Meta Learning Deep Visual Words for Fast Video Object Segmentation

Harkirat Singh Behl* Mohammad Najafi* Philip H.S. Torr
University of Oxford
{harkirat, monaj, phst}@robots.ox.ac.uk

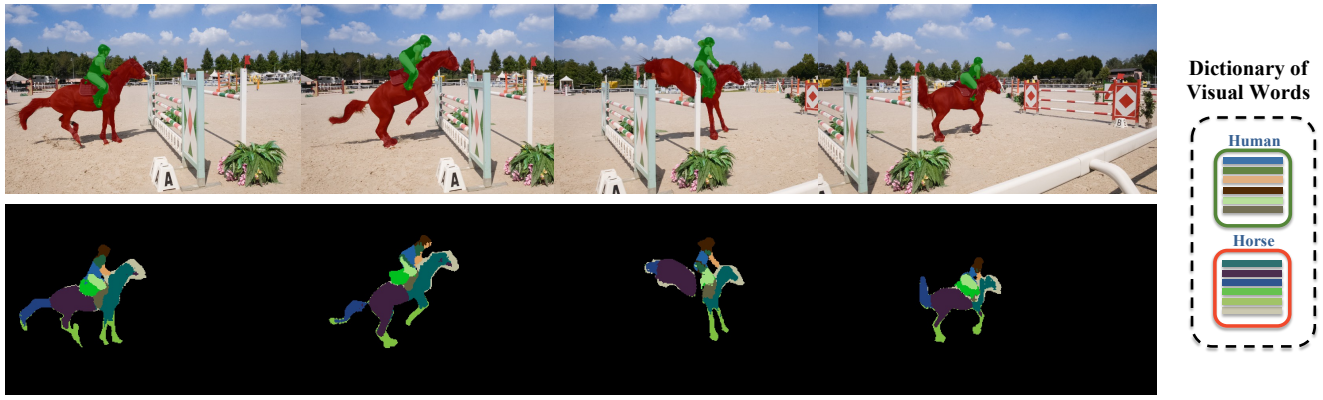

Figure 1: **Video object segmentation by unsupervised learning of a dictionary of deep visual words as object representation.** The proposed method meta learns a dictionary of deep visual words for representing object parts in the video (bottom), in order to perform video object segmentation (top) by assigning each pixel to the most relevant visual word. Note that no ground-truth for the visual words are provided at training or test time. The algorithm yields a meaningful set of visual words, which implicitly represent object parts, and are robust over time.

Abstract

Meta learning has attracted a lot of attention recently. In this paper, we propose a fast and novel meta learning based method for video object segmentation that quickly adapts to new domains without any fine-tuning. The proposed model performs segmentation by matching pixels to object parts. The model represents object parts using deep visual words, and meta learns them with the objective of minimizing the object segmentation loss. This is however not straightforward as no ground-truth information is available for the object parts. We tackle this problem by iteratively performing unsupervised learning of the deep visual words, followed by supervised learning of the segmentation problem, given the visual words. Our experiments show that the proposed method performs on-par with state-of-the-art methods, while being computationally much more efficient.

1. Introduction

Meta learning is a method for learning to learn [40, 3], which can be used in few-shot learning problems [39, 45, 42]. By analyzing the learning process over multiple tasks, a meta learning algorithm learns how to learn a new but similar task quickly and more efficiently. Within this paper, we aim to explore meta learning in the context of video object segmentation.

Video object segmentation is defined as the task of segmenting one or multiple objects in a video. The most recent setup for this task, as defined by the DAVIS challenge [5, 38], is when the ground-truth masks of objects in the first frame are provided, and the goal is to segment them in rest of the video. It is a very challenging task, because at different time points, the objects of interest may be observed in different configurations due to e.g. partial occlusion, disappearing and re-appearing, shape deformation, and pose and scaling variation. Thus, a good object representation should be robust to these intra-object variances, and be able to quickly adapt to new object configurations in time.

The key idea is that these intra-object variations can be accounted for by assigning a pixel to only one relevant part of the object, rather than forcing it to resemble the entire object (e.g. matching a pixel from leg to the leg only). In other words, segmentation will be better if the matching is done to object parts. We introduce a set of deep visual words into our model to represent the object parts, and let the model learn these visual words (in an unsupervised manner), with the objective of minimizing the object segmentation loss. It

*Joint first authorship.

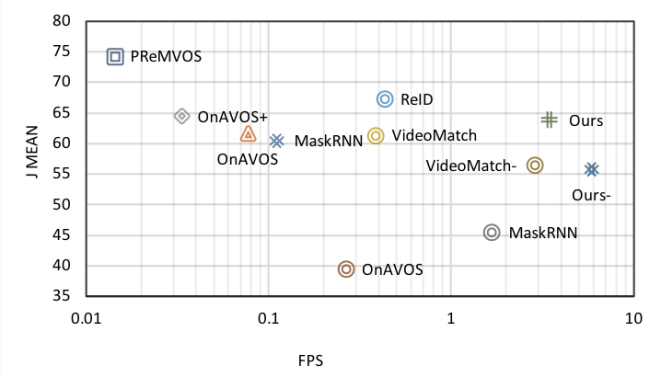

Figure 2: **Accuracy vs. Speed on DAVIS-2017 validation Dataset.** This figure illustrates the accuracy (mean $\mathcal{J}$) and speed (FPS) of various methods. The FPS axis is in log scale.

can be seen in Fig. 1 and Fig. 6 that the model can learn meaningful visual words which correspond to different object parts, which might be quite dissimilar, even though all are parts of the same object.

In this paper we present an iterative meta learning based algorithm, which learns the visual words from the first frame of the video using unsupervised learning. The algorithm then minimizes an object segmentation loss over the rest of the video, given the learned visual words. This procedure can be seen in Fig. 4. Finally, the model is able to do quick adaptation in video object segmentation, by matching the pixels to these learned meaningful visual words.

State-of-the-art methods for video object segmentation often fine-tune a pre-trained segmentation network on the first frame of the video [4, 46, 36, 2, 30], and some perform further online fine-tuning [46], to adapt better to the objects of interest . However, the fine-tuning process is very time consuming ($\sim$ 700s to 3h per video) [4, 27], because of which the best performing methods on DAVIS video-object segmentation challenge [38, 5] work at $\sim$ 15 seconds/frame. Thus, these methods are not very practical for online applications including autonomous cars and robots, where frame processing needs to be fast.

In contrast, our model is very simple and intuitive and does not include sophisticated modules that are often incorporated in more complex methods. Nevertheless, it performs on par with them by achieving a promising score of $\mathcal{J}\&\mathcal{F} = 67.3\%$ on DAVIS-2017, without any fine-tuning on the first frame of the video, while running at $\sim$ 3.5 fps, which is 1-2 orders of magnitude faster than them as shown in Fig. 2. Furthermore, our model enables objects to be represented by newer visual words on demand, as they change in shape or pose over time. This leads to a much more efficient domain adaptation, compared to the previous methods [46], where adaptation was addressed by fine tuning network parameters over time.

Contributions:
I) We show that efficient adaptation for video object segmentation can be achieved by learning to learn; We further present state-of-the-art results on two challenging video object segmentation benchmarks, among the methods without fine-tuning on the first frame of the test video; II) Our approach is very simple and intuitive, making it an appealing approach to video object segmentation. III) Our model constructs robust and meaningful visual words without ever seeing any visual words during training.

2. Related Work

Video object segmentation papers One of the widely used techniques for video object segmentation is to train a deep fully convolutional network (FCN) [32] for foreground/background object segmentation on a training dataset, and then further adapt it to the test video through a fine-tuning process, using dense ground-truth object masks that are available in the first frame of the video [4, 46, 19, 9, 36, 27, 41, 10]. However, fine-tuning process dramatically slows down these techniques, making them unsuitable for online applications. Our algorithm performs on-par with these methods, and is much faster as it does not need any fine-tuning.

A group of methods are proposed based on mask propagation [36, 47], where the segmentation network is guided by the predicted objects masks over time. These methods are vulnerable to object occlusion and may lose the track of objects at some points. Li et al. [30] addressed this problem by using re-id modules and traversing the video back-and-forth to recover any potential missed object prediction, however their method is not suitable for online and streaming applications.

There have been successful attempts to incorporate motion cues into video object segmentation systems, using *Optical Flow* [19, 18, 48, 2, 9, 13, 44, 50, 25]. However, Optical Flow maps are expensive to compute ($\sim$ 0.5s per frame using *FlowNet2* [22]) and slow down the algorithm. Unsupervised methods have also tackled the video object segmentation problem [29, 20, 28, 43, 26, 13], but they are not very well suited for multiple object segmentation and tracking, which is the aim of this paper.

Pixel-to-pixel matching has been used in [7, 21, 35, 41] to transfer the ground-truth information from the first frame to all subsequent frames of the video. Chen et al. [7] formulated the segmentation task as a pixel retrieval problem. In particular, they retrieved a set of similar pixels from the training pool and then applied a k-nearest-neighbour classifier, to predict the class label of a test pixel. Siamese networks have been used for pixel label transfer, using *soft matching layer* [21] and *stochastic pooling layer* [35]. Yoon et al. [41] proposed a pixel level matching module to match the image with the reference frame. The problem with

methods based on pixel-to-pixel matching is that they either have to explore all the pixels space, which is computationally expensive, or they might alleviate this problem by discarding many training pixels, which may lead to information loss.

Similar to our work, is [8], in which the authors performed video object segmentation via tracking object parts. More specifically, they first extract a fixed set of object parts using region proposals, and track their corresponding bounding boxes independently for the entire video. Next, they perform Region of Interest (ROI) segmentation within each bounding box to extract the object part from the background. Finally, they apply similarity based part aggregation to discard false positives. In contrast to [8], we do not put any constraint on the object parts, and all the visual words in our model are learned automatically through the meta-learning algorithm. Note that no ground-truth information is available for learning the visual words, and they are computed using an unsupervised approach, as described in Sec. 3.2. As shown in Fig. 1, the computed visual words are quite meaningful, even though we have not explicitly used any location information to learn them. Moreover, our model facilitates online adaptation, simply by updating the pool of visual words that are computed by the model itself, through time. In addition, unlike [8], our model achieves promising results, without any post-processing module such as Conditional Random Field (CRF), thanks to our end-to-end meta-training process.

Meta learning methods Meta learning has not been explored much for video object segmentation. Yang et al. [49] proposed a meta learning based method for fast adaptation of a deep segmentation network, where, unlike our approach, the parameters of the segmentation network are updated at the test time using different network modulators.

Our meta learning algorithm could be viewed as a generalization of Prototypical networks [42] and Matching networks [45]. In the Prototypical networks, the training data from each class is represented using only one single prototype, which as shown in our ablation study Table 4, is not sufficient for visual data representation in the complex tasks of object segmentation. On the other end of spectrum is Matching networks [45], where all training data samples play a role in the classification/matching task, regardless of how redundant or noisy they are. This deteriorates the performance of the system both in terms of accuracy and runtime speed. The proposed method in this paper combats these problems by introducing visual words into the meta learning process, which inherently represent object parts. It is a challenging task as there is no supervision for determining the visual words in the objects, and we propose to learn them in an unsupervised fashion.

Figure 3: **Formulation of video object segmentation as a meta learning problem.** Each video presents a new task; to learn from the ground truth object masks on the first frame (support set), to segment them on the rest of the frames in the video (query set).

3. Method

In this section, we first describe the formulation of video object segmentation problem as a meta-learning problem. Next, we explain our model and meta training strategy. Then we discuss our online adaptation approach.

3.1. Video Object Segmentation as Meta-Learning

Meta-learning, or learning to learn, is often defined as learning from a number of tasks in the training set, to become better at learning a new task in the test set [15, 42]. In the context of meta-learning for video object segmentation, the task is to learn from the ground-truth masks of the objects in the first frame of the video (support set) to segment and track them in rest of the video (query set). Our meta learning objective is to learn model parameters θ on a variety of tasks (videos), which are sampled from the distribution $p(\mathcal{T})$ of training tasks (i.e. meta-training set), such that the learned model performs well on a new unseen task (test video). Let the performance of the model on the n^{th} task, $\mathcal{T}_n$, be rendered by the loss $\mathcal{L}_{\mathcal{T}_n}(\theta)$. Then, the meta-training objective becomes:

$$\theta^* = \arg\min_{\theta} \sum_{\mathcal{T}_n \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_n}(\theta). \quad (1)$$

Fig. 3 illustrates our meta learning setup for video object segmentation based on this definition. In this setup, the support set $\mathcal{S}$ is the set of all labeled pixels in the first frame, $\mathcal{S} = \{x_i, y_i\}_{i=1}^{N}$. Here x_i represents the pixel i in the first frame, $y_i \in \mathcal{C} = \{1, ..., C\}$ is the ground truth class label of pixel x_i, N is the number of labeled pixels in the frame, and C is the number of object classes that need to be tracked and segmented in the video. Similarly the query set is defined by $\mathcal{Q} = \{x_j, y_j\}_{j=1}^{N \times F}$, where x_j denotes pixel j from the video, y_j represents the ground-truth class label for pixel j, and F is the number of frames in the video (excluding the

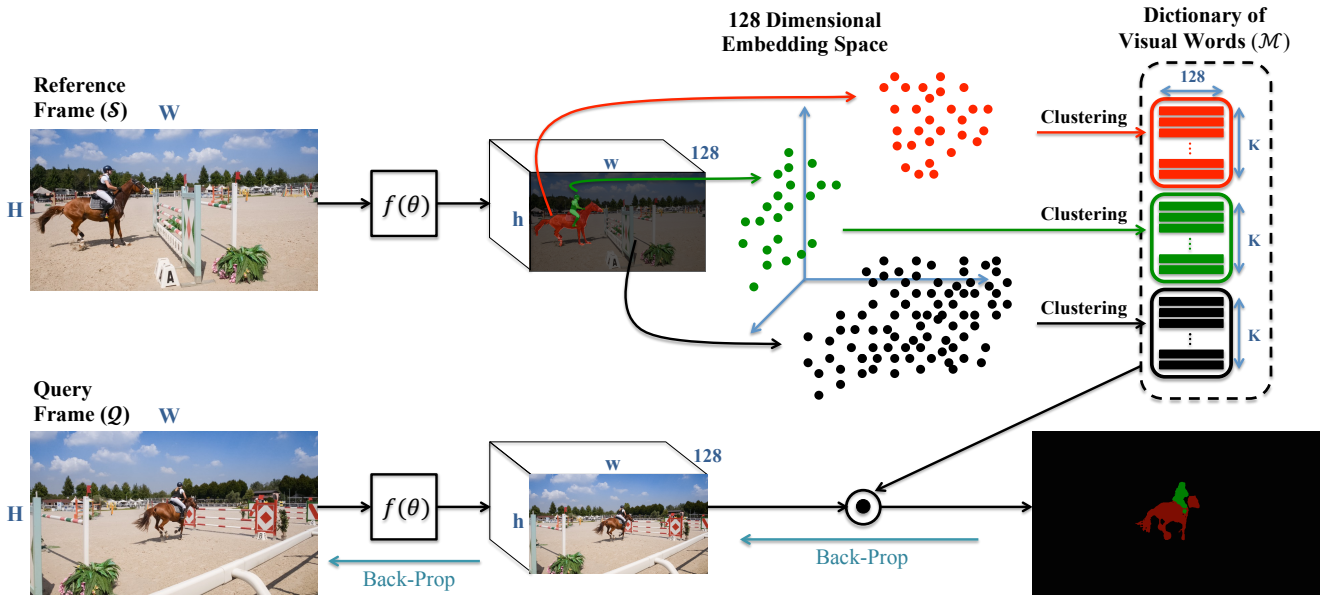

Figure 4: **Overview of the proposed method.** The first frame of the video (reference frame), which forms the support set $\mathcal{S}$ in our meta learning setup, passes through a deep segmentation network $f(\theta)$ to compute a 128D embedding vector for each pixel. Then a dictionary of deep visual words are learned by clustering these embeddings for each objects in the reference frame (Eq. 2). Then, pixels of the query frame are classified as one of the objects based to their similarities to the visual words (Eq. 3 and Eq. 4). The model is meta-trained by alternately learning the visual words given model parameters θ, and learning model parameters given the visual words.

first frame). The output of each task $\mathcal{T}$ is the set of predicted class labels for the pixels in $\mathcal{Q}$, i.e. $\hat{\mathcal{Y}} = \{\hat{y}_j\}_{j=1}^{N \times F}$.

Next, we describe our model for estimating the outputs of each task, i.e. the object label for every pixel in the query frames of the video.

3.2. Model

In order to predict the object class for each pixel in the query set $\mathcal{Q}$, we need to learn a representation for each object, using the information provided in the support set $\mathcal{S}$. In this paper, we propose to represent the objects in each video using a dictionary of deep visual words (Fig. 4). Each pixel in the query set is then classified into one of the object classes, based on the deep visual word it is assigned to. This process is described in more detail in Sec. 3.2.1 and Sec. 3.2.2.

Learning visual words is however a challenging task, as they do not come with any ground-truth information of the object parts. Therefore, the assignment of pixels to the visual words and consequently, the pixel-to-object assignment becomes an ill-posed problem. To address this, we propose a meta training algorithm, where we alternate between the unsupervised learning of deep visual words and supervised learning problem of pixel classification. More specifically, our model learns to learn a better classifier by optimizing these visual words.

3.2.1 Unsupervised Learning of Deep Visual Words

We initially pass the first frame of the video, which is the support set $\mathcal{S}$, through a deep segmentation network $f(\theta)$ to compute the encoding for each pixel x_i in $\mathcal{S}$, i.e., $f_\theta(x_i)$.

Next, we compute a set of deep visual words for all the pixels in each object class. In particular, let $\mathcal{S}_c$ be the set of pixels in $\mathcal{S}$ with class c. Each set $\mathcal{S}_c$ is partitioned into K clusters $\mathcal{S}_{c1}, ..., \mathcal{S}_{cK}$ using the k-means algorithm [1], with μ_{ck} being the respective centroids of the clusters, using the following objective:

$$\mathcal{S}_{c1}, ..., \mathcal{S}_{cK} = \underset{\mathcal{S}_{c1},....,\mathcal{S}_{cK}}{\arg\min} \sum_{k=1}^{K} \sum_{x_i \in \mathcal{S}_{ck}} \|f_\theta(x_i) - \mu_{ck}\|_2^2 , \tag{2a}$$

where

$$\mu_{ck} = \frac{1}{|\mathcal{S}_{ck}|} \sum_{x_i \in \mathcal{S}_{ck}} f_\theta(x_i). \tag{2b}$$

In other words, we represent the distribution of the pixels within each set $\mathcal{S}_c$ in the deep embedding space with a group of deep visual words $\mathcal{M}_c = \{\mu_{c1}, ..., \mu_{cK}\}$.

3.2.2 Supervised Learning for Pixel Classification

Once the deep visual words for each object are constructed, the probability of assigning a pixel $x_j \in \mathcal{Q}$ to the k^{th} visual word from object class c is computed using a non-parametric softmax classifier as follows:

$$p(c_k|x_j) = \frac{\exp\left(d(\mu_{ck}, f_\theta(x_j))\right)}{\sum_{\mu_i \in \mathcal{M}} \exp\left(d(\mu_i, f_\theta(x_j))\right)}, \quad (3)$$

where $\mathcal{M} = \bigcup_{c=1}^{C} \mathcal{M}_c$ is the dictionary of deep visual words for all objects present in the video, and d is the cosine similarity function.

We argue that, it is sufficient for a pixel to be very similar to at least one of the visual words within the object, in order for that pixel to be labelled as that object class. Hence, the probability of pixel x_j being a part of object class c is defined as:

$$p(y_j = c|x_j) = \frac{\max_{k \in \{1,..,K\}} p(c_k|x_j)}{\sum_{c'=1}^{C} \max_{k \in \{1,..,K\}} p(c'_k|x_j)}, \quad (4)$$

where the maximum operation selects the most similar visual word from each class c to pixel x_j. The key point is that we allow the model to account for intra-class variations by assigning a pixel to only one relevant visual word in the class, rather than forcing it to resemble all visual words. This is important because, as we see in Fig. 1 and Fig. 6, the model learns meaningful visual words that correspond to different object parts, which might be quite dissimilar, even though all are parts of the same object.

Next, we define a loss function for this pixel-wise classification problem, i.e., video object segmentation. Let $\mathcal{T}_n$ be the task in our meta training process. The loss for predicting class label $\hat{y}_j$ for pixels x_j in the query set is defined as:

$$\mathcal{L}_{\mathcal{T}_n} = -\frac{1}{|\mathcal{Q}|} \sum_{j=1}^{|\mathcal{Q}|} \log\left[p(\hat{y}_j = y_j|x_j)\right] - \frac{1}{|\mathcal{Q}|(C-1)} \sum_{j=1}^{|\mathcal{Q}|} \sum_{c=1, c\neq y_j}^{C} \log\left[1 - p(\hat{y}_j = c|x_j)\right], \quad (5)$$

where $|\mathcal{Q}|$ denotes the size of the query set (i.e. total number of pixels in the video $N \times F$) and y_j is the ground-truth class label for pixel x_j, as defined in Sec. 3.1.

The first term in Eq. 5 encourages the probability of the correct class to be high, whereas the second term tries to reduce the probability for all other classes. In other words, the proposed loss function attempts to pull each pixel closer towards the most similar visual word from the correct object class by maximizing the probability in Eq. 4 for the ground-truth class. At the same time, it aims to push the pixel away from the visual words of other classes.

Each iteration of our meta training algorithm is composed of the unsupervised learning process, where deep visual words are learned over the support set $\mathcal{S}$, followed by the supervised learning step, in which the model parameters θ are updated by minimizing the loss function in Eq. 5, according to Eq. 1. In other words, the model learns to learn deep visual words from the first frame of the video, to minimize an object segmentation loss over the rest of the video.

The proposed approach to meta learning is very flexible, in contrast to previous work, where each class is represented either by all data points from that class in the support set (as in Matching networks [45]), or by only one single prototype for that class (as in Prototypical networks [42]). Our model leverages the whole set of $\mathcal{S}$ to build a more robust and less noisy representation for all pixels in $\mathcal{S}$, using deep visual words.

Fig. 4 depicts the structure of the proposed model. We have used a ResNet-101 [17] architecture with dilated convolutions [6] as our segmentation network $f(\theta)$ to compute $f_\theta(x_i)$, though any state-of-the-art segmentation network can be used here as well. Deep segmentation networks ensure that the structural dependencies are leveraged when computing the encoding, which is crucial because we are dealing with a structured prediction problem. Thus, $f_\theta(x_i) \equiv f_\theta(x_i, \mathcal{S})$. The model architecture is described in more detail in Sec. 4.

3.3. Online Adaptation

An important attribute for video object segmentation algorithms is the ability to update online as the video progresses. This is vital because the objects of interest, as well as the background scene, might undergo significant deformation and changes in shape and appearance.

In this work we perform online adaptation by updating the set of visual words that represent objects. In particular, given a dictionary of deep visual words $\mathcal{M}$, captured up to the frame t_j, we predict the segmentation map in frame $t_{j+\delta}$, and treat it as a new support set $\mathcal{S}^\delta = \{x_i^\delta, y_i^\delta\}_{i=1}^N$, where y_i^δ is the predicted object class for pixel x_i^δ. Next, we compute an updated set of deep visual words $\mathcal{M}^\delta$ from the new support set using k-means algorithm as described in Sec. 3.2.1, and compute their corresponding cluster centroid representations by

$$\mu_{ck}^\delta = \frac{1}{|\mathcal{S}_{ck}^\delta|} \sum_{x_i \in \mathcal{S}_{ck}^\delta} f_\theta(x_i). \quad (6)$$

At this point, depending on how much the scene and objects have distorted in shape and appearance between frames t_j and $t_{j+\delta}$, these updated visual word representations could be quite similar or different to the previous ones. We update the main visual word set $\mathcal{M}$ with the new set $\mathcal{M}^\delta$, if there are $m^\delta \in \mathcal{M}_c^\delta$ and $m \in \mathcal{M}_c$, for which, $d(\mu_m^\delta, \mu_m) < 0.5$. In other words, if the objects undergo some deformation within the time interval δ, assuming δ is chosen moderately, we still expect the new visual words to resemble the main group of visual words $\mathcal{M}$ to some degree. However, if $d(\mu_m^\delta, \mu_m) > 0.5$, i.e. if there is a significant difference between the representation of the new visual

words and the ones in $\mathcal{M}$, it could be an indication of potential incorrect segmentation, which has led to assignment of irrelevant visual words $\mathcal{M}_c^\delta$ in the class c. As a result, they are discarded and will not be added to $\mathcal{M}$.

Note that during online adaptation, none of the existing visual words within $\mathcal{M}$ are discarded, because each object may revert to its original shape and appearance during a video sequence. This is where the max in Eq. 4 shows its merit, since no matter how many new visual words are added to the dictionary of class conditional deep visual words, the algorithm still picks up the most relevant one.

It is critical for online adaptation to use reliable and confident predictions of objects masks, in order to learn new visual words from them. We address this problem by applying a simple outlier removal process to the prediction outputs. More specifically, we refine the predictions of each object at every frame by, first finding its isolated predicted regions, and discarding the ones that have no intersection with predicted mask of the object in the previous frame. This process simply encodes the spatio-temporal consistency of the object masks over time. The effect of outlier removal on the performance of the system is investigated in the ablation study, Sec. 4.5.

4. Experiments

We evaluate our method on DAVIS-2017 [38], which is one of the primary benchmarks for assessing video object segmentation techniques. We perform a thorough analysis and ablation study on this dataset, and compare our method with state-of-the-arts, in terms of accuracy, as well as speed, which is a very important criterion for practical real-world applications. Furthermore, we report the results of evaluating our method on DAVIS-2016 [37] dataset, and compare it with state-of-the-art techniques.

4.1. Implementation Details

Our model architecture uses a Deeplab-ResNet-101 [6] segmentation network as the *encoder*. This encoder maps an input frame of size $[H, W]$ to an embedding of size $[H/8, W/8, 2048]$. This embedding is then upsampled to a $[H/2, W/2, 128]$ feature volume, where the number of channels in the feature space is dropped from 2048 down to 128 in order to keep the model computationally feasible. This is done using a *decoder* network, which is comprised of a bilinear upsampling layer in conjunction with a transposed convolution layer [12].

4.2. Model Pre-training

We first initialize our encoder network using the pre-trained Microsoft COCO weights [31], as done in other works [34, 46, 35]. Next, we pre-train our encoder-decoder network $f(\theta)$ using static images, following the training strategy in [35]. More specifically, we feed the network with images from Pascal instance segmentation dataset [14] as inputs, and use a binary cross-entropy loss function for training a Siamese network.

We then used the parameters of the pre-trained model as initialization for meta learning experiments.

4.3. Meta Learning

In order to meta train the model, we follow the *episodic* training procedure, which is the standard practice in meta learning-based approaches [45, 42, 15]. Each training episode is formed by sampling a support set $\mathcal{S}$ and a relevant query set $\mathcal{Q}$. The idea of episodic training is to, at each training iteration, mimic the inference procedure, where given the information provided by the support set, the query set should be classified. In this work, we build each episode by first randomly sampling a video from the training pool; treat the pixels of the first frame of the video as $\mathcal{S}$; and randomly selecting a set of query frames from the rest of the video and treat their pixels as $\mathcal{Q}$.

Following the proposed method in Sec. 3, we compute the loss according to Eq. 5 for the query set of each episode, and train the model.

4.4. Experimental Results

We report our results based on two standard metrics: *Jaccard* index ($\mathcal{J}$) and boundary F-score ($\mathcal{F}$).

DAVIS-2017

We meta trained the proposed model, which was already pre-trained as described in Sec. 4.2, on 60 video sequences taken from DAVIS-2017 training set.

During training, we learned 10 visual words from the support set of each episode, to represent each object (inc. background). However, during inference each object is represented by 50 visual words. Note that due to the complexity of the background scene compared to the foreground objects, four times more visual words are extracted from background.

We then evaluated the model on DAVIS-2017 validation set, which consists of 30 videos. This is a very challenging dataset, as some videos contain up to five dynamic objects to be tracked and segmented. Table 1 shows the performance of our method in comparison with the state-of-the-art methods. It is evident from this table that our method works on-par with the best performing methods, while performing at an encouraging speed of 0.29 second per frame which is 1-2 orders of magnitude faster. Also, our method gives the best accuracy ($\mathcal{J}\&\mathcal{F} = 0.673$) among the methods without fine-tuning on the first video frame. Note that our method does not utilize any fine-tuning over the first video frame, and can be applied to any test video straight-away without any over-head. In addition, we do not leverage Optical Flow

Figure 5: Qualitative segmentation outputs for some very challenging videos from DAVIS-2017 (each row), obtained using our model **without any fine-tuning**.

Method	FT	PP	OF	$\mathcal{J}(\%)$	$\mathcal{F}(\%)$	$\mathcal{J}\&\mathcal{F}(\%)$	Time(s)
MaskRNN [19]	✓		✓	60.5	–	–	9s
OSMN [49]	✓	✓		60.8	–	–	–
OnAVOS [46]	✓	✓		61.6	69.1	65.3	13s
OnAVOS† [46]	✓	✓		64.5	71.2	67.8	30s
VideoMatch [21]	✓			61.4	–	–	2.62s
ReID [30]	✓		✓	67.3	71.0	69.1	2.33s
OSVOS^{S} [4]	✓	✓		64.7	71.3	68.0	–
CINM† [2]	✓	✓	✓	67.2	74.4	70.7	$\sim$ 108s
PReMVOS† [23]	✓		✓	**74.3**	**82.2**	**78.2**	$\sim$ 70s
OnAVOS [46]				39.5	–	–	3.78s
MaskRNN [19]			✓	45.5	–	–	0.6s
VideoMatch [21]				56.5	–	–	0.35s
RGMP [47]				**64.8**	68.6	66.7	–
Ours^{-}				55.8	63.1	59.5	0.17s
Ours				63.9	**70.7**	**67.3**	0.29s

Table 1: **Results on DAVIS-2017 validation Dataset.** FT: Fine-Tuning on the first frame of the test video; PP: Post-Processing; OF: Optical Flow; $\mathcal{J}\&\mathcal{F}$: The mean of $\mathcal{J}$ and $\mathcal{F}$ metrics; Time(s): The time (in seconds) spent on each frame on average; Ours^{-}: Our model without online adaptation; †: Ensemble of models are used.

Method	FT	PP	OF	$\mathcal{J}(\%)$	$\mathcal{F}(\%)$	$\mathcal{J}\&\mathcal{F}(\%)$	Time(s)
MSK [36]	✓	✓	✓	79.7	75.4	77.5	12s
MaskRNN [19]	✓		✓	80.7	80.9	80.8	0.6s
OnAVOS [46]	✓	✓		86.1	84.9	85.5	13s
OSVOS [4]	✓	✓		79.8	80.6	80.2	9s
DRL [16]	✓			84.1	84.6	84.3	–
Lucid [27]	✓	✓	✓	84.8	–	–	$\sim$ 190s
OSVOS^{S} [4]	✓	✓		**85.6**	87.5	86.5	4.5s
PReMVOS† [23]	✓		✓	85.5	**88.6**	**87.0**	$\sim$ 70s
OTP [8]		✓		**82.4**	79.5	80.9	1.8s
CTN [25]			✓	73.5	69.3	71.4	1.33s
VPN [24]				70.2	65.5	67.8	0.63s
OTP [8]				77.9	76.0	76.9	0.60s
BVS [33]				60.0	58.8	59.4	0.37s
PML [7]				79.3	75.5	77.4	0.27s
OSMN [49]				74.0	–	–	0.14s
RGMP [47]				81.5	82.0	81.7	0.13s
Ours				81.4	**82.6**	**82.0**	0.25s

Table 2: **Results on DAVIS-2016 validation Dataset.** FT: Fine-Tuning on the first frame of the test video; PP: Post-Processing; OF: Optical Flow; $\mathcal{J}\&\mathcal{F}$: The mean of $\mathcal{J}$ and $\mathcal{F}$ metrics; Time(s): The time (in seconds) spent on each frame on average; †: Ensemble of models are used.

information [11] or any post-processing module, and this keeps the proposed method fast for practical applications.

Fig. 5 shows qualitative results computed using the proposed method for some videos from DAVIS-2017 validation set. Furthermore, Fig. 2 depicts an accuracy-vs-run-time diagram, which reveals that the proposed method offers a very good compensation between accuracy and speed, compared to other techniques.

DAVIS-2016

This dataset is a subset of DAVIS-2017, and includes 30 training videos and 20 validation videos. Moreover, the segmentation task is simplified to single-object tracking and segmentation. For this experiment, the pre-trained model was meta trained on DAVIS-2016 training set, where the episode configuration and optimization parameters were chosen similar to the DAVIS-2017 experiment. As shown in Table 2, our method achieves state-of-the-art performance among the fast approaches (i.e. approaches that do not fine-tune their model parameters on the first frame of the video), and is at par with the best-performing slow methods.

4.5. Ablation Study

In this section, we study the influence of different components and modules of our algorithm on the performance of the system.

Effect of Pre-Training and Meta Learning Table 3 reveals the contribution of each training stage in the final performance. Note that the initial model that only uses MS-COCO initializations for the encoder and random weights for the decoder, already performs on par with some recent techniques in Table 1 that are well trained for video object segmentation tasks. This indicates the potential of our algo-

Figure 6: **The effect of online adaptation on the representation of dynamic objects.** As the object pose changes over time, newer visual words (denoted by gray and light-blue colors) are learned and added to the dictionary of visual words.

Model Initialization	$\mathcal{J}(\%)$
MS-COCO	50.7
+ Pascal VOC	53.4
+ Meta Training on DAVIS 2017	**63.9**

Table 3: **The effect of different model initializations; DAVIS-2017 validation dataset.** Note that in the first row, only the encoder of our segmentation network is initialized with COCO weights and the decoder is randomly initialized. Despite that, the model still performs on par with some recent techniques, thanks to the representation power of visual words.

Dictionary Size (K)	1	5	10	20	50
$\mathcal{J}(\%)$	49.9	54.5	54.8	54.9	55.8

Table 4: **The effect of the size of visual word dictionary on model performance (without online adaptation); DAVIS-2017 validation dataset.** This table indicates the significance of visual words in the performance of the video object segmentation task. K denotes the number of visual words that are extracted from each object class. Note that except for the first case ($K = 1$), in all other cases $4 \times K$ visual words are extracted from the background class, due to its complexity.

rithm for visual word extraction and matching, and its ability to work well without much training effort. Nevertheless, it can still be further improved by training it on relevant data, as shown in Table 3.

Effect of the Size of Visual Word Dictionary Here we show how the performance of our method varies with the size of the visual word dictionary. The learned visual words for each object are in fact an approximate representation for the distribution of the pixels within that object. More complex objects with high intra-object variations require more visual words for a good representation, and one single prototype (as in [42]) may not be able to describe the entire pixel distribution of each object. Table 4 indicates the effect of number of visual words for each object on the performance of the object segmentation system. Evidently based on this table, our method outperforms prototypical networks [42] for video object segmentation task.

Interval (δ)	NA	30	20	10	5	2	1
$\mathcal{J}(\%)$	55.8	58.6	59.2	61.3	63.9	63.7	62.8

Table 5: **The effect of online adaptation on model performance; DAVIS-2017 validation dataset.** In this table, δ denotes the frame interval before every step of online adaptation. Choosing a moderate value for the online adaptation frequency is important. Very small values of δ might quickly enlarge the size of the visual word dictionary and potentially fill it with a lot of noisy visual words, obtained from the incorrect predicted masks. NA: No online adaptation.

Outlier Removal	Online Adaptation	$\mathcal{J}(\%)$
✗	✗	55.8
✓	✗	56.8
✗	✓	60.4
✓	✓	63.9

Table 6: **The effect of outlier removal on online adaptation; DAVIS-2017 validation dataset.** This table shows how refining the predicted object masks using the outlier removal process improves the online adaptation performance.

Effect of Online Adaptation Online adaptation, as described in Sec. 3.3, attempts to update the dictionary of visual words that are used for matching and label transfer. Table 5 illustrates how updating the dictionary $\mathcal{M}$ improves the performance of the system. By updating $\mathcal{M}$ more frequently (i.e. for smaller values of update interval δ), we let the system smoothly adapt to dynamic scenes and fast moving objects. However, very small values of δ might quickly enlarge the size of the visual word dictionary and potentially fill it with a lot of noisy visual words, obtained from the incorrect predicted masks.

Note that during online adaptation, model parameters are fixed and no forward/backward pass is performed in the network. Hence, our online adaptation scheme is still suitable for fast video object segmentation systems, unlike e.g. [46], where online adaptation incurs a computation time of 13s per frame. Fig. 6 illustrates how online adaptation updates the dictionary of visual words by incrementally learning new object parts automatically, and adding them to the set of visual words $\mathcal{M}$. Table 6 indicates the effect the outlier removal step in the performance of online adaptation algorithm.

5. Conclusion

In this paper, we proposed a novel method based on meta learning for quick adaptation in video object segmentation. The model inherently captures the intra-class variance, and also accommodates for object adaptation in time, without much computational effort. The proposed model represents objects using a dictionary of deep visual words, which are learned through an unsupervised learning process within the meta learning algorithm. We showed in the experimental results that our method performs on-par with state-of-the-art techniques, while being much faster as it does not need any fine-tuning.

References

[1] D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. In *ACM-SIAM Discrete Algorithms*, pages 1027–1035, 2007. 4

[2] L. Bao, B. Wu, and W. Liu. Cnn in mrf: Video object segmentation via inference in a cnn-based higher-order spatio-temporal mrf. In *CVPR*, June 2018. 2, 7

[3] S. Bengio, Y. Bengio, J. Cloutier, and J. Gecsei. On the optimization of a synaptic learning rule. In *Optimality in Biological and Artificial Networks*. 1995. 1

[4] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. One-shot video object segmentation. In *CVPR*, 2017. 2, 7

[5] S. Caelles, A. Montes, K.-K. Maninis, Y. Chen, L. Van Gool, F. Perazzi, and J. Pont-Tuset. The 2018 davis challenge on video object segmentation. *arXiv:1803.00557*, 2018. 1, 2

[6] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE PAMI*, 40(4):834–848, 2018. 5, 6

[7] Y. Chen, J. Pont-Tuset, A. Montes, and L. Van Gool. Blazingly fast video object segmentation with pixel-wise metric learning. In *CVPR*, June 2018. 2, 7

[8] J. Cheng, Y.-H. Tsai, W.-C. Hung, S. Wang, and M.-H. Yang. Fast and accurate online video object segmentation via tracking parts. In *CVPR*, June 2018. 3, 7

[9] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang. Segflow: Joint learning for video object segmentation and optical flow. In *ICCV*, Oct 2017. 2

[10] H. Ci, C. Wang, and Y. Wang. Video object segmentation by learning location-sensitive embeddings. In *ECCV*, September 2018. 2

[11] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, December 2015. 6

[12] V. Dumoulin and F. Visin. A guide to convolution arithmetic for deep learning. *ArXiv e-prints*, mar 2016. 6

[13] S. Dutt Jain, B. Xiong, and K. Grauman. Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos. In *CVPR*, July 2017. 2

[14] M. Everingham, S. M. Eslami, L. Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015. 6

[15] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*. 2017. 3, 6

[16] J. Han, L. Yang, D. Zhang, X. Chang, and X. Liang. Reinforcement cutting-agent learning for video object segmentation. In *CVPR*, June 2018. 7

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[18] P. Hu, G. Wang, X. Kong, J. Kuen, and Y.-P. Tan. Motion-guided cascaded refinement network for video object segmentation. In *CVPR*, June 2018. 2

[19] Y.-T. Hu, J.-B. Huang, and A. Schwing. Maskrnn: Instance level video object segmentation. In *NIPS*, 2017. 2, 7

[20] Y.-T. Hu, J.-B. Huang, and A. G. Schwing. Unsupervised video object segmentation using motion saliency-guided spatio-temporal propagation. In *ECCV*, September 2018. 2

[21] Y.-T. Hu, J.-B. Huang, and A. G. Schwing. Videomatch: Matching based video object segmentation. In *ECCV*, September 2018. 2, 7

[22] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017. 2

[23] B. L. J. Luiten, P. Voigtlaender. Premvos: Proposal-generation, refinement and merging for the davis challenge on video object segmentation 2018. *CVPR Workshops*, 2018. 7

[24] V. Jampani, R. Gadde, and P. V. Gehler. Video propagation networks. In *CVPR*, July 2017. 7

[25] W.-D. Jang and C.-S. Kim. Online video object segmentation via convolutional trident network. In *CVPR*, July 2017. 2, 7

[26] Y. Jun Koh and C.-S. Kim. Primary object segmentation in videos based on region augmentation and reduction. In *CVPR*, July 2017. 2

[27] A. Khoreva, R. Benenson, E. Ilg, T. Brox, and B. Schiele. Lucid data dreaming for object tracking. In *CVPR Workshops*, 2017. 2, 7

[28] S. Li, B. Seybold, A. Vorobyov, A. Fathi, Q. Huang, and C.-C. Jay Kuo. Instance embedding transfer to unsupervised video object segmentation. In *CVPR*, June 2018. 2

[29] S. Li, B. Seybold, A. Vorobyov, X. Lei, and C.-C. Jay Kuo. Unsupervised video object segmentation with motion-based bilateral networks. In *ECCV*, September 2018. 2

[30] X. Li and C. Change Loy. Video object segmentation with joint re-identification and attention-aware mask propagation. In *ECCV*, September 2018. 2, 7

[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *ECCV*, pages 740–755, 2014. 6

[32] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 2

[33] N. Maerki, F. Perazzi, O. Wang, and A. Sorkine-Hornung. Bilateral space video segmentation. In *CVPR*, 2016. 7

[34] K.-K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool. Video object segmentation without temporal information. *IEEE PAMI*, 2018. 6

[35] M. Najafi, V. Kulharia, T. Ajanthan, and P. H. S. Torr. Similarity learning for dense label transfer. *The 2018 DAVIS Challenge on Video Object Segmentation - CVPR Workshops*, 2018. 2, 6

[36] F. Perazzi, A. Khoreva, R. Benenson, B. Schiele, and A.Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017. 2, 7

[37] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *CVPR*, 2016. 6

[38] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 1, 2, 6

[39] S. Ravi and H. Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. 1

[40] J. Schmidhuber. Evolutionary principles in self-referential learning. (on learning how to learn: The meta-meta-... hook.). *Diploma thesis, Institut f. Informatik, Tech. Univ. Munich*, 1987. 1

[41] J. Shin Yoon, F. Rameau, J. Kim, S. Lee, S. Shin, and I. So Kweon. Pixel-level matching for video object segmentation using convolutional neural networks. In *ICCV*, Oct 2017. 2, 3

[42] J. Snell, K. Swersky, and R. Zemel. Prototypical networks for few-shot learning. In *NIPS*. 2017. 1, 3, 5, 6, 8

[43] P. Tokmakov, K. Alahari, and C. Schmid. Learning video object segmentation with visual memory. In *ICCV*, Oct 2017. 2

[44] Y.-H. Tsai, M.-H. Yang, and M. J. Black. Video segmentation via object flow. In *CVPR*, June 2016. 2

[45] O. Vinyals, C. Blundell, T. Lillicrap, k. kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In *NIPS*. 2016. 1, 3, 5, 6

[46] P. Voigtlaender and B. Leibe. Online adaptation of convolutional neural networks for video object segmentation. In *BMVC*, 2017. 2, 6, 7, 8

[47] S. Wug Oh, J.-Y. Lee, K. Sunkavalli, and S. Joo Kim. Fast video object segmentation by reference-guided mask propagation. In *CVPR*, June 2018. 2, 7

[48] H. Xiao, J. Feng, G. Lin, Y. Liu, and M. Zhang. Monet: Deep motion exploitation for video object segmentation. In *CVPR*, June 2018. 2

[49] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos. Efficient video object segmentation via network modulation. In *CVPR*, June 2018. 3, 7

[50] D. Yeo, J. Son, B. Han, and J. Hee Han. Superpixel-based tracking-by-segmentation using markov chains. In *CVPR*, July 2017. 2