Adaptive Masked Weight Imprinting for Few-Shot Segmentation

Mennatullah Siam 1 Boris Oreshkin 2

Abstract
Deep learning has mainly thrived by training on large-scale datasets. However, for a continual learning agent it is critical to incrementally update its model in a sample efficient manner. Learning semantic segmentation from few labelled samples can be a significant step toward such goal. We propose a novel method that constructs the new class weights from few labelled samples in the support set without back-propagation, while updating the previously learned classes. Inspiring from the work on adaptive correlation filters, an adaptive masked imprinted weights method is designed. It utilizes a masked average pooling layer on the output embeddings and acts as a positive proxy for that class. Our proposed method is evaluated on PASCAL-5 dataset and outperforms the state of the art in the 5-shot semantic segmentation. Unlike previous methods, our proposed approach does not require a second branch to estimate parameters or prototypes, and enables the adaptation of previously learned weights. Our adaptation scheme is evaluated on DAVIS video segmentation benchmark and our proposed incremental version of PASCAL and has shown to outperform the baseline model.

1. Introduction
Children are able to adapt their knowledge and learn about their surrounding environment with limited samples (Markman, 1989). One of the main bottlenecks in the current deep learning methods is their requirement to train on large-scale data. However, it is intractable to collect one large-scale dataset that contains all the required classes for different settings especially in robotics. It has even more impact when dealing with learning across unbalanced classes. That motivated the emergence of few-shot learning methods (Koch et al., 2015) (Vinyals et al., 2016) (Snell et al., 2017). Thus, the current challenge is to enable deep networks to mimic human behaviour, and update their model in a sample efficient and computationally efficient manner.

Most of the few-shot learning literature focused on image classification (Finn et al., 2017) (Koch et al., 2015) (Lin et al., 2017) (Vinyals et al., 2016) (Snell et al., 2017) (Qi et al., 2017) (Qiao et al., 2017). However, unlike image classification, semantic segmentation requires to learn a pixel-wise classification and can provide multiple classes in the support set. Thus, segmentation is more challenging in the low-shot regime than classification. One of the methods that are based on learning a parameter predictor was proposed by (Shaban et al., 2017). A conditional network method was proposed by (Rakelly et al., 2018) based on sparse or dense labels to guide the segmentation network. Another method that inspires from prototypical networks was proposed by (Dong & Xing, 2018). The previous methods require the training of an additional branch to act as a prototype learner or a parameter prediction branch. They can only operate in a static setting where one support set is provided to the model. If a new support set is provided to the model that has annotations for both novel classes and previously learned classes from few data, there is no direct extension to adapt their model.

In this paper we propose an adaptive masked weight imprinting scheme for few-shot semantic segmentation. Our main inspiration is from classical approaches in learning adaptive correlation filters (Bolme et al., 2010) (Henriques et al., 2015). Correlation filters date to 1980s by (Hester & Casasent, 1980) that proposed learning an averaged matched spatial filter constructed as a weighted linear combination of basis functions. (Bolme et al., 2010) proposed a fast object tracking method based on adaptive correlation filters, where the filters are updated using a running average. Our method proposes a novel scheme to compute convolutional filters to match the objects through masked weight imprinting, while adapting the learned ones. Weight imprinting (Qi et al., 2017) has been proposed for image classification and relates metric learning methods to softmax classification. It utilizes the normalized embeddings for the support set as proxies and concatenate it to the original weight matrix in the last classification layer. Since 1x1 convolution, that is typically used in segmentation networks, is equivalent to fully connected layers its filters can be imprinted as well. Nonetheless using the output embeddings directly can incorporate undesired features from other classes. Thus, a masked average pooling layer is utilized on the output embeddings with the segmentation label provided in the support set.
Unlike previous methods, our approach can easily operate with any pretrained network without the need to train a second branch. However, imprinting only the weights for the positive class, i.e. the newly added class, is insufficient as new samples will incorporate new information about other classes as well. For example, learning new class for boat will also entail learning new information about the background class which should include sea. Thus, a novel method for updating the weights of the old classes without back-propagation is proposed and termed as adaptive weight imprinting. This opens the door toward leveraging segmentation networks to continually learn objects in both sample efficient and computationally efficient manner. Our proposed method is evaluated on PASCAL-$5^i$ (Shaban et al., 2017) and DAVIS benchmark (Perazzi et al., 2016).

The contributions of this paper are: (1) we propose a masked weight imprinting scheme that is performed on multiple resolution levels. (2) we propose a novel adaptive weight imprinting scheme that inspires from adaptive correlation filters, in order to update the weights of previously learned classes. (3) Our method outperforms the state of the art on the 5-shot case on PASCAL-$5^i$, and the adaptation method outperforms the baseline method. (4) We propose iPASCAL which is the incremental version of PASCAL-VOC to evaluate the continual learning mode for segmentation.

2. Related Work

2.1. Few-shot Learning

Few-shot learning has mainly emerged in the classification domain. Datasets such as omniglot (Lake et al., 2011) and mini-ImageNet (Vinyals et al., 2016) are the most prominent datasets to evaluate few-shot learning methods for classification. During the evaluation for the few-shot learning methods, the model is provided with a support set and a query image. The support set contains the few labelled samples that can be used to train the model, while the query image is used to test the final model. The setup for few-shot learning is formulated as $n$ shot $m$ way classification, where $m$ denotes the number of classes provided in the support set, while $n$ denotes the number of samples that the model is required to classify among. In the few-shot classification problem the model is not required to localize and segment the different novel objects, which makes it easier than the few-shot segmentation problem.

One of the earliest attempts toward few-shot learning was using a Bayesian approach by (Fei-Fei et al., 2006). (Vinyals et al., 2016) proposed matching networks that learns an end-to-end differentiable nearest neighbour. (Finn et al., 2017) presented a model agnostic meta learning method that can adapt to learning new tasks. (Snell et al., 2017) proposed prototypical networks based on the assumption that, there exist an embedding space in which points belonging to one class, cluster around their corresponding prototype. (Qiao et al., 2017) proposed a parameter predictor method based on the activations learned during large-scale training. Finally, a method for computing imprinted weights was proposed by (Qi et al., 2017).

2.2. Few-shot Semantic Segmentation

Few-shot segmentation requires the model to provide pixel-wise classification for the query image. The current dataset used in the literature to evaluate the performance of few-shot segmentation is PASCAL-$5^i$ (Shaban et al., 2017), which is based on PASCAL VOC 2012 dataset (Everingham et al., 2015). The dataset is sub-divided into 4 folds each contain 5 classes as shown in Table 1. A fold contains labelled samples from 5 classes that are used for evaluating the few-shot learning method, while the rest 15 classes are used to train the model in the large-scale data regime. The baselines that were proposed by (Shaban et al., 2017) included the baseline classifiers with the nearest neighbour and logistic regression method trained on the extracted features from FCN-32s (Long et al., 2015) network that was trained on the 15 classes from PASCAL-$5^i$. Another baseline was proposed based on a siamese method that compares the support set and query image, and performs verification on the pixel-level. Finally, simple fine-tuning of the last layers of the network with the support set labels was proposed as another baseline to compare against. (Shaban et al., 2017) proposed a 2-branch method where the first branch is responsible for the segmentation. While the second branch is responsible for predicting the parameters for the final classification. A simulated few-shot setting for sampling a support set and its corresponding query image is used to train the model. (Rakelly et al., 2018) proposed another 2-branch method where the second branch acts as a conditioning branch instead. It has the ability to work with both dense and sparse annotations, and the second branch takes the image-label pairs as input. Finally, (Dong & Xing, 2018) inspired from prototypical networks, designed a method to learn prototypes for the few-shot segmentation problem. It proposed a 2-branch architecture as well, where the second branch is responsible for learning prototypes that can aid the final segmentation task. However, in all the previously proposed methods an extra branch is added to the initial segmentation architecture and trained in a simulated few-shot setting. There is also no direct extension on the above methods in order to adapt the previously learned weights when faced with a continuous stream of data.
3. Proposed Method

3.1. Few-shot Problem Setup

We formulate a problem similar to (Shaban et al., 2017), we define an initial phase of training with large scale dataset $D_{train}$. The training set $D_{train}$ includes semantic label maps for classes in $L_{train}$. Similar to the few shot learning literature a support set is sampled that is labelled with novel classes in $L_{test}$, where $L_{train} \cap L_{test} = \emptyset$. It is worth noting that during training if images contain labels from $L_{test}$ they are rather labelled as background or ignored in the back-propagation. Only images that include at least one pixel belonging to $L_{train}$ are included in $D_{train}$ for large-scale training.

During the test phase a support set is randomly sampled that contains pairs $S = \{(I_i, Y_i(l))\}_{i=1}^k$, where $I_i$ is the $i^{th}$ image in the set and $Y_i(l)$ is the corresponding binary mask. The binary mask $Y_i(l)$ is constructed with novel class $l$ labelled as foreground while the rest of the pixels are considered background. Similar to the k-shot setting in the few shot learning literature, $k$ indicates the number of images provided in the support set. A query image is randomly sampled from the test set in similar fashion to the support set.

3.2. Base Network

The backbone architecture used in our segmentation network is a VGG-16 (Simonyan & Zisserman, 2014) that is pre-trained on ImageNet (Deng et al., 2009). Similar to FCN8s architecture (Long et al., 2015) skip connections are used to benefit from higher resolution feature maps, and a 1x1 convolution layers are used to map from the feature space to the label space. However, unlike FCN8s we solely utilize bilinear interpolation layers with fixed weights for the upsampling. The main reason behind that choice, as it is hard to update the weights for the transposed convolution layers based on the support set.

On the other hand the 1x1 convolutional layers can benefit from masked weight imprinting to update its weights and accommodate the novel classes. During the test phase an extra normalization layer is utilized before all 1x1 convolutional layers that maps to the label space and normalizes the output tensors. An extension on the above base network uses dilated convolution (Yu & Koltun, 2015) and termed as Dilated-FCN8s. The last two pooling layers are replaced by dilated convolution with dilation factors 2 and 4 respectively. Thus, increasing the receptive field without affecting the resolution and improving the segmentation accuracy. Finally, a more compact version of the network with two final convolutional layers removed is denoted as Reduced-DFCN8s.

3.3. Masked Weight Imprinting

Inspiring from the work in few-shot image classification with imprinted weights (Qi et al., 2017) we propose to utilize a masked weight imprinting scheme. The weight imprinting method is based on the relation between metric learning methods and softmax classification. Metric learning methods such as neighbourhood component analysis (Goldberger et al., 2005) learn a distance metric using a softmax-like loss function. A modified version of neighbourhood component analysis learns it by using proxies as shown in equation (1).

$$L_{proxy}(x) = -\log \frac{\exp (-d(x, p(x)))}{\sum_{p(z) \in p(Z)} \exp (-d(x, p(z)))}$$

Where $p(Z)$ is the set of negative proxies, and the distance used is based on the L2 distance $d(x_1, x_2) = \|x_1 - x_2\|_2^2$. In the case of unit vectors minimizing the squared L2 distance becomes equivalent to maximizing the dot product.

$$\min\|x - p(x)\|_2^2 = \max x^T p(x)$$

When substituting in equation (1) we get a similar form to softmax classification as shown in equation (3).

$$L(x, p_k(x)) = -\log \frac{x^T p_k(x)}{\sum_{c \in C} x^T p_c(x)}$$

Where $p_k(x)$ is the proxy for class $k$, while $p_c(x)$ is the proxies representing the rest of classes in $C$. Similar to (Qi et al., 2017) this intuition can be further used to utilize the normalized embeddings for the few labelled samples from the novel class as proxies. These proxies can be fused directly as weights in the fully connected classification layer. There exist some major differences between the classification setting and the segmentation setting: (1) only convolutional layers are utilized in semantic segmentation. (2) the support set provides additional information not only to the novel class but it can include updated information about older classes as well. (3) The output embeddings are 3D tensors unlike in classification where the output embedding vector can be used directly. (4) Multi-resolution support is necessary to ensure the segmentation accuracy.
It is well known that 1x1 convolution are equivalent to fully connected layers so the same concept can be utilized in fully convolutional networks. In this case the weight filter is used as a proxy that convolves the output feature map and computes the extent to which the different parts in the feature map matches these proxies. In order to incorporate the pixels that belong mainly to the novel class, masked feature maps with the binary labels provided in the support set are used. This is followed by average pooling the masked feature maps per channel as in equation (4), we denote this layer as masked average pooling.

$$ p_l = \frac{1}{k} \sum_{i=1}^{k} \frac{1}{N} \sum_{x \in X} F^i(x) Y_l^i(x) $$ (4)

Where $Y_l^i$ is a binary mask for $i^{th}$ image with the novel class $l$, $F^i$ is the corresponding output feature maps for $i^{th}$ image. $X$ is the set of all possible spatial locations and $N$ is the number of pixels that are labelled as foreground for class $l$. The output from the masked average pooling layer $p_l$ can be further used as proxies representing class $l$. In the case of a novel class the imprinted weights can be utilized directly as the weight filter representing that new class. An average of all the masked pooling features for the k-shot samples provided in the support set is used.

3.4. Adaptive Weight Imprinting

In case of the older classes, it is not possible to use the weights directly as it overrides what the network learned from large-scale training although that would be valuable. It is not efficient either to ignore the newly added information for these negative proxies. An example that shows the reason to incorporate negative proxies is the addition of class boat, it is clear that the background class needs to be updated to include sea as well. Similar to adaptive correlation filters (Bolme et al., 2010) that was used in visual object tracking with handcrafted features, the convolutional layer weights in our model can be updated with the newly imprinted weights for that class in an adaptive scheme.

A running average is used to update the weights following equation (5) for older classes with the update rate $\alpha$. The update rate can either be treated as a hyper parameter or it can be learned separately according to the input embeddings. It can also be a learnable scalar value or it can vary according to which neuron is being updated. Figure 1 shows the adaptive masked weight imprinting scheme with each new support set for the positive proxy and the negative proxies used to update the weights of the previously known classes.

$$ W_{new} = (1 - \alpha) W_{old} + \alpha W_{imprinted} $$ (5)

Another major difference to the classification setting, is the use of the skip connections to improve the accuracy of the segmentation. Thus, masked weight imprinting is performed on the multiple resolution levels. The imprinted filters are computed on the two 1x1 convolutional layers following dilated convolutions in the case of Dilated-FCN8s. While imprinted filters are used in the 1x1 convolutional layers following the 3rd and 4th pooling layers in FCN8s. The output heat-maps from each resolution level are combined using...
summation, which results in the final probability maps.

3.5. Continuous Problem Setup

We formulate another setup for the continuous case in order to assess the effectiveness of providing an adaptive method. The PASCAL VOC dataset (Everingham et al., 2015) classes are split into $L_{\text{train}}$ and $L_{\text{incremental}}$ with 10 classes each, where $L_{\text{train}} \cap L_{\text{incremental}} = \emptyset$. The classes belonging to the $L_{\text{train}}$ are used to construct the training dataset $D_{\text{train}}$, and pre-train the segmentation network. Unlike the static setting in the few shot case, the continuous segmentation mode provides the image-label pairs incrementally with different encountered tasks. Each task introduces two novel classes to learn. The tasks are in the form of triplets $(t_i, (X_i, Y_i))$, where $(X_i, Y_i)$ represent the overall batch of images and labels from task $t_i$. The batch labels are for the two novel classes belonging to task $t_i$, and the previously learned classes in the encountered tasks $t_0, ..., t_{i-1}$.

The continuous setup can be a way to assess how adapting for novel classes in a task affects the previous tasks. In each task the model encounters each image-label pair from the current batch only once. This simulates the realistic setting where an agent is required to learn about novel objects incrementally. The proposed setup for the continuous mode on PASCAL VOC is termed as iPASCAL.

4. Experimental Results

Our proposed method is evaluated on pascal-5i (Shaban et al., 2017) in order to compare with the state of the art. In order to assess the effectiveness of proposing an adaptive method, we evaluate our method on DAVIS benchmark (Perazzi et al., 2016) and our proposed setup for iPASCAL. Evaluation is done using mean intersection over union (mIoU).

4.1. Few-Shot Segmentation Evaluation

The setup for pretraining the models to be tested on PASCAL-5i is detailed. The base network is trained using RMSProp (Hinton) with learning rate $10^{-6}$, and L2 regularization with a factor of $5 \times 10^{-4}$ on the 15 classes outside of the current fold. In the few-shot evaluation 1000 samples are used similar to OSLSM setup (Shaban et al., 2017). The alpha parameter used for adapting the previously learned weights is 0.5. Each sample contains a support set that has k labelled examples for the k-shot segmentation, and a query image. Both the support set and the query image are labelled with binary masks.

Table 3 and Table 4 show the results for the 1-shot and 5-shot segmentation respectively on PASCAL-5i using mIoU of the foreground class. Our method is compared to OSLSM (Shaban et al., 2017) and the baseline methods for few-shot segmentation. It shows that our method outperforms the baseline fine-tuning method by 7.6% in terms of mIoU, without the need for extra back-propagation iterations through directly using the imprinted weights. Our method performs on par with OSLSM (Shaban et al., 2017) method in the 1-shot in terms of mean across folds, while it outperforms OSLSM in the 5-shot case. However, unlike OSLSM our method does not need to train an extra branch for predicting the parameters. We conducted an experiment while setting $\alpha$ parameter to 0 to ensure that the adaptation of the background class weights improves the overall mIoU. The results on fold 0 was 13% instead of the 36% with $\alpha$ set to 0.5. This shows that the imprinting method standalone without the adaptation mechanism will degrade by 23% in mIoU.

Table 2 shows our method in comparison to the state of the art methods in terms of the mean on all folds for 1-shot and 5-shot segmentation with a different evaluation. The same evaluation utilized by (Rakelly et al., 2018) and (Dong & Xing, 2018) is rather used, which computes mIoU as the mean of the foreground and background IoU. Our proposed method outperforms the baseline FG-BG proposed by (Dong & Xing, 2018) with 2% and 4.4% mIoU in the 1-shot and 5-shot cases respectively. When our method is coupled with few iterations of back-propagation through the last layers it outperforms the the state of the art methods in the 5-shot case. In our experiments 10 iterations are used in the case of 1-shot evaluation, while 2 iterations per sample
Adaptive Masked Weight Imprinting for Few-Shot Segmentation

Table 3. Quantitative results for 1-way 1-shot segmentation on PASCAL-5i dataset. FT: denotes Fine-tuning. OSLSM method by (Shaban et al., 2017). Baseline methods evaluation are reported from (Shaban et al., 2017).

| Method  | 1-NN | Siamese | FT | OSLSM | ours (FCN8s) | ours (Dilated-FCN8s) | ours (Reduced-DFCN8s) |
|---------|------|---------|----|-------|--------------|----------------------|-----------------------|
| Fold 0  | 25.3 | 28.1    | 24.9 | 33.6  | 33.4         | 36.0                 | 39.2                  |
| Fold 1  | 44.9 | 39.9    | 38.8 | **55.3** | 46.8       | 48.0                 | 48.0                  |
| Fold 2  | **41.7** | 31.8 | 36.5 | 40.9  | 38.7         | 39.3                 | 39.3                  |
| Fold 3  | 18.4 | 25.8    | 30.1 | 33.5  | 33.2         | 33.8                 | **34.2**              |
| Mean    | 32.6 | 31.4    | 32.6 | **40.8** | 38.0       | 39.3                 | **40.2**              |

Table 4. Quantitative results for 1-way 5-shot segmentation on PASCAL-5i dataset. OSLSM method by (Shaban et al., 2017). LogReg baseline reported from (Shaban et al., 2017).

| Method  | LogReg | OSLSM | ours (FCN8s) | ours (Dilated-FCN8s) | ours (Reduced-DFCN8s) |
|---------|--------|-------|--------------|----------------------|-----------------------|
| Fold 0  | 35.9   | 35.9  | 37.4         | 40.5                 | **45.3**              |
| Fold 1  | 51.6   | **58.1** | 50.9       | 52.5                 | 51.4                  |
| Fold 2  | 44.5   | 42.7  | 44.0         | 44.8                 | **44.9**              |
| Fold 3  | 25.6   | 39.1  | 39.1         | **39.9**             | 39.5                  |
| Mean    | 39.3   | **43.9** | 42.9       | 44.4                 | **45.3**              |

Figure 2. Qualitative evaluation on PASCAL-5i. The support set and our proposed method prediction on the query image are shown in pairs for the 1-way 1-shot setting.

| Method              | mIoU | Blackswan | Bmx-Trees | Dance-Twirl | Motocross | Soapbox |
|---------------------|------|-----------|-----------|-------------|-----------|---------|
| Baseline            | 71.6 | 79.3      | 33.1      | 67          | 75.9      | 50.5    |
| Adaptive Imprinting | **72.2** | **81.3** | **35.3**  | **69.1**    | **77.4**  | **54.4** |

Table 5. Quantitative comparison between baseline and the adaptive masked imprinting scheme on DAVIS validation set.
4.2. Effectiveness of Adaptation

One of the major benefits of the inspiration from adaptive correlation filters is its ability to incrementally update the learned weights. In order to assess the adaptation capability of our model we compare it against the baseline method. Initially our method is evaluated on DAVIS-2016 benchmark (Perazzi et al., 2016), where our base network is a Wide ResNet model similar to (Voigtlaender & Leibe, 2017), in order to use a better baseline method on DA VIS dataset. The last 2 convolutional layers in our model has been changed to dilation rate of 1 instead of 12. High dilation rate corrupts the imprinting process, since the output spatial locations belonging to foreground class participate as well to the background class features. Figure 3 shows that effect and motivates the reason to use dilation rate of 1 at the layer where the imprinting process and masked average pooling will occur. The model is trained on PASCAL dataset first then finetuned on DAVIS training set. The model is then adapted with the first frame initialization in the validation set sequence during the weight imprinting process. We compare then both our adaptive masked imprinting scheme against the baseline model.

The alpha parameter during imprinting is set to 0.001 when operating on DAVIS since the model has learned background-foreground segmentation on the large-scale training set. During adaptation, the weights responsible for foreground and background classes are both updated using our proposed adaptive imprinting scheme. This is unlike PASCAL setup in which only the background class is updated, while the novel class is imprinted directly as no previously trained weights exist for it. Table 5 shows the mIoU over all the validation set and over some specific sequences. It shows generally that the imprinting scheme outperforms the baseline method. It is worth noting that the adaptation process does not incorporate fine-tuning and can be performed directly, but can also be paired with finetuning for a better improvement.

We conducted further experiments on iPASCAL, where triplets for the task, the corresponding images and semantic labels are provided. Semantic labels include the new classes in the current and previous encountered tasks. Figure 4 shows the comparison between naive fine-tuning from random weights against our proposed adaptive masked weight imprinting without any fine-tuning operations in terms of mIoU. It shows that masked imprinting provides better mIoU in comparison to fine-tuning that will lead to overfitting. Fine-tuning was conducted using RMSProp with learning rate $10^{-10}$. Fine-tuning is performed only to the last layers responsible for pixel-wise classification, while the feature extraction weights for VGG16 are fixed. It is worth noting that the current evaluation setting is a $n$-way 1-shot, where $n$ increases with 2 additional classes with each encountered task resulting in 10-way 1-shot evaluation in the last task. This explains the discrepancy between the mIoU in Table 3 and Figure 4, it demonstrates the fact that $n$-way classification is more challenging than 1-way.
Adaptive Masked Weight Imprinting for Few-Shot Segmentation

4.3. Learned Proxies Analysis

Since our method is highly dependant on the learned proxies, a further analysis on them is conducted. Figure 6 shows the T-SNE (Maaten & Hinton, 2008) embeddings for the proxies learned as output from our masked weight imprinting scheme. The plot shows the 5 classes belonging to fold 0 in PASCAL-5^{i}. Since our model performs imprinting on multiple resolution levels, the plot visualizes for the 3 different resolution levels. It motivates the reason for using the proxies as a way to imprint the weights for the final classification layer.

We compute the average Euclidean distance between the 1000 sampled support set and query image features. Figure 5 shows the box plots showing the average, minimum and maximum distances for the 3 different resolution levels. It shows how the average distance decreases as we increase the resolution level, which signifies the benefit form using higher resolution feature maps.

5. Conclusion

In this paper we proposed a novel approach for few-shot semantic segmentation using an adaptive masked imprinting scheme. Our proposed method outperforms the state of the art few-shot segmentation methods in the 5-shot setting, while it alleviates the need for training a second branch as the previous literature. Our method provides a scheme to adapt the weights of previously learned classes which can benefit continuous semantic segmentation. We proposed a novel setup iPASCAL to evaluate such an approach. The effectiveness of our proposed adaptation scheme is evaluated further on DAVIS video object segmentation dataset, where our adaptation outperforms the baseline model.

References

Bolme, D. S., Beveridge, J. R., Draper, B. A., and Lui, Y. M. Visual object tracking using adaptive correlation filters. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 2544–2550. IEEE, 2010.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009.
Adaptive Masked Weight Imprinting for Few-Shot Segmentation

IEEE Conference on, pp. 248–255. Ieee, 2009.

Dong, N. and Xing, E. P. Few-shot semantic segmentation with prototype learning. In BMVC, volume 3, pp. 4, 2018.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. The pascal visual object classes challenge: A retrospective. International journal of computer vision, 111(1):98–136, 2015.

Fei-Fei, L., Fergus, R., and Perona, P. One-shot learning of object categories. IEEE transactions on pattern analysis and machine intelligence, 28(4):594–611, 2006.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. arXiv preprint arXiv:1703.03400, 2017.

Goldberger, J., Hinton, G. E., Roweis, S. T., and Salakhutdinov, R. R. Neighbourhood components analysis. In Advances in neural information processing systems, pp. 513–520, 2005.

Henriques, J. F., Caseiro, R., Martins, P., and Batista, J. High-speed tracking with kernelized correlation filters. IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(3):583–596, 2015.

Hester, C. F. and Casasent, D. Multivariant technique for multiclass pattern recognition. Applied Optics, 19(11):1758–1761, 1980.

Hinton, G. Neural Networks for Machine Learning. Lecture Notes: overview of mini-batch gradient descent. URL: https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

Koch, G., Zemel, R., and Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In ICML Deep Learning Workshop, volume 2, 2015.

Lake, B., Salakhutdinov, R., Gross, J., and Tenenbaum, J. One shot learning of simple visual concepts. In Proceedings of the Annual Meeting of the Cognitive Science Society, volume 33, 2011.

Lin, X., Wang, H., Li, Z., Zhang, Y., Yuille, A., and Lee, T. S. Transfer of view-manifold learning to similarity perception of novel objects. arXiv preprint arXiv:1704.00033, 2017.

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440, 2015.

Maaten, L. v. d. and Hinton, G. Visualizing data using t-sne. Journal of machine learning research, 9(Nov):2579–2605, 2008.

Markman, E. M. Categorization and naming in children: Problems of induction. Mit Press, 1989.

Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., and Sorkine-Hornung, A. A benchmark dataset and evaluation methodology for video object segmentation. In Computer Vision and Pattern Recognition, 2016.

Qi, H., Brown, M., and Lowe, D. G. Learning with imprinted weights. arXiv preprint arXiv:1712.07136, 2017.