasicU-Net and DilateU-Net is that the former has a different *center* module with 6 atrous convolutions and 512 feature maps each. Then, DenseU-Net has 4 *down* modules with 4 corresponding *up* modules. Each *down* and *up* module has 4 convolutions with skip-connections (as presented in Fig. 2). The modules have 12, 24, 48 and 64 features maps, respectively. The *transition* represents a convolution, having the role of reducing the output number of feature maps from each module. The BasicU-Net has 34M parameters, while the DilateU-Net has 18M parameters. DenseU-Net has only 3M parameters, but uses skip-connections inside the *up* and *down* blocks in order to make up for the difference in the number of parameters. All three U-Nets have 256×256 input and 256×256 output. All networks use ReLU activation functions. Please see Fig. 2

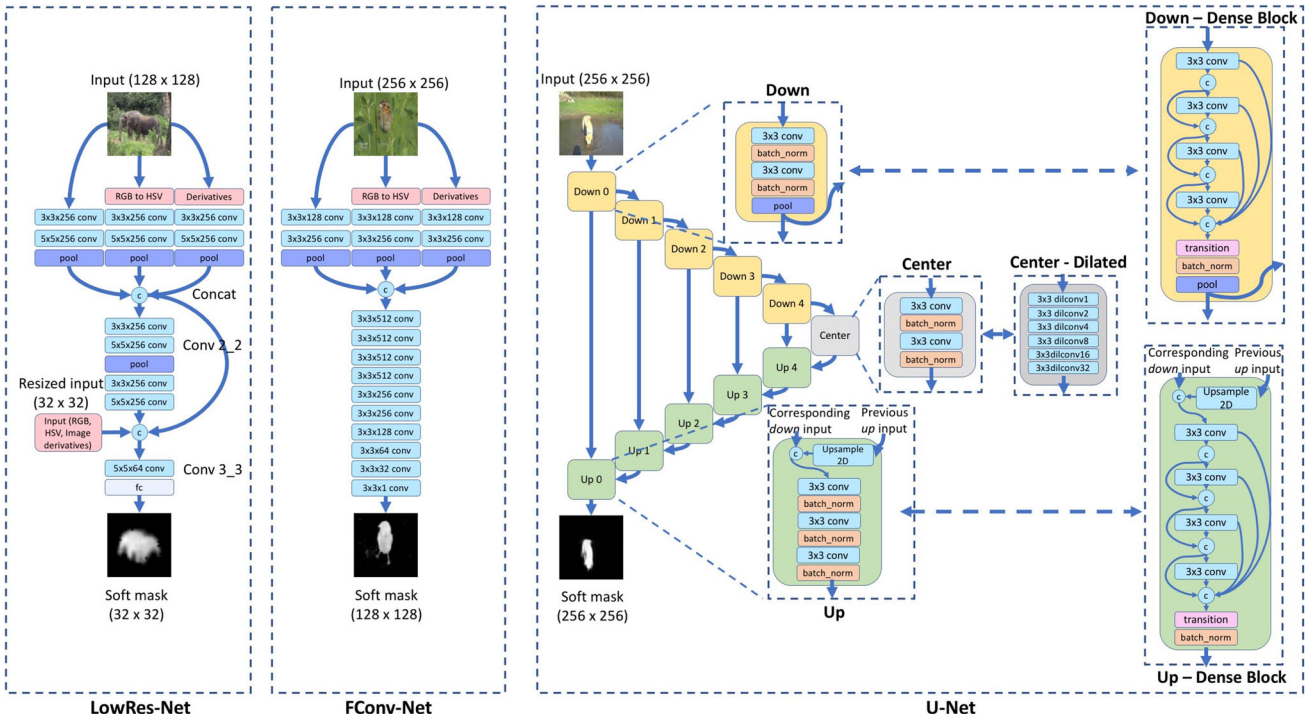

Fig. 2 Different architectures for the "student" networks, each processing a single image. They are trained to predict the unsupervised label masks given by the teacher pathway, frame by frame. The architectures vary from the more classical baseline LowRes-Net (left), with low resolution output, to more recent architectures, such as the fully convolutional one (middle) and different types of U-Nets (right). For the U-Net architecture the blocks denoted with double arrows can be interchanged to obtain a new architecture. We noticed that on the task of bounding box fitting the simpler low-resolution network performed very well, while being outperformed by the U-Nets on fine object segmentation

for more specific details regarding the architectures of the different models.

Given the current setup, the student nets do not learn to identify specific object classes. They will learn to softly segment the main foreground objects present, regardless of their particular category. The main difference in their performance is in their ability to produce fine object segmentations. While the LowRes-Net tends to provide a good support for estimating the object's bounding box due to its simpler output, the other ConvNets (especially the U-Nets), with higher resolution, are better at finely segmenting objects. The different student architectures bring diversity to their outputs. Due to the different ways in which the particular models make mistakes, they are stronger when forming an ensemble and can also be used, as seen in Sect. 4.3, to train a different network for segmentation evaluation, used as the new selection module C. As explained later, that network, namely EvalSeg-Net, will learn to predict the output masks agreement among the students, which statistically takes place when the masks are of good quality. In experiments we also show that the student nets outperform their teacher and are able to detect objects from categories that were not seen during training.

Combining several student nets The student networks with different architectures produce varied results that differ qualitatively. While the bounding boxes computed from their soft-masks have similar accuracy, the actual soft-segmentation output looks different. They have different strengths, while making different kinds of mistakes. Their diversity will be the basis for creating the teacher pathway at the next generation (Sects. 4.2 and 4.3).

We experimented with the idea of using several student networks, by combining them to form an ensemble or by letting them produce separate independent segmentations for each image. In our final system we preferred the latter approach, which is more practical, easier to implement and gives the freedom of having the students run independently, in parallel with no need to synchronize their outputs. As shown in Sects. 4.2 and 4.3, together with the EvalSeg-Net used for selection, independent individual students from Iteration 1 will form the teacher pathway at the next generation. However, note that even a single student net along with the new EvalSeg-Net selector can be effectively used as next teacher pathway (See experimental Sect. 5.1, Table 6 and Fig. 7).

When forming an actual ensemble, which we term Multi-Net, the final output is the one obtained by multiplying

pixel-wise the soft-masks produced by each individual student net. Thus, only positive pixels, on which all nets agree, survive to the final segmentation. As somehow expected, Multi-Net offers robust masks of higher precision than each individual network. However, it might lose details around the border of objects having a lower recall (see Fig. 5). We provide results of the Multi-Net ensemble only for comparison purposes. Please note, however, that in our final system the output of the ensemble was not used to train the students at the next generation. The students at the second iteration are all trained directly on outputs from individual students at the first iteration, filtered with EvalSeg-Net. As explained in more detail later in this Section, Multi-Net is used only to train the unsupervised selection network, EvalSeg-Net.

Technical details: training the students We treat foreground object segmentation as a multidimensional regression problem, where the soft mask given by the unsupervised video segmentation system acts as the desired output. Let $\mathbf{I}$ be the input RGB image (a video frame) and $\mathbf{Y}$ be the corresponding 0–255 valued soft segmentation given by the unsupervised teacher for that particular frame. The goal of our network is to predict a soft segmentation mask $\hat{\mathbf{Y}}$ of width W and height H (where $W = H = 32$ for the basic architecture, $W = H = 128$ for fully convolutional architecture and $W = H = 256$ for U-Net architectures), that approximates as well as possible the mask $\mathbf{Y}$. For each pixel in the output image, we predict a 0–255 value, so that the total difference between $\mathbf{Y}$ and $\hat{\mathbf{Y}}$ is minimized. Thus, given a set of N training examples, let $\mathbf{I}^{(n)}$ be the input image (a video frame), $\hat{\mathbf{Y}}^{(n)}$ be the predicted output mask for $\mathbf{I}^{(n)}$, $\mathbf{Y}^{(n)}$ the soft segmentation mask (corresponding to $\mathbf{I}^{(n)}$) and $\mathbf{w}$ the network parameters. $\mathbf{Y}^{(n)}$ is produced by the video discoverer after processing the video that $\mathbf{I}^{(n)}$ belongs to. Then, our loss is:

$$L(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}\sum_{p=1}^{W\times H}\left(\mathbf{Y}_p^{(n)} - \hat{\mathbf{Y}}_p^{(n)}(\mathbf{w}, \mathbf{I}^{(n)})\right)^2 \quad (1)$$

where $\mathbf{Y}_p^{(n)}$ and $\hat{\mathbf{Y}}_p^{(n)}$ denotes the p-th pixel from $\mathbf{Y}^{(n)}$, respectively $\hat{\mathbf{Y}}^{(n)}$.

We observed that in our tests, the L2 loss performed better than the cross-entropy loss, due to the fact that the soft-masks used as labels have real values, not discrete ones. Also, they are not perfect, so the idea of thresholding them for training does not perform as well as directly predicting their real values. We train our network using the Tensorflow (Abadi et al. 2015) framework with the Adam optimizer (Kingma and Ba 2014). All models are trained end-to-end using a fixed learning rate of 0.001 for 10 epochs. The training time for any given model is about 3–5 days on a Nvidia GeForce GTX 1080 GPU, for the first iteration and about 2 weeks for the second iteration students.

Post-processing The student CNN outputs a $W \times H$ soft mask. In order to fairly compare our models with other methods, we have two different post processing steps: (1) bounding box fitting and (2) segmentation refinement. For fitting a box around the soft mask, we first up-sample the $W \times H$ output to the original size of the image, then threshold the mask (validated on a small subset), determine the connected components and fit a tight box around each of the components. We perform segmentation refinement (point 2) in a single case, on the Object Discovery in Internet Images dataset as also specified in the experiments section. For that, we use the OpenCV implementation of GrabCut (Rother et al. 2004) to refine our soft mask, up-sampled to the original size. In all other tests we use the original output of the networks.

4.2 Teacher (Module B): Unsupervised Object Discovery

There are several methods available for discovering objects and salient regions in images and videos (Borji et al. 2012; Cheng et al. 2015; Hou and Zhang 2007; Jiang et al. 2013; Cucchiara et al. 2003; Barnich and Van Droogenbroeck 2011) with reasonably good performance. More recent methods for foreground objects discovery such as Papazoglou and Ferrari (2013) are both relatively fast and accurate, with runtime around 4 seconds per frame. However, that runtime is still long and prohibitive for training the student CNN that requires millions of images. For that reason we used at the first generation (Iteration 1 of Algorithm 1) for module B in Fig. 1, the VideoPCA algorithm, which is a part of the whole system introduced in Stretcu and Leordeanu (2015). It has lower accuracy than the full system, but it is much faster, running at 50–100 fps. At this speed we can produce one million unsupervised soft segmentations in a reasonable time of about 5–6 h.

VideoPCA The main idea behind VideoPCA is to model the background in video frames with Principal Component Analysis. It finds initial foreground regions as parts of the frames that are not reconstructed well with the PCA model. Foreground objects are smaller than the background, have contrasting appearance and more complex movements. They could be seen as outliers, within the larger background scene. That makes them less likely to be captured well by the first PCA components. Thus, for each frame, an initial soft-mask is produced from an error image, which is the difference between the original image and the PCA reconstruction. These error images are first smoothed with a large Gaussian filter and then thresholded. The binary masks obtained are used to learn color models of foreground and background, based on which individual pixels are classified as belong-

ing to foreground or not. The object masks obtained are further multiplied with a large centered Gaussian, based on the assumption that foreground objects are often closer to the image center. These are the final masks produced by VideoPCA. For more technical details, the reader is invited to consult Stretcu and Leordeanu (2015). In this work, we use the method exactly as found online[1] without any parameter tuning.

Teacher at the next generation At the next iteration of Algorithm 1, VideoPCA (in module B) is replaced by student nets trained at the previous generation. We tested with three different ideas: one is to use a single student network and combine it with the more powerful selection module to form a stronger full teacher pathway (modules B and C). While this approach is very effective and proves the relevance of selection, it is not the most competitive. Using all student nets is always more powerful and this can be done in two ways, as discussed in the previous Section. One possibility is to create Multi-Net ensemble by multiplying their outputs and the other, equally powerful but easier to implement is to use all student nets independently and let each image the possibility to have several output masks, as separate (input image, soft mask) pairs for training the next generation. We prefer the latter approach which, in combination with the EvalSeg-Net network will constitute the full teacher pathway at the second iteration. Next, we present in detail how we perform mask selection and how we train EvalSeg-Net.

4.3 Unsupervised Soft Masks Selection (Module C)

The performance of the student net is influenced by the quality of the soft masks provided as labels by the teacher branch. The cleaner the masks, the more chances the student has to learn to segment well objects in images. VideoPCA tends to produce good results if the object present in the video stands out well against the background scene, in terms of motion and appearance. However, if the object is occluded at some point, does not move w.r.t the scene or has a similar appearance to its background, the resulting soft masks might be poor. In the first generation, we used a simple measure of masks quality to select only the good soft-masks for training the student pathway, based on the following observation: when VideoPCA masks are close to the ground truth, the average of their nonzero values is usually high. Thus, when the discoverer is confident, it is more likely to be right. The average value of non-zero pixels in the soft mask is then used as a score indicator for each segmented frame. Only masks of certain quality according to this indicator are selected and used for training the student nets. This represents module C in Fig. 1 at the first generation of Algorithm 1. While being effective at iteration 1, the simple average value over all pixels cannot capture the goodness of a segmentation at the higher level of overall shape. At the next iterations, we therefore explore new ways to improve it.

Training EvalSeg-Net At the next iterations, we propose an unsupervised way for learning the EvalSeg-Net to estimate segmentation quality. As mentioned previously, Multi-Net provides masks of higher quality as it cancels errors from individual student nets. Thus, we use the cosine similarity between a given individual segmentation and the ensemble Multi-Net mask, as a cost for "goodness" of segmentation. Having this unsupervised segmentation cost we train the EvalSeg-Net deep neural net to predict it. As previously mentioned, this net acts as an automatic mask evaluation procedure, which in subsequent iterations becomes module C in Fig. 1, replacing the simple mask average value used at Iteration 1. Only masks that pass a certain threshold are used for training the student path. As it turns out in experiments, EvalSeg-Net becomes an effective selection procedure (module C) that improves the teacher pathway regardless of the teacher module B used.

The architecture of EvalSeg-Net is similar to LowRes-Net (Fig. 2), with the difference that the input channel containing image derivatives is replaced by the actual soft-segmentation that requires evaluation and it does not have skip connections. After the last fully connected layer (size 512) we add a last one-neuron layer to predict the segmentation quality score, which is a single real valued number.

Let $\mathbf{I}$ be an input RGB image, $\mathbf{S}$ an input soft-mask, $\hat{\mathbf{Y}} = \prod_{i=1}^{5} \hat{\mathbf{Y}}_{N_i}$ be the output of our Multi-Net where $\hat{\mathbf{Y}}_{N_i}$ denotes the output of network N_i. We treat the segmentation "goodness" evaluation task as a regression problem where we want to predict the cosine similarity between $\mathbf{S}$ and $\hat{\mathbf{Y}}$. So, our loss for EvalSeg-Net is defined as follows:

$$L(\mathbf{w}) = \frac{1}{K} \sum_{k=1}^{K} \left(\hat{o}^{(k)}(\mathbf{w}, \mathbf{I}^{(k)}, \mathbf{S}^{(k)}) - \frac{\mathbf{S}^{(k)} . \hat{\mathbf{Y}}^{(k)}}{\left\| \mathbf{S}^{(k)} \right\| \left\| \hat{\mathbf{Y}}^{(k)} \right\|} \right)^2 \tag{2}$$

where K represents the number of training examples and $\hat{o}^{(k)}(\mathbf{w}, \mathbf{I}^{(k)}, \mathbf{S}^{(k)})$ represents the output of EvalSeg-Net for image $\mathbf{I}^{(k)}$ and soft mask $\mathbf{S}^{(k)}$.

Given a certain metric for segmentation evaluation (depending on the learning iteration), we keep only the soft masks above a threshold for each dataset [e.g. VID (Russakovsky et al. 2015), YTO (Prest et al. 2012), YouTube Bounding Boxes (Real et al. 2017)]. In the first iteration, this threshold was obtained by sorting the VideoPCA soft-masks based on their score and keeping only the top 10 percentile, while on the second iteration we validate a threshold ($= 0.8$) on a small dataset and select each mask independently by

[1] https://sites.google.com/site/multipleframesmatching/.

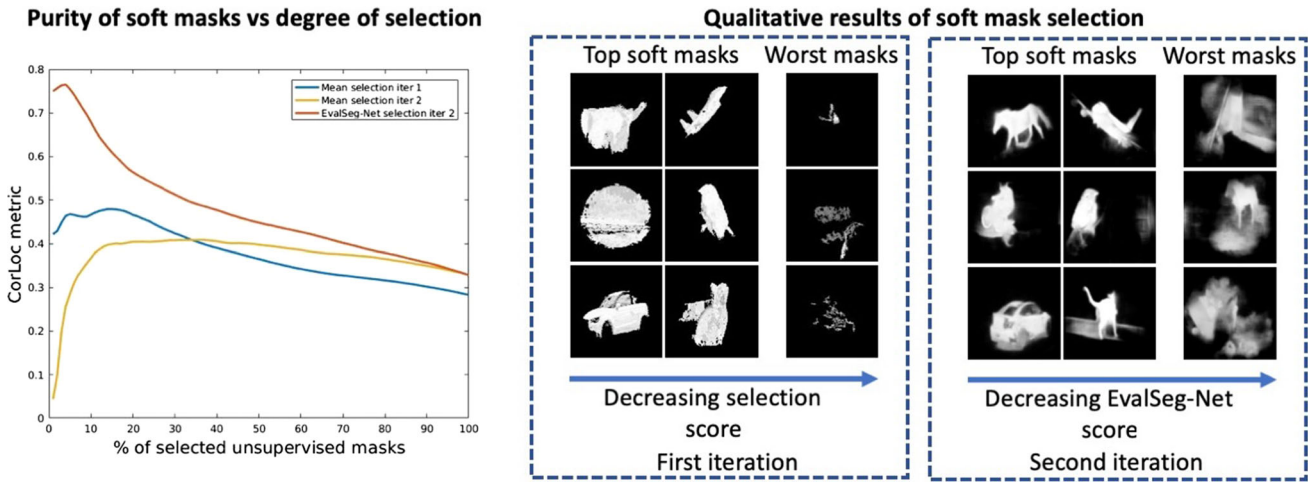

Fig. 3 Quality of soft masks versus degree of selection at module C. When selectivity increases, the true quality of the training frames that pass through selection improves. At the first iteration of Algorithm 1 we select masks using a simple selection procedure based on the mean value of non-zero mask pixels. At the second iteration, we select masks using the much more powerful EvalSeg-Net. The plots are computed using results from the VID dataset, where there is an annotation for each input frame. Note the superior quality of masks selected at the second iteration (red vs. blue lines, in the left plot). We have also compared the simple "mean" based selection procedure used at iteration 1 (yellow line) with EvalSeg-Net used at iteration 2 (red line), on the same soft masks from iteration 2. The EvalSeg-Net is clearly more powerful, which justifies its use at the second iteration when it replaces the very simple "mean" based procedure (Color figure online)

using this threshold on the single value output of EvalSeg-Net.

Mask selection evaluation In Fig. 3 we present the dependency of segmentation performance w.r.t ground truth object boxes (used only for evaluation) versus the percentile p of masks kept after the automatic selection, for each generation. We notice the strong correlation between the percentage of frames kept and the quality of segmentations. It is also evident that the EvalSeg-Net is vastly superior to the simpler procedure used at iteration 1. EvalSeg-Net is able to correctly evaluate soft segmentations even in more complex cases (see Fig. 4).

Even though we can expect to improve the quality of the unsupervised masks by drastically pruning them (e.g. keeping a smaller percentage), the fewer we are left with, the less training data we get, increasing the chance to overfit. We make up for the losses in training data by augmenting the set of training masks and by also enlarging the actual unlabeled training set at the second generation. There is a trade-off between level of selectivity and training data size: the more selective we are about what masks we accept for training, the more videos we need to collect and process through the teacher pathway, to obtain the sufficient training data size.

Data augmentation A drawback of the teacher at the first learning iteration (VideoPCA) is that it can only detect the main object if it is close to the center of the image. The assumption that the foreground is close to the center is often true and indeed helps that method, which has no deep learned

Fig. 4 Qualitative results of the unsupervised EvalSeg-Net used for measuring segmentation "goodness" and filtering bad masks (Module C, iteration 2). For each input image we present five soft-masks candidates (from first iteration students) along with their "goodness" scores given by EvalSeg-Net, in decreasing order of scores. Note the effectiveness of EvalSeg-Net at ranking soft segmentations

knowledge, to produce soft masks with a relatively high precision. Not surprisingly, it often fails when the object is not in the center, therefore its recall is relatively low. Our data augmentation procedure addresses this limitation and can be concisely described as follows: randomly crop patches of the input image, covering 80% of the original image and scale up the patch to the expected input size. This produces slightly larger objects at locations that cover the whole image area, not just the center. As experiments show, the student net is able to see objects at different locations in the image, unlike its raw teacher (VideoPCA at iteration 1), which is strongly biased towards the image center.

At the second generation, the teacher branch is superior at detecting objects at various locations and scales in the image. Therefore, while artificial data augmentation remains useful (as it is usually the case in deep learning), its importance diminishes at the second iteration of learning (Algorithm 1). Adding more unlabeled data helps at both generations up to a point. If more difficult training cases are added, they improve learning only at the second generation, as discussed in the experimental Section (Table 5).

4.4 Implementation Pipeline

Now that we have presented in technical detail all major components of our system, we concisely present the actual steps taken in our experiments, in sequential order, and show how they relate to our general Algorithm 1 for unsupervised learning to segment foreground objects.

1. Run VideoPCA on input images from VID and YouTube Objects datasets (Algorithm 1, Iteration 1, Step 1)
2. Select VideoPCA masks using first generation selection procedure (Algorithm 1, Iteration 1, Step 2)
3. Train first generation student ConvNets on the selected masks, namely LowRes-Net, FConv-Net, BasicU-Net, DilateU-Net and DenseU-Net (Algorithm 1, Iteration 1, Step 3).
4. Create first generation student ensemble Multi-Net by multiplying the outputs of all students and train EvalSeg-Net to predict the similarity between a particular mask and the mask produced by Multi-Net. (Algorithm 1, Iteration 1, Step 4).
5. Add new data from YouTube Bounding Boxes. (Algorithm 1, Iteration 1, Step 5)
6. Return to Step 1, the teacher pathway: predict multiple soft-masks per input image on the enlarged unlabeled video set, using the student nets from Iteration 1 (Module B, Iteration 2), which will be then selected with EvalSeg-Net at Module C. (Algorithm 1, Iteration 2, Step 1)
7. Select only sufficiently good masks evaluated with EvalSeg-Net (Algorithm 1, Iteration 2, Step 2)
8. Train the second generation students on the newly selected masks. We use the same architectures as in Iteration 1 (Algorithm 1, Iteration 2, Step 3)

The method presented in the introduction sections (Algorithm 1) is a general algorithm for unsupervised learning from video to segment objects in single images. It presents a sequence of high level steps followed by different modules for an unsupervised learning system. The modules are complementary to each other and function in tandem, each focusing on a specific aspect of the unsupervised learning process. Thus, we have a module for generating data, where soft-masks are produced. There is a module that selects good quality masks. Then, we have a module for training the next generation students. While, our concept is first presented in high level terms, we also present a specific implementation that represents the first two iterations of the algorithm. While our implementation is costly during training, in terms of storage and computation time, at test time it is very fast.

Computation and storage costs During training, the computation time for passing through the teacher pathway during the first iteration of Algorithm 1 is about 2–3 days: it requires processing data from VID and YTO datasets, including running the VideoPCA module. Afterwards, training the first iteration students, with access to 6 GPUs, takes about 5 days: 6 GPUs are needed for training the 5 different student architectures, since training FConv-Net requires two GPUs in parallel. Next, training the EvalSeg-Net requires 4 additional days on one GPU. At the second iteration, processing the data through the teacher pathway takes about 1 week on 6 GPUs in parallel—it is more costly due to the larger training set from which only a small percent (about 10%) is kept after selection with EvalSeg-Net in order to have in the end 1M data for training. Finally, training the second generation students takes 2 additional weeks. In conclusion, the total computation time required for training, with full access to 6 GPUs is about 5 weeks, when everything is optimized. The total storage cost is about 4TB. At test time the student nets are fast, taking aprox 0.02 s per image, while the ensemble nets take around 0.15 s per image.

5 Experimental Analysis

In the first set of experiments we evaluate the impact of the different components of our system. We experimentally verify that at each iteration the students perform better than their teachers. Then, we test the ability of the system to improve from one generation to the next. We also test the effects of data selection and increasing training data size. Then, we compare the performances of each individual network and their combined ensembles.

In Sect. 5.2, we compare our algorithm to state of the art methods on object discovery in videos and images. We perform tests on three datasets: YouTube Objects (Prest et al. 2012), Object Discovery in Internet Images (Rubinstein et al. 2013) and Pascal-S (Li et al. 2014). In Sect. 5.3, we verify that our unsupervised deep features are also useful on a well-known transfer learning task for object detection on the Pascal VOC2012 dataset (Everingham et al. 2010).

Datasets Unsupervised learning requires large quantities of unlabeled video data. We have chosen for training data, videos from three large datasets: ImageNet VID dataset (Russakovsky et al. 2015), YouTube Objects (YTO) (Prest et al. 2012) and YouTube Bounding Boxes (YouTubeBB) (Real et al. 2017). VID is one of the largest video datasets publicly available, being fully annotated with ground truth bounding boxes. The dataset consists of about 4000 videos, having a total of about 1.2M frames. The videos contain objects that belong to 30 different classes. Each frame could have zero, one or multiple objects annotated. The benchmark challenge associated with this dataset focuses on the supervised object detection and recognition problem, which is different from the one that we tackle here. Our system is not trained to identify different object categories, so we do not report results compared to the state of the art on object class recognition and detection, on this dataset.

YouTube Objects (YTO) is a challenging video dataset with objects undergoing strong changes in appearance, scale and shape, going in and out of occlusion against a varying, often cluttered background. YTO is at its second version now and consists of about 2500 videos, having a total of about 700K frames. It is specifically created for unsupervised object discovery, so we perform comparisons to state of the art on this dataset.

YouTube Bounding Boxes (YTBB or YouTubeBB) is a large scale video dataset, having approximately 240k videos with single-object bounding box annotations. We use a subset of the large number of videos to augment our existing video database. In this dataset there are 23 types of object categories often undergoing strong changes in appearance, scale and shape, making it the most difficult dataset used in our foreground object segmentation setup.

For unsupervised training of our system we used approximately 200k frames (after selection) from videos chosen from each dataset (120k from VID and 80k from YTO), at learning iteration 1—those frames which survived after the data selection module. At the second learning iteration, besides improving the classifier, it is important to have access to larger quantities of new unlabeled data. Therefore, for training the second generation of classifiers we enlarge our training dataset to 1 million soft-masks, as follows: 600k frames from VID + YTO and 400k from the YouTubeBB dataset—those frames which survived after filtering with the EvalSeg-Net data selection module. For experiments presenting results without selection, the frames were randomly chosen from each set, VID, YTO or YouTubeBB, until the total of 1M was reached. We did not add more frames due to heavy computation and storage limitations.

Evaluation metrics We use different kinds of metrics in our experiments, which depend on the specific task that requires either bounding box fitting or fine segmentation:

- *CorLoc*—for evaluating the detection of bounding boxes the most commonly used metric is CorLoc. It is defined as the percentage of images correctly localized according to the PASCAL criterion: $\frac{B_p \cap B_{GT}}{B_p \cup B_{GT}} \geq 0.5$, where B_P is the predicted bounding box and B_{GT} is the ground truth bounding box.
- F-$\beta = \frac{(1-\beta^2) precision \times recall}{\beta^2 \times precision + recall}$ for evaluating the segmentation score on Pascal-S dataset. We use the official evaluation code when reporting results. As in all previous works, we set $\beta^2 = 0.3$.
- *P-J metric* P refers to the precision per pixel, while J is the Jaccard similarity (the intersection over union between the output mask the and ground truth segmentations). We use this metric only on Object Discovery in Internet Images. For computing the reported results we use the official evaluation code.
- *MAE*—Mean Absolute Error is defined as the average pixel-wise difference between the predicted mask and the ground truth. Different from the other metrics, for this metric a lower value is better.
- *mean IoU* score is defined as $\frac{|G \cap Y|}{|G \cup Y|}$ where G represents the ground truth and Y the predicted mask.
- *mAP* represents the mean average precision. It is used when reporting results for the transfer learning experiments on the Pascal VOC 2012 dataset.

5.1 Ablation Study

Student versus Teacher In Fig. 8 we present qualitative results on VID dataset as compared to VideoPCA and between iterations. We can see that the masks produced by VideoPCA are of lower quality, often having holes, non-smooth boundaries and strange shapes. In contrast, the students (at both iterations) learn more general shape and appearance characteristics of objects in images, reminding of the grouping principles governing the basis of visual perception as studied by the Gestalt psychologists (Rock and Palmer 1990) and the more recent work on the concept of "objectness" (Alexe et al. 2010). The object masks produced by the students are simpler, with very few holes, have nicer and smoother shapes and capture well the foreground-background contrast and organization. Another interesting observation is that the students

Table 1 Results of our networks and ensembles on YouTube Objects v1 (Prest et al. 2012) dataset (CorLoc metric) at both iterations (generations)

	LowRes-Net	FConv-Net	DenseU-Net	BasicU-Net	DilateU-Net	Avg (students)	Multi-Net (ensemble)
Iteration 1	62.1	57.6	54.6	59.1	61.8	59.0	65.3
Iteration 2	65.7	64.9	59.5	65.5	66.4	64.4	67.1
Gain	3.6↑	7.3↑	4.9↑	6.4↑	4.6↑	5.4↑	1.8↑

We present the average of CorLoc metric of all 10 classes from YTO dataset for each model and the ensemble, as well as the average of all single models. As it can be seen, at the second generation there is a clear increase in performance for all models. Note that, at the second generation a single model is able to outperform all the methods (single or ensemble) from the first generation. Also, we want to empathize that, the students of the second generation were not trained on the output of Multi-Net, but on the output of individual generation 1 students (Module B), selected with the unsupervised EvalSeg-Net (Module C)

Table 2 Results of our networks and ensemble on Object Discovery in Internet Images (Rubinstein et al. 2013) dataset (CorLoc metric) at both iterations

	LowRes-Net	FConv-Net	DenseU-Net	BasicU-Net	DilateU-Net	Avg (students)	Multi-Net (ensemble)
Iteration 1	85.8	79.8	83.3	86.8	85.6	84.3	85.8
Iteration 2	87.5	84.2	87.9	87.9	88.3	87.2	89.1
Gain	1.7↑	4.4↑	4.6↑	1.1↑	2.7↑	2.9↑	3.3↑

We present the average CorLoc metric of all 3 classes from Object Discovery in Internet Images for each model and the ensemble, as well as the average of all single models. Note that at the second generation there is a clear increase in performance for all methods. The students of the second generation were not trained on the output of the Multi-Net ensemble, but on the output of individual generation 1 students (Module B), selected with the unsupervised EvalSeg-Net (Module C)

Table 3 Results of our networks and ensemble on Pascal-S (Li et al. 2014) dataset (F-β metric) at both iterations

	LowRes-Net	FConv-Net	DenseU-Net	BasicU-Net	DilateU-Net	Avg (students)	Multi-Net (ensemble)
Iteration 1	64.6	51.5	65.2	65.4	65.8	62.5	67.8
Iteration 2	66.8	58.9	69.1	67.9	68.2	66.2	69.6
Gain	2.2↑	7.4↑	3.9↑	2.5↑	2.4↑	3.7↑	1.3↑

We present the F-β metric of each model and the ensemble, as well as the average of all single models. Note that in this case, since we evaluate actual segmentations and not bounding box fitting, nets with higher resolution output perform better (DenseU-Net, BasicU-Net and DilateU-Net). We mention that the students of the second generation were not trained on the output of Multi-Net, but on the output of individual generation 1 students (Module B), selected with the unsupervised EvalSeg-Net (Module C). Again, the ensemble outperforms single models and the second iteration brings a clear gain in every case

are sometimes able to detect multiple objects, a feature that is less commonly achieved by the teacher.

A key fact in our experiments with learning over two generations is that every single module becomes better from one iteration to the next: all individual models and the selector (Module C), all improve and each contributes, in a complementary way, along with the addition of extra unlabelled data, to the overall improvement at the next iteration. The result suggests that we can repeat the process over several iterations and continue to improve. It is also encouraging that the individual nets, which see a single image, are able to generalize and detect objects better than what the initial VideoPCA teacher discovers in videos.

As seen in Tables 1, 2, 3 and Fig. 5 at the second generation we obtain a clear gain over the first, on all experiments and datasets. In Fig. 3, left plot shows the significant improvement of the unsupervised selection network at iteration 2 (EvalSeg-Net) vs the simple selection procedure (based on the mean value of white mask pixels) used at iteration 1.

Our proposed algorithm starts from a completely unsupervised object discoverer in video (VideoPCA) and is able to train neural nets for foreground object segmentation, while improving their accuracy over two generations. It uses the students from iteration 1 as teachers at iteration 2. At the second iteration, it also uses more unlabeled training data and it is better at automatically filtering out poor quality segmentations.

Training data size versus Learning iteration Next we consider the influence of increasing the data size from one iteration to the next vs. learning from a more powerful teacher pathway. In order to better understand the importance of each, we have tested our models at each iteration with two training data sets: a smaller set consisting of 200k images (only from VID + YTO datasets) and a larger dataset formed by

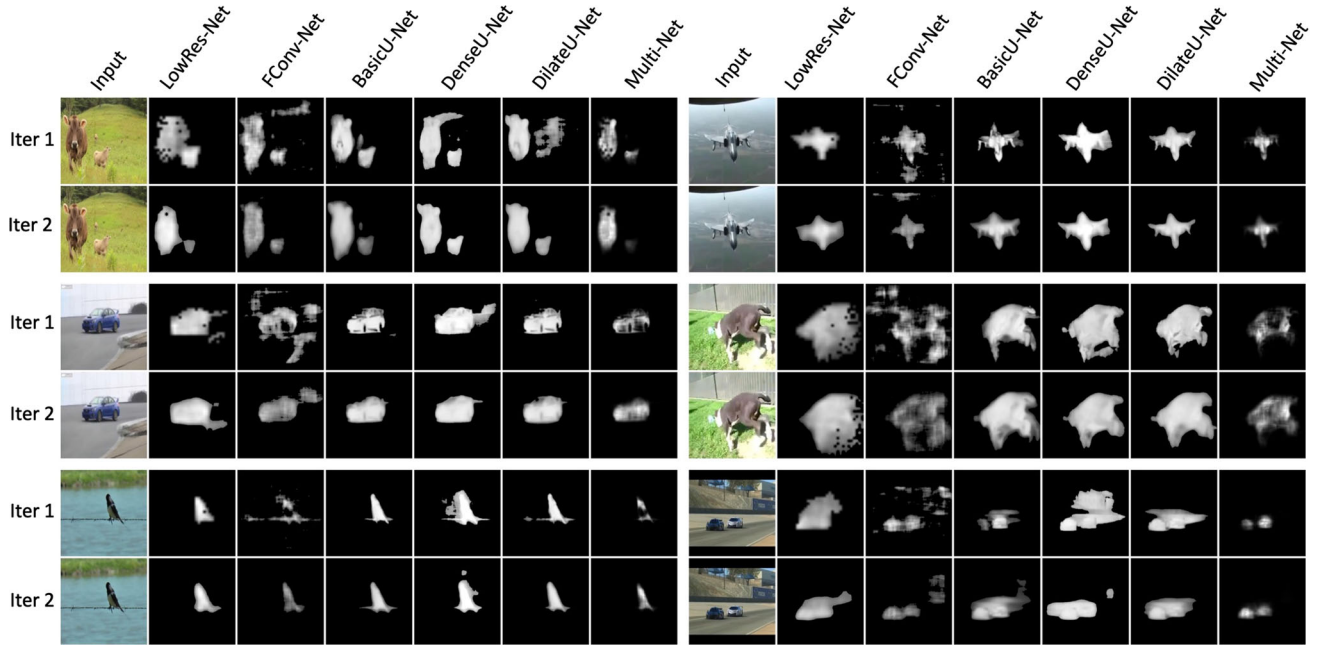

Fig. 5 Visual comparison between models at each iteration (generation). The Multi-Net, shown for comparison, represents the pixel-wise multiplication between the five models. Note the superior masks at the second generation students, with better shapes, fewer holes and sharper edges. Also note the relatively poorer recall of the ensemble Multi-Net, which produces smaller, eroded masks

increasing the dataset size to 1M frames by adding frames from VID, YTO and also from YouTubeBB. Each generation of student nets is trained using the teacher and selection method corresponding to that particular iteration. We present the results in Table 4 (mean CorLoc and standard deviation over five students).

The results are interesting: at Iteration 2, as expected we obtain better accuracy when adding more data (by 1.1%). However, at Iteration 1 adding more data helps initially (as seen in Table 5 when using the LowRes-Net model), but as data becomes more difficult the performance may drop. We have a similar change in performance on tests on image segmentation on the Object Discovery dataset, using the same three training sets on Iteration 1 as in Table 5, where the LowRes-Net model initially improves performance (meanP from 87.7 to 88.4% and meanJ from 61.2 to 62.3%), then it starts losing accuracy as data increases from 200k to 1M frames (meanP goes down to 86.8% and meanJ goes down to 60.7%).

This phenomenon, related to observations in Tokmakov et al. (2016) when working with more difficult images, is probably due to the weaker teacher path at iteration 1. Images in YouTubeBB are significantly more difficult than the ones from the initial 200k frames set. Neither VideoPCA, nor the very simple selection method used along the teacher pathway, at Iteration 1, are powerful enough to cope with these images and produce good training masks. Therefore, even though we have more masks to train on, their quality is poorer and the overall result degrades.

Table 4 Influence of adding more unlabeled data, tested on YTO dataset

	Training data	No fr	Avg CorLoc
Iteration1	VID + YTO	200k	59.0 ± 3.1
Iteration1	VID + YTO + YTBB	1M	58.6 ± 1.2
Iteration2	VID + YTO	200k	63.3 ± 2.6
Iteration2	VID + YTO + YTBB	1M	64.4 ± 2.7

The results represent the average of all five students for each iteration trained with the specified number of frames obtained after selection. As it can be seen adding more data increases the performance in iteration 2 when the teacher and the selection module are better

On the other hand, the second iteration, with a stronger teacher, which is able to produce and select good masks on the more difficult frames from YouTubeBB set, is able to take advantage of the extra large amounts of unlabeled data. It is important to increase the data from one generation to the next in order to avoid simply imitating the teacher of the previous generation. The idea of increasing the data size and complexity in stages, from fewer easy cases to many and more complex ones, is also related to insights from curriculum learning (Bengio et al. 2009).

Data selection versus Teacher Data selection is important (see Figs. 3, 6 and 7 and Table 6). The more selective we are,

Table 5 Influence of adding more unlabeled training data at iteration 1 and iteration 2 for the LowRes-Net student net, evaluated on YTO with the CorLoc metric

	Training data	No fr	CorLoc
LowRes-Net iter1	VID	120k	56.1
LowRes-Net iter1	VID + YTO	200k	62.2
LowRes-Net iter1	VID + YTO + YTBB	1M	58.8
LowRes-Net iter2	VID + YTO	200k	63.7
LowRes-Net iter2	VID + YTO + YTBB	1M	65.7

The number of frames represent the actual number of training frames—that remain after selection. Note that the performance initially increases as data is added for both iterations. When data becomes more difficult only the second iteration LowRes-Net student benefits

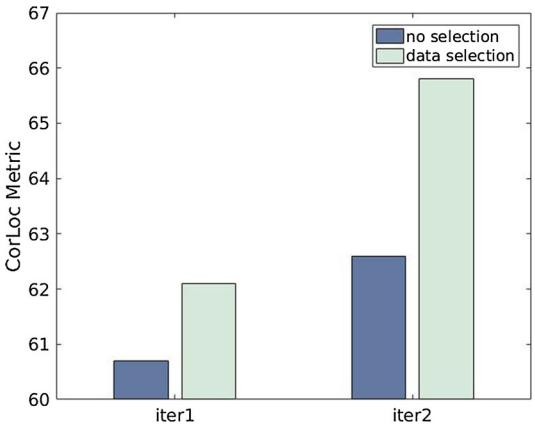

Fig. 6 Impact of data selection for both iterations. Data selection (module C) strongly affects the results at each iteration. The results from iteration 2 with no selection are only slightly better than the ones from iteration 1 with selection. Note that the students trained with selection at iteration 1 become the teacher at the second iteration (module B), without selection. The slight improvement is due to the increase in the training data size. The results represent the average over 10 classes on YouTube Objects using CorLoc percentage metric

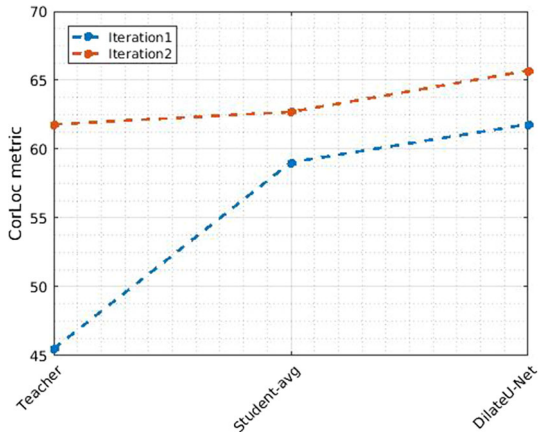

Fig. 7 Comparison across two generations (blue line—first iteration; red line—second iteration) when the individual model DilateU-Net trained at Iteration 1, becomes the teacher for the second generation. DilateU-Net is helped along the teacher pathway at the second iteration, by the EvalSeg-Net selection module, which explains the improvement from one iteration to the next. Note that in this case DilateU-Net improves while being trained, from scratch, on its own good masks allowed to pass by EvalSeg-Net. Also note that individual students (for which we report average values) outperform the teacher on both iterations. The plots are computed over results on the YouTube Objects dataset using the CorLoc metric (percentage) (Color figure online)

Table 6 Different results, averaged over all student nets after being trained at Iteration 2 with different teachers, with or without data selection by EvalSeg-Net

Teacher	Avg CorLoc
Multi-Net without selection	62.49
Multi-Net with selection	63.32
One model (DilateU-Net) with selection	62.69
All models with selection	63.34

In the cases when data selection was used, we present three results, when the selection is applied to the output of the Multi-Net ensemble (second row), to the output of a single model (third row) and the output of our proposed approach (fourth row), where we applied selection to all masks from all students without multiplying them as in Multi-Net (so we obtain five times more masks from single models). We compared these cases "with selection" with the case of learning from an ensemble without data selection (first row). Note that data selection clearly brings an improvement. Even when a single model is used with data selection as a teacher, we could outperform the case of learning from an ensemble without selection

when accepting or rejecting soft-masks used for training, the better the end result. Also note that being more selective means decreasing the training set. There is a trade-off between selectivity and training data size.

We study the impact of data selection (Module C) along the teacher pathway w.r.t to the masks produced by the teacher (Module B), which could be a group of students or a single student net learned at the previous iteration. We want to better understand the roles of the two modules in learning and how they can work best in combination. We did the following experiments: (1) we trained all our student models at iteration 2 with soft-segmentations extracted from Multi-Net created from students trained at iteration one (active module B), but no data selection applied (no module C); (2) then we performed the same experiment as above, but with data selection using EvalSeg-Net (active module C), such that only the masks that passed through selection were used for training; (3) we trained all our models with soft-segmentation masks obtained from a single student, DilateU-Net (active module B) and selected using EvalSeg-Net (active module C); and (4) we used as teacher all student models acting independently, with EvalSeg-Net active, such that for each input image we could have several masks. As stated before this setup is our choice in the final system.

For these experiments we used a small set of 200k images for training. We report average CorLoc on YTO dataset for all 5 students trained at Iteration 2 with different choices of

Table 7 Results on Youtube Objects dataset, versions v1 (Prest et al. 2012)–first eight entries- and v2.2 (Kalogeiton et al. 2016)–last five entries-

Method	Aero	Bird	Boat	Car	Cat	Cow	Dog	Horse	Mbike	Train	Avg	Time	Version
Prest et al. (2012)	51.7	17.5	34.4	34.7	22.3	17.9	13.5	26.7	41.2	25.0	28.5	N/A	v1
Papazoglou and Ferrari (2013)	65.4	67.3	38.9	65.2	46.3	40.2	65.3	48.4	39.0	25.0	50.1	4s	
Jun Koh et al. (2016)	64.3	63.2	73.3	68.9	44.4	62.5	71.4	52.3	**78.6**	23.1	60.2	N/A	
Haller and Leordeanu (2017)	76.3	**71.4**	65.0	58.9	**68.0**	55.9	70.6	33.3	69.7	42.4	61.1	0.35s	
LowRes-Net$_{\text{iter1}}$	77.0	67.5	**77.2**	68.4	54.5	68.3	72.0	**56.7**	44.1	34.9	62.1	0.02s	
LowRes-Net$_{\text{iter2}}$	**83.3**	**71.4**	74.3	69.6	57.4	**80.0**	77.3	**56.7**	50.0	37.2	65.7	0.02s	
DilateU-Net$_{\text{iter2}}$	**83.3**	66.2	**77.2**	**70.9**	63.4	75.0	**80.0**	53.3	50.0	**44.2**	**66.4**	0.02s	
Multi-Net$_{\text{iter2}}$ (ensemble)	*87.4*	*72.7*	*77.2*	64.6	62.4	75.0	*82.7*	*56.7*	52.9	39.5	*67.1*	0.15s	
Haller and Leordeanu (2017)	76.3	**68.5**	**54.5**	50.4	**59.8**	42.4	53.5	30.0	**53.5**	**60.7**	54.9	0.35s	v2.2
LowRes-Net$_{\text{iter1}}$	75.7	56.0	52.7	57.3	46.9	57.0	48.9	44.0	27.2	56.2	52.2	0.02s	
LowRes-Net$_{\text{iter2}}$	79.0	48.2	51.0	**62.1**	46.9	65.7	55.3	**50.6**	36.1	52.4	54.7	0.02s	
DilateU-Net$_{\text{iter2}}$	**84.3**	49.9	52.7	61.4	50.3	**68.8**	**56.4**	47.1	36.1	56.7	**56.4**	0.02s	
Multi-Net$_{\text{iter2}}$ (ensemble)	83.1	53.2	54.3	*63.7*	50.6	*69.2*	*61.0*	*51.1*	37.2	48.7	*57.2*	0.15s	

We achieve state of the art results on both versions. Please note that the baseline LowRes-Net already achieves top results on v1, while being close to the best on v2.2. We present results of the top individual models and the ensemble and also keep the baseline LowRes-Net at both iterations, for reference. Note that complete results on this dataset v1 for all models are also presented in Table 1. For each column we highlight with bold the best model and in italic the cases where the ensemble is better or equal

teacher pathways (Table 6). The results indicate the power of data selection, which could overcome the advantage brought by an ensemble. The ensemble is generally stronger than each individual, as it outputs the mask that represents the multiplication of each student soft segmentation. While its output is more robust to noises, it does not guarantee agreement between student models nor quality. In fact, the final mask obtained by multiplication could be destroyed in the process. For example, in the case when a good mask existed among the students, that would be lost through multiplication. This is a limitation of an ensemble which could be overcome by our approach in which all students are allowed to speak, independently and separately. The EvalSeg-Net, which is a mask selection network trained to predict the agreement among the student models, brings in novel, complementary information and whose output is strongly correlated with the goodness of segmentation (Figs. 3, 6). Such a network could be used to select only good masks. Thus, any teacher, being it a single model or an ensemble, in combination with the selection module is more powerful than without.

The performance of each trained student is boosted through the selection process by 0.8% on average, when the Multi-Net is used as teacher. The relevance of selection could also be seen in the fact that even a simpler teacher with a single model and no ensemble (third row) can be more effective than the ensemble by itself (by 0.2%). The role of selection is again evident when we compare the average results of models at Iteration 1 and those at Iteration 2 when trained by a single model from Iteration 1 (DilateU-Net) with selection, with an increase by 3.7% (compare results in Tables 6 and 7).

Maybe the most conclusive result in favor of selection is when the student model (DilateU-Net) itself improves its own performance when trained (from scratch) on its own outputs from Iteration 1, used as teacher, with selection (third row in Table 6), by no less than 3.89%, increasing the CorLoc on YTO from 61.8% at Iteration 1, to 65.7% at Iteration 2. This improvement can also be seen in Fig. 7 where we presented the results having DilateU-Net acting as a teacher in the second iteration.

The fourth row presents the case when we do not use the Multi-Net ensemble (as teacher), and let all segmentations from all models pass through selection, as explained in Sect. 4.3. As we see, the performance of this approach is almost identical to that of using the ensemble with selection (compare rows 2 and 4 in in Table 6). As previously mentioned, this approach is more effective: if for Multi-Net we need to wait for all 5 models to produce an output until we consider a mask for selection, in this "All models" case we can pass masks through the selection process as they are produced, in parallel, without having to synchronize all five. For this reason, as discussed previously this is our first choice, generally being referred to as "our proposed approach" in the paper and tested in the next Sections.

Analysis of different network architectures As seen in Tables 1, 2 and 3 different network architecture yield different results, while the ensemble always outperforms individual models. Our experiments show that different architectures are better at different tasks. LowRes-Net, for example, per-

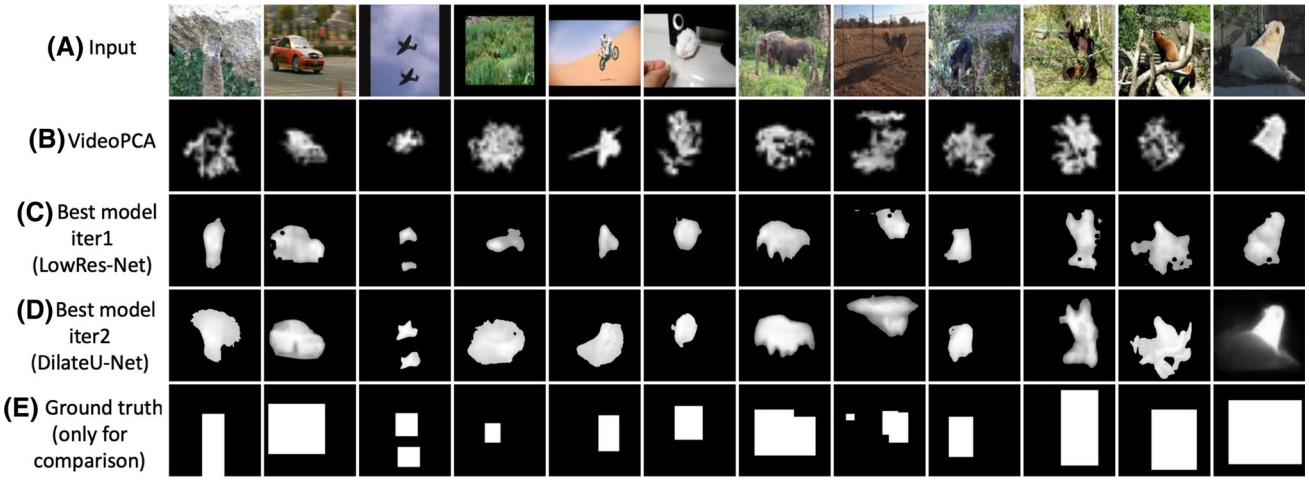

Fig. 8 Qualitative results on the VID dataset (Russakovsky et al. 2015) on input image (**a**) as compared to the iteration 1 teacher–VideoPCA (**b**). For each iteration, we show results of the best individual models (**c, d**), in terms of CorLoc metric. Note the superior quality of our models compared to the VideoPCA. We also present the ground truth bounding boxes (**e**). For more qualitative results please visit our project page https://sites.google.com/view/unsupervisedlearningfromvideo

forms well on the task of box fitting since that does not require a fine sharp object mask. On the other hand, when evaluating the exact segmentation, nets with higher resolution output, such as the ones based on the U-Net design which are more specialized for this task, perform better. Among those, qualitatively, we observed that DenseU-Net produces masks with fewer "holes" when compared to DilateU-Net, which turns out to be the top model for segmentation. The quantitative differences between architectures are shown in Tables 1, 2 and 3, while the qualitative differences can be seen in Figs. 5 and 8.

5.2 Experiments on Foreground Segmentation

Object discovery in video We first performed comparisons with methods specifically designed for object discovery in video. For that, we choose the YouTube Objects dataset and compare it to the best methods on this dataset in the literature (Table 7). Evaluations are conducted on both versions of YouTube Objects dataset, YTOv1 (Prest et al. 2012) and YTOv2.2 (Kalogeiton et al. 2016). On YTOv1 we follow the same experimental setup as (Jun Koh et al. 2016; Prest et al. 2012), by running experiments on all annotated frames from the training split. We have not included in Table 7 the results reported by Stretcu and Leordeanu (2015) because they use a different setup, testing on all videos from YTOv1. It is important to stress out, again, the fact that while the methods presented here for comparison have access to whole video shots, ours only needs a single image at test time. Despite this limitation, our method outperforms the others on 7 out of 10 classes and has the best overall average performance. Note that even our baseline LowRes-Net at the first iteration achieves top performance. The feed-forward CNN processes each image in 0.02 s, being at least one to two orders of magnitude faster than all other methods (see Table 7). We also mention that in all our comparisons, while our system is faster at test time, it takes much longer during its training phase and requires large quantities of unsupervised training data.

Object discovery in images We compare our system against other methods that perform object discovery in images. We use two different datasets for this comparison: Object Discovery in Internet Images and Pascal-S datasets. We report results using metrics that are commonly used for these tasks, as presented at the beginning of the experimental section.

Object Discovery in Internet Images is a representative benchmark for foreground object detection in single images. This set contains internet images and it is annotated with high detail segmentation masks. In order to enable comparison with previous methods, we use the 100 images subsets provided for each of the three categories: airplane, car and horse. The methods evaluated on this dataset in the literature, aim to either discover the bounding box of the main object in a given image or its fine segmentation mask. We evaluate our system on both. Note that different from other works, we do not need a collection of images during test time, since each image can be processed independently by our system. Therefore, unlike other methods, our performance is not affected by the structure of the image collection or the number of classes of interest being present in the collection.

In Table 8 we present the performance of our method as compared to other unsupervised object discovery methods in terms of CorLoc on the Object Discovery dataset. We compare our predicted box against the tight box fitted around the ground-truth segmentation as done in Cho et al. (2015), Tang et al. (2014). Our system can be considered in the mixed class

Table 8 Results on the object discovery in internet images (Rubinstein et al. 2013) dataset (CorLoc metric)

Method	Airplane	Car	Horse	Avg
Kim et al. (2011)	21.95	0.00	16.13	12.69
Joulin et al. (2010)	32.93	66.29	54.84	51.35
Joulin et al. (2012)	57.32	64.04	52.69	58.02
Rubinstein et al. (2013)	74.39	87.64	63.44	75.16
Tang et al. (2014)	71.95	93.26	64.52	76.58
Cho et al. (2015)	82.93	94.38	**75.27**	84.19
Cho et al. (2015) mixed	81.71	94.38	70.97	82.35
LowRes-Net$_{iter1}$	87.80	**95.51**	74.19	85.83
LowRes-Net$_{iter2}$	93.90	93.25	**75.27**	87.47
DilateU-Net$_{iter2}$	**95.12**	**95.51**	74.19	**88.27**
Multi-Net$_{iter2}$ (ensemble)	*97.56*	*95.51*	74.19	*89.09*

The results obtained in the first iteration are further improved in the second one. We present the best single models and ensemble, along with the baseline LowRes-Net at both iterations. Among the single models DilateU-Net is often the best when evaluating box fitting. For each column we highlight with bold the best model and in italic the cases where the ensemble is better or equal

category: it does not depend on the structure of the image collection. It treats each image independently. The performance of the other algorithms degrades as the number of main categories increases in the collection (some are not even tested by their authors on the mixed-class case), which is not the case with our approach.

We obtain state of the art results on all classes, improving by 6% over the method of Cho et al. (2015). When the method in Cho et al. (2015) is allowed to see a collection of images that are limited to a single majority class, its performance improves and it is equal with ours on one class. However, our method has no other information necessary besides the input image, at test time.

We also tested our method on the task of fine foreground object segmentation and compared to the best performers in the literature on the Object Discovery dataset in Table 9. For refining our soft masks we apply the GrabCut method, as it is available in OpenCV. We evaluate based on the same P, J evaluation metric as described by Rubinstein et al. (2013)—the higher P and J, the better. In Figs. 9 and 10 we present some qualitative results for each class. As mentioned previously, these segmentation experiments on Object Discovery in Internet Images are the only ones on which we apply GrabCut as a post-processing step, as also used by all other methods presented in Table 9.

Another important dataset used for the evaluation of a related task, that of salient object detection, is Pascal-S dataset, consisting of 850 images annotated with segmentation mask. As seen from Table 10 we achieve top results on all three metrics against methods that do not use any supervised pre-trained features. Being a foreground object detection method, our approach is usually biased towards the main object in the image—even though it can also detect multiple ones. Images in Pascal-S usually have more objects, so we consider our results very encouraging being close to approaches that use features pre-trained in a supervised manner. Also note that we did not use GrabCut for these experiments.

On single image experiments, our system was trained, as discussed before on other, video datasets (VID, YTO and YTBB). It has not previously seen any of the images in Pascal-S or Object Discovery datasets during training.

5.3 Experiments on Transfer Learning

While the main focus of the paper is unsupervised learning of foreground object segmentation, we also want to test the use-

Table 9 Results on the object discovery in internet images (Rubinstein et al. 2013) dataset using (P, J metric) on segmentation evaluation

	Airplane		Car		Horse	
	P	J	P	J	P	J
Kim et al. (2011)	80.20	7.90	68.85	0.04	75.12	6.43
Joulin et al. (2010)	49.25	15.36	58.70	37.15	63.84	30.16
Joulin et al. (2012)	47.48	11.72	59.20	35.15	64.22	29.53
Rubinstein et al. (2013)	88.04	55.81	85.38	64.42	82.81	51.65
Chen et al. (2014)	90.25	40.33	87.65	64.86	86.16	33.39
LowRes-Net$_{iter1}$	**91.41**	61.37	86.59	70.52	87.07	55.09
LowRes-Net$_{iter2}$	90.70	63.15	**87.00**	73.24	**87.78**	**55.67**
DenseU-Net$_{iter2}$	90.27	**63.37**	86.08	**73.25**	87.40	55.49
Multi-Net$_{iter2}$ (ensemble)	91.39	*65.07*	86.61	73.09	*88.34*	55.53

We present results of the top single model and the ensemble, along with LowRes-Net at both iterations. On the task of fine object segmentation the best individual model tends to be DenseU-Net as also mentioned in the text. Note that we applied GrabCut only on these experiments as a post-processing step, since all methods reported in this Table also used it. For each column we highlight with bold the best model and in italic the cases where the ensemble is better

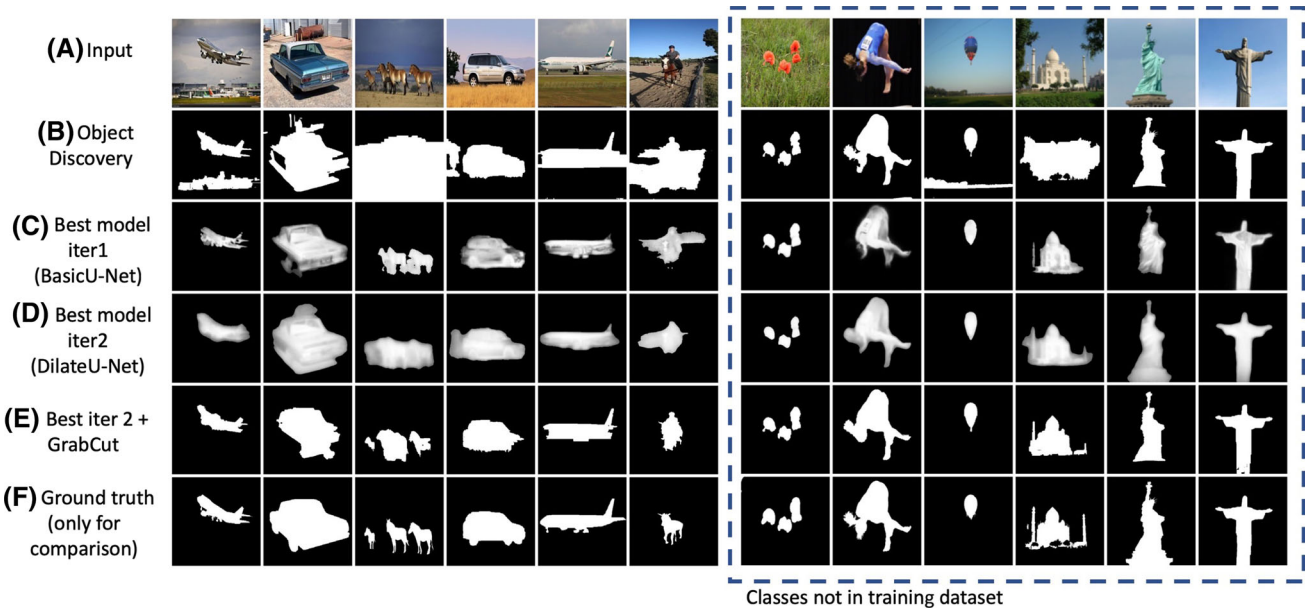

Fig. 9 Qualitative results on the Object Discovery dataset on input image (**a**) as compared to (**b**) Rubinstein et al. (2013). For both iterations, we present the results of the top model (**c, d**), without using GrabCut. We also present the results when GrabCut is used with the top model (**e**) and the ground truth segmentation (**f**). Note that our models are able to segment objects from classes that were not present in the training set (examples on the right side). Also, note that the initial VideoPCA teacher cannot be applied on single images

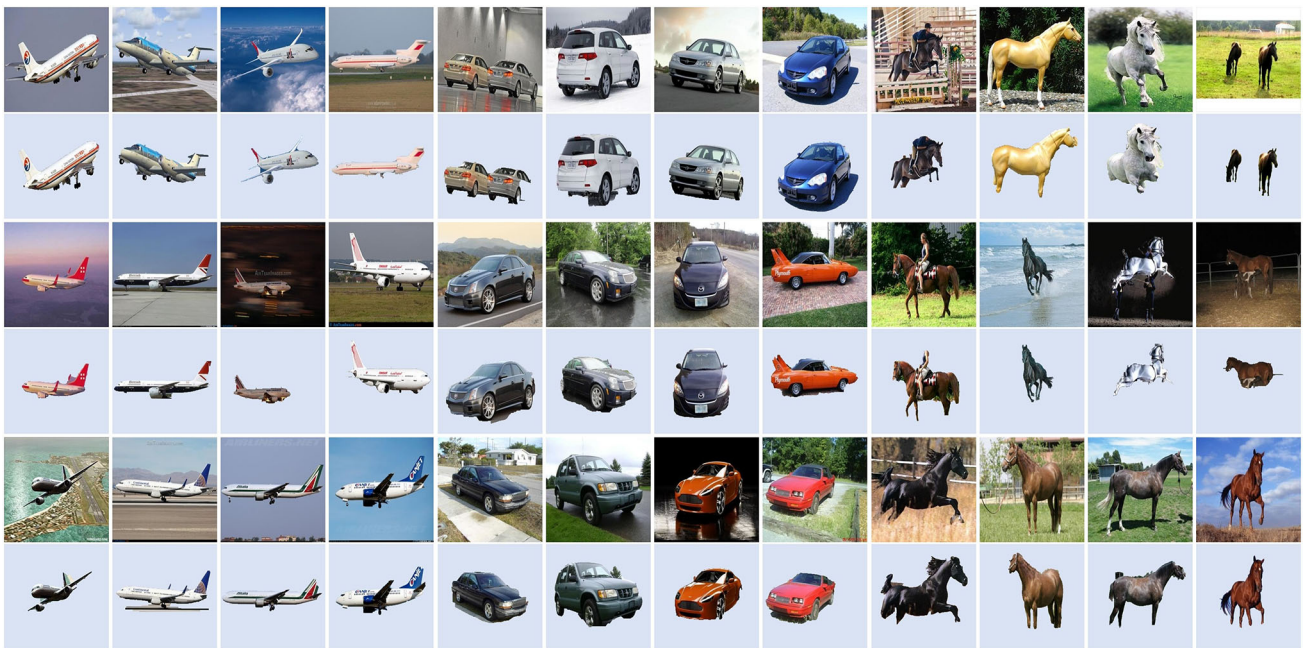

Fig. 10 Qualitative results on the Object Discovery in Internet Images (Rubinstein et al. 2013) dataset. For each example we show the input RGB image and immediately below our segmentation result, with GrabCut post processing for obtaining a hard segmentation. Note that our method produces good quality segmentation results, even in images with cluttered background (Color figure online)

Table 10 Results on the PASCAL-S dataset compared against other unsupervised methods

Method	F_β	MAE	Mean IoU	Pre-trained supervised features?
Wei et al. (2012)	56.2	22.6	41.6	No
Li et al. (2015)	56.8	19.2	42.4	No
Zhu et al. (2014)	60.0	19.7	43.9	No
Yang et al. (2013)	60.7	21.7	43.8	No
Zhang et al. (2015)	60.8	20.2	44.3	No
Tu et al. (2016)	60.9	19.4	45.3	No
Zhang et al. (2017a)	68.0	***14.1***	***54.9***	init VGG
LowRes-Net$_{\text{iter1}}$	64.6	19.6	48.7	No
LowRes-Net$_{\text{iter2}}$	66.8	18.2	**51.7**	No
DenseU-Net$_{\text{iter2}}$	**69.1**	**17.6**	50.9	No
Multi-Net$_{\text{iter2}}$ (ens)	*69.6*	19.5	*52.3*	No

For MAE score lower is better, while for F_β and mean IoU higher is better. We reported max F_β, min MAE and max mean IoU for every method. In bold we presented the top results when no supervised pre-trained features were used per single model and in italic the cases were the ensemble is better. We highlight, in bold italic, the cases where a method, that uses supervised features, has better results

Table 11 Comparison with state of the art on VOC 2012 Everingham et al. (2010) using Fast R-CNN initialized with different methods

Method	All	> c1	> c2	> c3	> c4	> c5
Agrawal et al. (2015)	37.4	36.9	34.4	28.9	24.1	17.1
Pathak et al. (2016)	39.1	36.4	34.1	29.4	24.8	13.4
Krähenbühl et al. (2015)	42.8	42.2	40.3	37.1	32.4	26.0
Owens et al. (2016)	42.9	42.3	40.6	37.1	32.0	26.5
Wang and Gupta (2015)	43.5	44.6	44.6	44.2	41.5	35.7
Zhang et al. (2017b)	43.8	45.6	45.6	46.1	44.1	37.6
Zhang et al. (2016)	44.5	44.9	44.7	44.4	42.6	38.0
Donahue et al. (2016)	44.9	44.6	44.7	42.4	38.4	29.4
Pathak et al. (2017)	48.6	48.2	**48.3**	**47.0**	**45.8**	**40.3**
Doersch et al. (2015)	**49.9**	**48.8**	44.4	44.3	42.1	33.2
Ours—AlexNet-Seg(MO)	46.8	46.2	43.9	41.0	36.5	29.9

The numbers represent the mAP for the validation split of VOC 2012. In bold, we highlight the best performing method. Each column represents to what extent the network was fine-tuned, so for example >c1 represents that all layers after the first convolution are fine-tuned, while 'All' represents the case where all the layers are fine-tuned

Table 12 Comparison with Pathak et al. (2017) on unsupervised object discovery

Method	YTO	Obj-Disc	Pascal-S	Training data
Pathak et al. (2017)	39.4	82.1	64.7	YFCC100m
AlexNet-Seg(SO)	62.7	86.4	67.4	Our data
AlexNet-Seg(SO)	56.6	83.6	65.5	YFCC100m
AlexNet-Seg(MO)	54.7	86.0	65.9	Our data
AlexNet-Seg(MO)	58.3	84.4	65.2	YFCC100m
AlexNet-Seg(MO)	58.1	85.2	66.1	Both

For the YTO and Obj-Disc datasets we report the CorLoc metric and for Pascal-S the F_β metric. With SO we represent our AlexNet-Seg student trained from a single object (SO) teacher, while with MO, we represent our student trained from a multiple object (MO) teacher—as explained in the text. As it can be seen, our proposed multi object scheme affects in a negative way the results on foreground object segmentation. This happens, because these datasets have mainly one single object

fulness of our features in a transfer learning setup, namely for the task of multiple object detection. For this purpose, we follow the well known experimental setup used in the recent transfer learning literature, in which an AlexNet-like (Krizhevsky et al. 2012) network, initialized in an unsupervised way, is fine-tuned on supervised object detection within the Fast R-CNN framework (Girshick 2015).[2] We closely follow the work of Pathak et al. (2017), with code, documentation and training data available online, which, as mentioned in the related work section, also starts by learning from videos to segment objects in single images in an unsupervised fashion. In these experiments, we adapted in the same way the last part of AlexNet in order to produce a soft segmentation of the image (instead of an output class). We used the same base architecture as the methods we compared against, to make sure that the results come from the learned features, not from the architecture we used.

Initial unsupervised training for object segmentation We used the adapted AlexNet-based model (which we term AlexNet-Seg) described in this section as a student in our unsupervised learning framework at iteration 2. Thus, the AlexNet-Seg will be trained by the unsupervised teacher pathway at our Iteration 2—in this case, the teacher will be a single network, namely DilateU-Net at module B, combined with the EvalSeg-Net mask selector, at module C. In order to see how the actual training data influences the final transfer learning outcome, we experimented with both our data and the data used by Pathak et al. (2017) which is obtained from YFCC100m (Thomee et al. 2015) dataset, having 1.6M frames. The results are presented in Table 13.

As it is, our unsupervised learning system prefers to segment main, foreground objects in images and it is less versatile on segmenting complex images containing many objects. Since the images in Pascal VOC2012 contain complex scenes with multiple objects and the final transfer learning task is of multiple object detection, we also tested the case when we adapted our system to better cope with multiple objects. For that, we divide each training image into 5 large patches (a grid with one image at each corner and one in the middle, each crop being about 60% of the original size for both dimensions) which we pass through the teacher pathway at Iteration 2. The results are combined into a single image, by superimposing the soft masks and taking the maximum over all, at each location in the original image. Thus, we obtain soft segmentations that better capture multiple objects in the input image. Note that the original image passed through the teacher pathway without the 5-point grid division is referred to as the Single Object (SO) teacher in Tables 12 and 13 and Fig. 11, while the 5-grid version just described is referred to as the Multiple Object (MO) teacher in the same Tables and Figures. We train the AlexNet-Seg student on these multiple object soft-segs. We thus transfer knowledge from our unsupervised student models to AlexNet-Seg and prepare it for the next task, of supervised object detection.

Table 13 Comparison between different types of training images and training data for the AlexNet-Seg student we used

Model	VOC2012 (mAP)	Training data
AlexNet-Seg(SO)	46.2	Our data
AlexNet-Seg(SO)	45.7	YFCC100m
AlexNet-Seg(MO)	46.1	Our data
AlexNet-Seg(MO)	46.8	YFCC100m
AlexNet-Seg(MO)	46.8	Both

With SO we represent our AlexNet-Seg student trained from a single object (SO) teacher, while with MO, we represent our student trained from a multiple object (MO) teacher—as explained in the text. We show how the training data affects the transfer learning results. The numbers represent the mAP metric for the Pascal VOC 2012 validation split when finetuning the whole network. Please note that the proposed multiple object approach brings only a small improvement. Also, on these transfer learning tests, the larger training dataset—YFCC100m, tends to bring improved results, by a small margin

Transferring to object detection As the other methods we compare to, we conduct transfer learning experiments on the Pascal VOC 2012 (Everingham et al. 2010) dataset. We train on the *train* split of VOC 2012 and we report our results on the *validation* split. We also use multi scale training and testing and remove difficult objects during training. We report the comparisons results in Table 11. We see that the unsupervised knowledge learned by our approach is indeed useful for transfer learning, as our results are in the top 3 among current published methods. This is interesting, as our unsupervised learning algorithm is mainly designed for foreground object segmentation, not classification.

Foreground segmentation versus Multiple object detection The transfer learning task, which we test our approach on, is both about localization (detection) and classification. In this context, as already discussed in the introduction section, the work of Pathak et al. (2017) is most related to ours in the sense that they also learn from video to segment objects in single images in an unsupervised manner. They do so by using a teacher that produces soft masks from optical flow in video. Beyond the theoretical connection between the two works, we wanted to better understand how the two approaches relate on actual foreground segmentation experiments. Their method generally produce masks that cover larger areas in the image than ours and are better suited for transfer learning experiments, as results show.

On the other hand, when tested on foreground segmentation tasks, our approach, in turn, seems to yield better results

[2] https://github.com/rbgirshick/py-faster-rcnn.

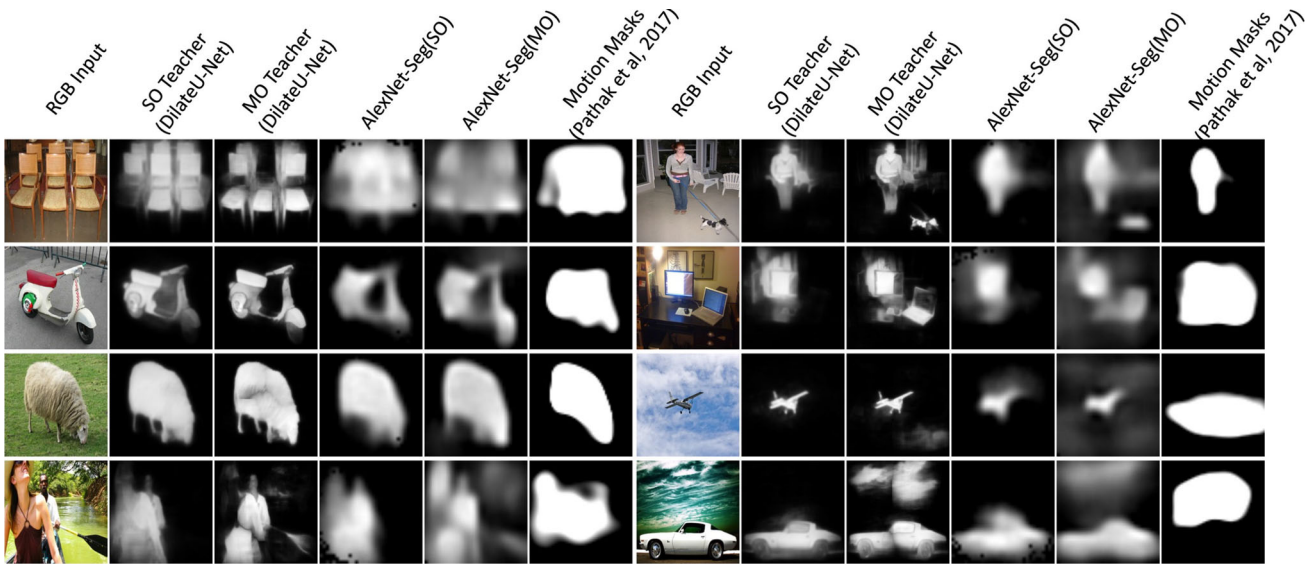

Fig. 11 Representative visual results on Pascal VOC 2012. We show the output of the unsupervised teacher pathway used in transfer learning for both the original case (SO—DilateU-Net) and for the multiple objects scenario (MO—combined DilateU-Net outputs combined on a 5-grid), the trained students for both cases (AlexNet-Seg(SO) and AlexNet-Seg(MO)), as well as the output of Pathak et al. (2017)

(see Table 12). The results we obtained are in agreement with observations made by the authors when testing their method on detecting main objects in single frames against human annotations (e.g. Precision: 29.9, Recall: 59.3, Mean IoU: 24.8). Their high recall and lower precision agree with our observations that their segmentation covers larger parts of the image, while ours provides sharper and smaller masks. This observation lead us towards extracting large crops on a 5-point grid and combining the results (termed AlexNet-Seg MO). As seen in experiments, taking multiple outputs over the grid eventually brought a relatively small improvement of 0.6% (see Table 13). In Fig. 11 we present qualitative results of our DilatedU-Net teacher for the AlexNet-Seg student trained with a single foreground object detected (termed SO Teacher) and our 5-grid multiple objects (termed MO Teacher) segmentation result. We also present the results of our AlexNet-Seg student on both cases as well as the outputs of Pathak et al. (2017) for comparison. While our method has better results on the task of segmentation, their method is more suited for transfer learning experiments. We suspect that their larger masks (Fig. 11), with lower precision but relatively high recall and high confidence values, could be more flexible and less conservative for the final transfer learning stage where multiple objects need to be detected over the whole image. At the same time ours is specialized in obtaining generally sharper and better quality foreground segmentation masks. Overall, the transfer learning experiments show that our approach is suited for such task, as we obtain a performance that is in the top three among the state of the art methods using the same experimental setup.

5.4 Concluding Remarks on Experiments

One of the interesting conclusions in our experimental analysis is that the system is able to improve its performance from iteration 1 to iteration 2. There are several factors that are involved in this outcome, which we believe are related through the following relationship: (1) Multiple students of diverse structures ensure diversity and somewhat independent mistakes; (2) In turn, point (1) makes possible the unsupervised training of a mask selection module that learns to predict agreements; (3) thus, the selection module at 2) becomes a good mask evaluation network; (4) once that evaluation network (from 3) is available, we can then add larger and potentially more complex data to select a larger set with good object masks of more interesting cases at the next iteration; (5) finally, (4) ensures the improvement at the next iteration and now we could return to point (1).

6 Short Discussion on Unsupervised Learning

The ultimate goal of unsupervised learning might not be about matching the performance of the supervised case but rather about reaching beyond the capabilities of the classical supervised scenario. An unsupervised system should be able to learn and recognize different object classes, such as animals, plants and man-made objects, as they evolve and change over time, from the past and into the unknown future. It should also be able to learn about new classes that might be formed, in relation to others, maybe known ones. We see

this case as fundamentally different from the supervised one in which the classifier is forced to learn from a distribution of samples that is fixed and limited to a specific period of time—that when the human labeling was performed. Therefore, in the supervised learning paradigm a car from the future, should not be classified as car, because it is not a car, according to the supervised distribution of cars given at present training time, when human annotations are collected. On the other hand, a system that learns by itself should be able to track how cars have been changing in time and recognize such objects as "cars"—with no step by step human intervention.

Current unsupervised learning methods might still not be able to learn profound semantic information (Bau et al. 2017), but the ability to learn to segment foreground objects in an unsupervised fashion constitutes evidence that we are moving in the right direction. In order to understand and learn about semantic classes, the system would need to learn by itself about how such objects interact with each other and what role they play within the larger spatiotemporal story. While our unsupervised methods are still far from reaching this level of interpretation, the ability to learn about and detect objects that constitute the foreground within their local spatial context could constitute an important building block. It is an element that could be used to further learn about more complex interactions and behaviour in both space and time.

From the larger spatiotemporal perspective, unsupervised learning is about continuous learning and adaptation to huge quantities of data that are perpetually changing. Human annotation is extremely limited in an ocean of data and not able to provide the so called "ground truth" information continuously. Therefore, unsupervised learning, and especially its weaker version—learning from large quantities of data with minimal human intervention—will soon become a core part, larger than the supervised one, in the future of artificial intelligence.

7 Conclusions and Future Work

In this article, we present a novel and effective approach to learning from large collections of images and videos, in an unsupervised fashion, to segment foreground objects in single images. We present a relatively general algorithm for this task, which offers the possibility of learning several generations of students and teachers. We demonstrate in practice that the system improves its performance over the course of two generations. We also test the impact of the different system components on performance and show state of the art results on three different datasets, while also showing top performance on challenging transfer learning experiments. Our system is one of the first in the literature that learns to detect and segment foreground objects in images in an unsupervised fashion, with no pre-trained features given or manual labeling, while requiring only a single image at test time.

The convolutional networks trained along the student pathway are able to learn general "objectness" characteristics, which include good form, closure, smooth contours, as well as contrast with the background. What the simpler initial VideoPCA teacher discovers over time, the deep, complex student is able to learn across several layers of image features at different levels of abstraction. Our results on transfer learning experiments are also encouraging and show additional cases in which such a system could be useful. In future work we plan to further grow our computational and storage capabilities to demonstrate the power of our unsupervised learning algorithm along many generations of student and teacher networks. We believe that our approach, tested here in extensive experiments, could bring a valuable contribution to computer vision research.

Acknowledgements This work was supported by UEFISCDI, under Projects PN-III-P4-ID-ERC-2016-0007, PN-III-P2-2.1-PED-2016-1842, PN-III-P1-1.1-TE-2016-2182 and PN-III-P1-1.2-PCCDI-2017-0734.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2015). Tensorflow: Large-scale machine learning on heterogeneous systems. Software available from https://www.tensorflow.org/

Agrawal, P., Carreira, J., & Malik, J. (2015). Learning to see by moving. In *Proceedings of the IEEE international conference on computer vision* (pp. 37–45).

Alexe, B., Deselaers, T., & Ferrari, V. (2010). What is an object? In *CVPR*.

Barnich, O., & Van Droogenbroeck, M. (2011). Vibe: A universal background subtraction algorithm for video sequences. *IEEE Transactions on Image processing*, *20*(6), 1709–1724.

Bau, D., Zhou, B., Khosla, A., Oliva, A., & Torralba, A. (2017). Network dissection: Quantifying interpretability of deep visual representations. In *International conference on computer vision and pattern recognition (CVPR)*.

Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning* (pp. 41–48). ACM.

Borji, A., Sihite, D., & Itti, L. (2012). Salient object detection: a benchmark. In *ECCV*.

Chen, X., Shrivastava, A., & Gupta, A. (2014). Enriching visual knowledge bases via object discovery and segmentation. In *CVPR*.

Cheng, J., Tsai, Y. H., Wang, S., & Yang, M. H. (2017). Segflow: Joint learning for video object segmentation and optical flow. In *The IEEE international conference on computer vision (ICCV)*.

Cheng, M., Mitra, N., Huang, X., Torr, P., & Hu, S. (2015). Global contrast based salient region detection. *PAMI*, *37*(3), 569–582.
Cho, M., Kwak, S., Schmid, C., & Ponce, J. (2015). Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *CVPR*.
Croitoru, I., Bogolin, S.V., & Leordeanu, M. (2017). Unsupervised learning from video to detect foreground objects in single images. In *2017 IEEE international conference on computer vision (ICCV)* (pp. 4345–4353). IEEE.
Cucchiara, R., Grana, C., Piccardi, M., & Prati, A. (2003). Detecting moving objects, ghosts, and shadows in video streams. *PAMI*, *25*(10), 1337–1342.
Deselaers, T., Alexe, B., & Ferrari, V. (2012). Weakly supervised localization and learning with generic knowledge. *IJCV*, *100*(3), 275–293.
Doersch, C., Gupta, A., & Efros, A. A. (2015). Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision* (pp. 1422–1430).
Donahue, J., Krähenbühl, P., & Darrell, T. (2016). Adversarial feature learning. arXiv preprint arXiv:1605.09782.
Dutt Jain, S., Xiong, B., & Grauman, K. (2017). Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.
Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, *111*(1), 98–136.
Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The PASCAL visual object classes (voc) challenge. *International Journal of Computer Vision*, *88*(2), 303–338.
Finn, C., Goodfellow, I., & Levine, S. (2016). Unsupervised learning for physical interaction through video prediction. In *Advances in neural information processing systems* (pp. 64–72)
Girshick, R. (2015). Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440–1448).
Goroshin, R., Mathieu, M. F., & LeCun, Y. (2015). Learning to linearize under uncertainty. In *Advances in neural information processing systems* (pp. 1234–1242).
Haller, E., & Leordeanu, M. (2017). Unsupervised object segmentation in video by efficient selection of highly probable positive features. In *The IEEE international conference on computer vision (ICCV)*.
Hou, X., & Zhang, L. (2007). Saliency detection: A spectral residual approach. In *CVPR*.
Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys*, *31*(3), 264–323.
Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., & Bengio, Y. (2017). The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)* (pp. 1175–1183). IEEE.
Jiang, H., Wang, J., Yuan, Z., Wu, Y., Zheng, N., & Li, S. (2013). Salient object detection: A discriminative regional feature integration approach. In *CVPR*.
Joulin, A., Bach, F., & Ponce, J. (2010). Discriminative clustering for image co-segmentation. In *CVPR*.
Joulin, A., Bach, F., & Ponce, J. (2012). Multi-class cosegmentation. In *CVPR*.
Joulin, A., Tang, K., & Fei-Fei, L. (2014). Efficient image and video co-localization with Frank–Wolfe algorithm. In *ECCV*.
Jun Koh, Y., Jang, W.D., & Kim, C. S. (2016). Pod: Discovering primary objects in videos based on evolutionary refinement of object recurrence, background, and primary object models. In *CVPR*.
Kalogeiton, V., Ferrari, V., & Schmid, C. (2016). Analysing domain shift factors between videos and images for object detection. *PAMI*, *38*(11), 2327–2334.
Khoreva, A., Benenson, R., Hosang, J.H., Hein, M., & Schiele, B. (2017). Simple does it: Weakly supervised instance and semantic segmentation. In *CVPR* (Vol. 1, p. 3).
Kim, G., Xing, E., Fei-Fei, L., & Kanade, T. (2011). Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *ICCV*.
Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
Krähenbühl, P., Doersch, C., Donahue, J., & Darrell, T. (2015). Data-dependent initializations of convolutional neural networks. arXiv preprint arXiv:1511.06856.
Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).
Kuettel, D., Guillaumin, M., & Ferrari, V. (2012). Segmentation propagation in imagenet. In *ECCV*.
Larsson, G., Maire, M., & Shakhnarovich, G. (2016). Learning representations for automatic colorization. In *European Conference on Computer Vision* (pp. 577–593). Springer, Berlin.
Lee, H. Y., Huang, J. B., Singh, M., & Yang, M. H. (2017). Unsupervised representation learning by sorting sequences. In *2017 IEEE international conference on computer vision (ICCV)* (pp. 667–676). IEEE.
Lee, Y. J., Kim, J., & Grauman, K. (2011). Key-segments for video object segmentation. In *2011 IEEE international conference on computer vision (ICCV)* (pp. 1995–2002). IEEE.
Leordeanu, M., Collins, R., & Hebert, M. (2005). Unsupervised learning of object features from video sequences. In *CVPR*.
Leordeanu, M., Sukthankar, R., & Hebert, M. (2012). Unsupervised learning for graph matching. *International Journal of Computer Vision*, *96*, 28–45.
Li, D., Hung, W. C., Huang, J. B., Wang, S., Ahuja, N., & Yang, M. H. (2016). Unsupervised visual representation learning by graph-based consistent constraints. In *ECCV*.
Li, N., Sun, B., & Yu, J. (2015). A weighted sparse coding framework for saliency detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5216–5223).
Li, Y., Hou, X., Koch, C., Rehg, J. M., & Yuille, A. L. (2014). *The secrets of salient object segmentation*. Atlanta: Georgia Institute of Technology.
Liu, D., Chen, T. (2007) A topic-motion model for unsupervised video object discovery. In *CVPR*.
Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431–3440).
Misra, I., Zitnick, C. L., & Hebert, M. (2016). Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*.
Nguyen, M., Torresani, L., La Torre, F. D., & Rother, C. (2009). Weakly supervised discriminative localization and classification: A joint learning process. In *CVPR*.
Noroozi, M., & Favaro, P. (2016). Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision* (pp. 69–84). Springer, Berlin.
Owens, A., Wu, J., McDermott, J. H., Freeman, W. T., & Torralba, A. (2016). Ambient sound provides supervision for visual learning. In *European conference on computer vision* (pp. 801–816). Springer, Berlin.
Papazoglou, A., & Ferrari, V. (2013). Fast object segmentation in unconstrained video. In *ICCV*.
Parikh, D., & Chen, T. (2007). Unsupervised identification of multiple objects of interest from multiple images: Discover. In *Asian conference on computer vision*.

Pathak, D., Girshick, R., Dollar, P., Darrell, T., & Hariharan, B. (2017). Learning features by watching objects move. In *CVPR*.

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2536–2544).

Pinheiro, P.O., Lin, T. Y., Collobert, R., & Dollár, P. (2016). Learning to refine object segments. In *ECCV*.

Prest, A., Leistner, C., Civera, J., Schmid, C., & Ferrari, V. (2012). Learning object class detectors from weakly annotated video. In *CVPR* (pp. 3282–3289). IEEE.

Radenović, F., Tolias, G., & Chum, O. (2016). CNN image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *ECCV*.

Raiko, T., Valpola, H., & LeCun, Y. (2012). Deep learning made easier by linear transformations in perceptrons. In *AISTATS* (Vol. 22, pp. 924–932).

Raina, R., Battle, A., Lee, H., Packer, B., & Ng, A. Y. (2007). Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on machine learning* (pp. 759–766). ACM.

Real, E., Shlens, J., Mazzocchi, S., Pan, X., & Vanhoucke, V. (2017). Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 7464–7473). IEEE.

Rochan, M., & Wang, Y. (2014). Efficient object localization and segmentation in weakly labeled videos. In *Advances in visual computing* (pp. 172–181). Springer, Berlin.

Rock, I., & Palmer, S. (1990). Gestalt psychology. *Scientific American*, *263*, 84–90.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention* (pp. 234–241). Springer, Berlin.

Rother, C., Kolmogorov, V., & Blake, A. (2004). Grabcut: Interactive foreground extraction using iterated graph cuts. In *ACM transactions on graphics* (Vol. 23, pp. 309–314).

Rubinstein, M., Joulin, A., Kopf, J., & Liu, C. (2013). Unsupervised joint object discovery and segmentation in internet images. In *CVPR*.

Rubio, J., Serrat, J., & López, A. (2012). Video co-segmentation. In *ACCV*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *IJCV*, *115*(3).

Siva, P., Russell, C., Xiang, T., & Agapito, L. (2013). Looking beyond the image: Unsupervised learning for object saliency and detection. In *CVPR*.

Sivic, J., Russell, B., Efros, A., Zisserman, A., & Freeman, W. (2005). Discovering objects and their location in images. In *ICCV*.

Stretcu, O., & Leordeanu, M. (2015). Multiple frames matching for object discovery in video. In *BMVC*.

Tang, K., Joulin, A., Li, L. J., & Fei-Fei, L. (2014). Co-localization in real-world images. In *CVPR*.

Thomee, B., Shamma, D. A., Friedland, G., Elizalde, B., Ni, K., Poland, D., Borth, D., & Li, L. J. (2015). Yfcc100m: The new data in multimedia research. arXiv preprint arXiv:1503.01817.

Tokmakov, P., Alahari, K., & Schmid, C. (2016). Learning semantic segmentation with weakly-annotated videos. In *ECCV* (Vol. 1, p. 6).

Tokmakov, P., Alahari, K., & Schmid, C. (2017). Learning motion patterns in videos. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.

Tu, W. C., He, S., Yang, Q., & Chien, S. Y. (2016). Real-time salient object detection with a minimum spanning tree. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2334–2342).

Vicente, S., Rother, C., & Kolmogorov, V. (2011). Object cosegmentation. In *CVPR*.

Wang, X., & Gupta, A. (2015). Unsupervised learning of visual representations using videos. In *The IEEE international conference on computer vision (ICCV)*.

Wei, Y., Wen, F., Zhu, W., & Sun, J. (2012). Geodesic saliency using background priors. In *European conference on computer vision* (pp. 29–42). Springer, Berlin.

Xue, T., Wu, J., Bouman, K., & Freeman, B. (2016). Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances in neural information processing systems* (pp. 91–99).

Yang, C., Zhang, L., Lu, H., Ruan, X., & Yang, M. H. (2013). Saliency detection via graph-based manifold ranking. In *2013 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 3166–3173). IEEE.

Yu, F., & Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122.

Zhang, D., Han, J., & Zhang, Y. (2017a). Supervision by fusion: Towards unsupervised learning of deep salient object detector. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4048–4056).

Zhang, J., Sclaroff, S., Lin, Z., Shen, X., Price, B., & Mech, R. (2015). Minimum barrier salient object detection at 80 fps. In *Proceedings of the IEEE international conference on computer vision* (pp. 1404–1412).

Zhang, R., Isola, P., & Efros, A. A. (2016). Colorful image colorization. In *European conference on computer vision* (pp. 649–666). Springer, Berlin.

Zhang, R., Isola, P., & Efros, A. A. (2017b). Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR* (Vol. 1, p. 5).

Zhu, W., Liang, S., Wei, Y., & Sun, J. (2014). Saliency optimization from robust background detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2814–2821).