Hypercoringation Squeeze for Few-Shot Segmentation

Juhong Min  Dahyun Kang  Minsu Cho
Pohang University of Science and Technology (POSTECH), South Korea
http://cvlab.postech.ac.kr/research/HSNet/

Abstract

Few-shot semantic segmentation aims at learning to segment a target object from a query image using only a few annotated support images of the target class. This challenging task requires to understand diverse levels of visual cues and analyze fine-grained correspondence relations between the query and the support images. To address the problem, we propose Hypercorrelation Squeeze Networks (HSNet) that leverages multi-level feature correlation and efficient 4D convolutions. It extracts diverse features from different levels of intermediate convolutional layers and constructs a collection of 4D correlation tensors, i.e., hypercorrelations. Using efficient center-pivot 4D convolutions in a pyramidal architecture, the method gradually squeezes high-level semantic and low-level geometric cues of the hypercorrelation into precise segmentation masks in coarse-to-fine manner. The significant performance improvements on standard few-shot segmentation benchmarks of PASCAL-5i, COCO-20i, and FSS-1000 verify the efficacy of the proposed method.

1. Introduction

The advent of deep convolutional neural networks [17, 20, 64] has promoted dramatic advances in many computer vision tasks including object tracking [28, 29, 45], visual correspondence [22, 44, 48], and semantic segmentation. To cope with the challenge, there have been various attempts in semi- and weakly-supervised segmentation approaches [6, 26, 39, 66, 72, 77, 88] which in turn effectively alleviated the data-hunger issue. However, given only a few annotated training examples, the problem of poor generalization ability of the deep networks is yet the primary concern that many few-shot segmentation methods [10, 12, 13, 19, 33, 36, 37, 46, 54, 61, 63, 69, 70, 74, 75, 80, 83, 86, 87, 89] struggle to address.

Figure 1: Our model performs visual reasoning in coarse-to-fine manner by gradually squeezing high-dimensional hypercorrelation to the target segmentation mask with efficient 4D convolutions.

In contrast, human visual system easily achieves generalizing appearances of new objects given extremely limited supervision. The crux of such intelligence lies at the ability in finding reliable correspondences across different instances of the same class. Recent work on semantic correspondence shows that leveraging dense intermediate features [38, 42, 44] and processing correlation tensors with high-dimensional convolutions [30, 58, 71] are significantly effective in establishing accurate correspondences. However, while recent few-shot segmentation research began active exploration in the direction of correlation learning, most of them [36, 37, 46, 65, 73, 75, 80] neither exploit diverse levels of feature representations from early to late layers of a CNN nor construct pair-wise feature correlations to capture fine-grained correlation patterns. There have been some attempts [74, 86] in utilizing dense correlations with multi-level features, but they are yet limited in the sense that they simply employ the dense correlations for graph attention, using only a small fraction of intermediate conv layers.

In this work we combine the two of the most influential techniques in recent research of visual correspondence, multi-level features and 4D convolutions, and design a novel...
framework, dubbed Hypercorrelation Squeeze Networks (HSNet), for the task of few-shot semantic segmentation. As illustrated in Fig. 1, our network exploits diverse geometric/semantic feature representations from many different intermediate CNN layers to construct a collection of 4D correlation tensors, i.e., hypercorrelations, which represent a rich set of correspondences in multiple visual aspects. Following the work of FPN [34], we adapt pyramidal design to capture both high-level semantic and low-level geometric cues for precise mask prediction in coarse-to-fine manner using deeply stacked 4D conv layers. To reduce computational burden caused by such heavy use of high-dimensional convs, we devise an efficient 4D kernel via reasonable weight-sparsification which enables real-time inference while being more effective and light-weight than the existing ones. The improvements on standard few-shot segmentation benchmarks of PASCAL-5 [61], COCO-20 [35], and FSS-1000 [33] verify the efficacy of the proposed method.

2. Related Work

Semantic segmentation. The goal of semantic segmentation is to classify each pixel of an image into one of the predefined object categories. Prevalent segmentation approaches [5, 7, 47, 49, 52, 62, 76] typically employ encoder-decoder structure in their architecture; the encoder aggregates features along deep convolutional pathways and provides high-dimensional feature map in low-resolution and the corresponding decoder takes the output to predict segmentation mask by reversing this process [49]. Although the methods clearly show the effectiveness of the encoder-decoder architecture in the task of semantic segmentation, offering useful insights to our study, they still suffer apparent disadvantages of data-driven nature of neural networks: lack of generalizibility under insufficient training data.

Few-shot learning. To resolve the generalization problem, many recent approaches to image classification made various attempts in training deep networks with a few annotated examples [1, 18, 25, 31, 50, 53, 59, 65, 67, 73, 79, 84, 85]. Vinyals et al. [73] propose matching networks for one-shot learning; the method utilizes a special kind of mini-batches called episodes to match training and testing environments, facilitating better generalization on novel classes. Snell et al. [65] introduce prototypical networks which compute distances between representative embeddings, i.e., prototypes, for few-shot classification. With the growing interests in few-shot learning in classification domain, the problem of few-shot segmentation has attracted a great deal of attention as well. Shaban et al. [61] propose one-shot semantic segmentation networks which (meta-) learns to generate parameters of FCN [62]. Inspired by the prototypical networks [65], utilizing prototype representations to guide mask prediction in a query image became a popular paradigm in few-shot segmentation literature [10, 36, 37, 46, 63, 75, 80, 87, 89].

Witnessing the limitation of prototypical approaches, e.g., loss of spatial structure due to masked average pooling [89], work of [74, 86] build pair-wise feature correlations, e.g., graph attention, to retain the spatial structure of the images for fine-grained mask prediction. Note that both prototypical and graph-based methods fundamentally focus on learning to find reliable correspondences between support and query images for accurate mask prediction. In this work, we advance this idea and focus on learning to analyze correspondences using adequately designed learnable layers, e.g., 4D convolutions [58], for effective semantic segmentation.

Learning visual correspondences. The task of visual correspondence aims to find reliable correspondences under challenging degree of variations [3, 14, 15, 43, 60]. Many methods [21, 22, 30, 38, 42, 44, 56, 58, 81] typically built upon convolutional features pretrained on classification task [9], showing they serve as good transferable representations. Recent approaches to semantic correspondence [21, 38, 42, 44] show that efficiently exploiting different levels of convolutional features distributed over all intermediate layers clearly benefits matching accuracy. In wide-baseline matching literature, a trending choice is to employ 4D convolutions [30, 41, 57, 58, 71] on dense feature matches to identify spatially consistent matches by analyzing local patterns in 4D space. The use of multi-level features and relational pattern analysis using 4D convs are the two widely adopted techniques in the field of visual correspondence.

In this paper we adapt the two most influential methodologies in visual correspondence to tackle few-shot segmentation: multi-level features and 4D convolutions. Inspired by the previous matching methods [42, 44, 27], which use multi-level features to build effective “appearance features”, we construct high-dimensional “relational features” using intermediate CNN features and process them with a series of 4D convolutions. However, their quadratic complexity still remains a major bottleneck in designing cost-effective deep networks, constraining many previous matching methods [30, 57, 58, 71] to use only a few 4D conv layers. To resolve the issue, we develop a light-weight 4D convolutional kernel by collecting only a small subset of vital parameters for effective pattern recognition, which eventually leads to an efficient decomposition into a pair of 2D conv kernels with a linear complexity. Our contributions can be summarized as follows:

- We present the Hypercorrelation Squeeze Networks that analyze dense feature matches of diverse visual aspects using deeply stacked 4D conv layers.
- We propose center-pivot 4D conv kernel which is more effective than the existing one in terms both accuracy and speed, achieving real-time inference.
- The proposed method sets a new state of the art on three standard few-shot segmentation benchmarks: PASCAL-5 [61], COCO-20 [35], and FSS-1000 [33].
3. Problem Setup

The goal of few-shot semantic segmentation is to perform segmentation given only a few annotated examples. To avoid the risk of overfitting due to insufficient training data, we adopt widely used meta-learning approach called episodic training [73]. Let us denote the respective training and test sets as $D_{\text{train}}$ and $D_{\text{test}}$ which are disjoint w.r.t. to object classes. Both sets consist of multiple episodes each of which is composed of a support set $S = (I^s, M^s)$ and a query set $Q = (I^q, M^q)$ where $I^s$ and $M^s$ are an image and its corresponding mask label respectively. During training, our model iteratively samples an episode from $D_{\text{train}}$ to learn a mapping from $(I^s, M^s, I^q)$ to query mask $M^q$. Once the model is trained, it uses the learned mapping for evaluating without further optimization, i.e., the model takes randomly sampled $(I^s, M^s, I^q)$ from $D_{\text{test}}$ to predict query mask.

4. Proposed Approach

In this section, we present a novel few-shot segmentation architecture, Hypercorrelation Squeeze Networks (HSNet), which capture relevant patterns in multi-level feature correlations between a pair of input images to predict fine-grained segmentation mask in a query image. As illustrated in Fig. 2, we adopt an encoder-decoder structure in our architecture: the encoder gradually squeezes dimension of the input hypercorrelations by aggregating their local information to a global context, and the decoder produces the encoded context to predict a query mask. In Sec. 4.1-4.3, we demonstrate each pipeline in one-shot setting, i.e., the model predicts the query mask given $I^q$ and $S = (I^s, M^s)$. In Sec. 4.4, to mitigate large resource demands of 4D convs, we present a lightweight 4D kernel which greatly improves model efficiency in terms of both memory and time. In Sec. 4.5, we demonstrate how the model can be easily extended to $K$-shot setting, i.e., $S = \{(I_k, M_k)\}_{k=1}^K$, without loss of generality.

4.1. Hypercorrelation construction

Inspired by recent semantic matching approaches [38, 42, 44], our model exploits a rich set of features from the intermediate layers of a convolutional neural network to capture multi-level semantic and geometric patterns of similarities between the support and query images. Given a pair of query and support images, $I^q, I^s \in \mathbb{R}^{3 \times H \times W}$, the backbone network produces a sequence of $L$ pairs of intermediate feature maps $\{(F_l^q, F_l^s)\}_{l=1}^L$. We mask each support feature map $F_l^s \in \mathbb{R}^{H_l \times H_l \times W_l}$ using the support mask $M^s \in \{0, 1\}^{H \times W}$ to discard irrelevant activations for reliable mask prediction:

$$\hat{F}_l^s = F_l^s \odot \zeta(M^s),$$

where $\odot$ is Hadamard product and $\zeta(\cdot)$ is a function that bilinearly interpolates input tensor to the spatial size of the feature map $\hat{F}_l^s$ at layer $l$ followed by expansion along channel dimension such that $\zeta : \mathbb{R}^{H \times W} \rightarrow \mathbb{R}^{H_l \times H_l \times W_l}$. For the subsequent hypercorrelation construction, a pair of query and masked support features at each layer forms a 4D correlation tensor $C_l \in \mathbb{R}^{H_l \times W_l \times H_l \times W_l}$ using cosine similarity:

$$\hat{C}_l(x^q, x^s) = \text{ReLU} \left( \frac{F_l^q(x^q) \cdot \hat{F}_l^s(x^s)}{||F_l^q(x^q)|| \cdot ||\hat{F}_l^s(x^s)||} \right),$$

where $x^q$ and $x^s$ denote 2-dimensional spatial positions of feature maps $F_l^q$ and $\hat{F}_l^s$ respectively, and ReLU suppresses noisy correlation scores. From the resultant set of 4D correlations $\{\hat{C}_l\}_{l=1}^L$, we collect 4D tensors if they have the same spatial sizes and denote the subset as $\{\hat{C}_l\}_{l \in L_p}$ where $L_p$ is a subset of CNN layer indices $\{1, \ldots, L\}$ at some pyramidal layer $p$. Finally, all the 4D tensors in $\{\hat{C}_l\}_{l \in L_p}$ are concatenated along channel dimension to form a hypercorrelation $C_p \in \mathbb{R}^{L_p \times H_p \times W_p \times H_p \times W_p}$ where $(H_p, W_p, H_p, W_p)$, with abuse of notation, represents the
spatial resolution of the hypercorrelation at pyramidal layer \( p \). Given \( P \) pyramidal layers, we denote hypercorrelation pyramid as \( C = \{C_p\}_{p=1}^P \), representing a rich collection of feature correlations from multiple visual aspects.

### 4.2. 4D-convolutional pyramid encoder

Our encoder network takes the hypercorrelation pyramid \( C = \{C_p\}_{p=1}^P \) to effectively squeeze it into a condensed feature map \( Z \in \mathbb{R}^{128 \times H \times W} \). We achieve this correlation learning using two types of building blocks: a squeezing block \( f_p^{\text{sqz}} \) and a mixing block \( f_p^{\text{mix}} \). Each block consists of three sequences of multi-channel 4D convolution, group normalization [78], and ReLU activation as illustrated in Fig. 3. In the squeezing block \( f_p^{\text{sqz}} \), large strides periodically squeeze the last two (support) spatial dimensions of \( C_p \) down to \( (H_p, W_p) \) while the first two spatial (query) dimensions remain the same as \( (H_q, W_q) \), i.e., \( f_p^{\text{sqz}} : \mathbb{R}^{C_p \times H_q \times W_q \times H_q \times W_q} \rightarrow \mathbb{R}^{128 \times H_p \times W_p \times H_q \times W_q} \), where \( H_p > H_q \) and \( W_p > W_q \). Similar to FPN [34] structure, two outputs from adjacent pyramidal layers, \( p \) and \( p+1 \), are merged by element-wise addition after upsampling the (query) spatial dimensions of the upper layer output by a factor of 2. The mixing block \( f_p^{\text{mix}} : \mathbb{R}^{128 \times H_p \times W_p \times H_q \times W_q} \rightarrow \mathbb{R}^{128 \times H_p \times W_p \times H_q \times W_q} \) then processes this mixture with 4D convolutions to propagate relevant information to lower layers in a top-down fashion. After the iterative propagation, the output tensor of the lowest mixing block \( g_p^{\text{mix}} \) is further compressed by average-pooling its last two (support) spatial dimensions, which in turn provides a 2-dimensional feature map \( Z \in \mathbb{R}^{128 \times H \times W} \) that signifies a condensed representation of the hypercorrelation \( C \).

### 4.3. 2D-convolutional context decoder

The decoder network consists of a series of 2D convolutions, ReLU, and upsampling layers followed by softmax function as illustrated in Fig. 2. The network takes the context representation \( Z \) and predicts two-channel map \( \hat{M}^q \in \{0, 1\}^{2 \times H \times W} \) where two channel values indicate probabilities of foreground and background. During training, the network parameters are optimized using the mean of cross-entropy loss between the prediction \( \hat{M}^q \) and the ground-truth \( M^q \) over all pixel locations. During testing, we take the maximum channel value at each pixel to obtain final query mask prediction \( \hat{M}^q \in \{0, 1\}^{H \times W} \) for evaluation.

### 4.4. Center-pivot 4D convolution

Apparently, our network with such a large number of 4D convolutions demands a substantial amount of resources due to the curse of dimensionality, which constrained many visual correspondence methods [22, 30, 32, 58, 71] to use only a few 4D conv layers. To address the concern, we revisit the 4D convolution operation and delve into its limitations.

---

\*The correlation tensor \( c \) is the output of cosine similarity (Eqn. 2) between a pair of feature maps, \( F, F' \in \mathbb{R}^{H \times W} \), and \( x \) and \( x' \) denote 2-dimensional spatial positions of the respective feature maps.
that settle the latter problem (numerical instability). In this work we introduce a novel weight-sparsification scheme to address both issues at the same time.

**Center-pivot 4D convolution.** Our goal is to design a lightweight 4D kernel that is efficient in terms of both memory and time while effectively approximating the existing ones [58, 81]. We achieve this via a reasonable weight-sparsification; from a set of neighborhood positions within a local 4D window of interest, our kernel aims to disregard a large number of activations located at fairly insignificant positions in the 4D window, thereby focusing on a small subset of relevant activations only. Specifically, we consider the activations at positions that *pivots* either one of 2-dimensional *centers*, e.g., \( x \) or \( x' \), as the foremost influential ones as illustrated in Fig. 4. Given 4D position \((x, x')\), we collect its neighbors if and only if they are adjacent to either \( x \) or \( x' \) in its corresponding 2D subspace and define two respective sets as \( \mathcal{P}_c(x, x') = \{(p, p') \in \mathcal{P}(x, x') : p = x\} \) and \( \mathcal{P}_c(x, x') = \{(p, p') \in \mathcal{P}(x, x') : p' = x'\} \). The set of *center-pivot neighbors* is defined as \( \mathcal{P}_{CP}(x, x') = \mathcal{P}_c(x, x') \cup \mathcal{P}_c(x, x') \). Based on these two subsets of neighbors, center-pivot 4D convolution can be formulated as a union of two separate 4D convolutions:

\[
(c * k_{CP})(x, x') = (c * k_c)(x, x') + (c * k_{c'})(x, x')
\]

(4)

where \( k_c \) and \( k_{c'} \) are 4D kernels convolved on \( \mathcal{P}_c(x, x') \) and \( \mathcal{P}_{c'}(x, x') \) respectively. Note that \( (c * k_c)(x, x') \) is equivalent to convolutions with a 2D kernel \( k_{c}^{2D} = k(0, :) \in \mathbb{R}^{k \times k} \) performed on 2D slice of 4D tensor \( c(x, :) \). Similarly, with \( k_{c'}^{2D} = k(:, 0) \in \mathbb{R}^{k \times k} \), we reformulate Eqn. 4 as follows

\[
(c * k_{CP})(x, x') = \sum_{p' \in \mathcal{P}(x')} c(x, p') k_{c'}^{2D}(p' - x')
\]

(5)

\[
+ \sum_{p \in \mathcal{P}(x)} c(p, x') k_{c}^{2D}(p - x),
\]

which performs two different convolutions on separate 2D subspaces, having a linear complexity. In Sec. 5.2, we experimentally demonstrate the superiority of the center-pivot 4D kernels over the existing ones [58, 81] in terms of accuracy, memory, and time. We refer the readers to our supplementary materials for a complete derivation of Eqn. 5.

4.5. **Extension to \( K \)-shot setting**

Our network can be easily extended to \( K \)-shot setting: Given \( K \) support image-mask pairs \( S = \{(I_s^k, M_s^k)\}_{k=1}^K \) and a query image \( I^q \), model performs \( K \) forward passes to provide a set of \( K \) mask predictions \( \{\hat{M}_s^k\}_{k=1}^K \). We perform voting at every pixel location by summing all the \( K \) predictions and divide each output score by the maximum voting score. We assign foreground labels to pixels if their values are larger than some threshold \( \tau \) whereas the others are classified as background. We set \( \tau = 0.5 \) in our experiments.

5. **Experiment**

In this section we evaluate the proposed method, compare it with recent state of the arts, and provide in-depth analyses of the results with ablation study.

**Implementation details.** For the backbone network, we employ VGG [64] and ResNet [17] families pre-trained on ImageNet [9], e.g., VGG16, ResNet50, and ResNet101. For VGG16 backbone, we extract features after every conv layer in the last two building blocks: from \( \text{conv4}_{x} \) to \( \text{conv5}_{x} \), and after the last maxpooling layer. For ResNet backbones, we extract features at the end of each bottleneck before ReLU activation: from \( \text{conv3}_{x} \) to \( \text{conv5}_{x} \). This feature extracting scheme results in 3 pyramidal layers \((P = 3)\) for each backbone. We set spatial sizes of both support and query images to \( 400 \times 400 \), i.e., \( H, W = 400 \), thus having \( H_1, W_1 = 50, H_2, W_2 = 25 \), and \( H_3, W_3 = 13 \). The network is implemented in PyTorch [51] and optimized using Adam [24] with learning rate of 1e-3. We freeze the pre-trained backbone networks to prevent them from learning class-specific representations of the training data.

**Datasets.** We evaluate the proposed network on three standard few-shot segmentation datasets: PASCAL-5i [61], COCO-20i [35], and FSS-1000 [33]. PASCAL-5i is created from PASCAL VOC 2012 [11] with extra mask annotations [16], consisting of 20 object classes that are evenly divided into 4 folds: \( \{5^i : i \in \{0, 1, 2, 3\}\} \). COCO-20i consists of mask-annotated images from 80 object classes divided into 4 folds: \( \{20^i : i \in \{0, 1, 2, 3\}\} \). Following common training/evaluation scheme [37, 46, 70, 74, 80], we conduct cross-validation over all the folds; for each fold \( i \),
samples from the other remaining folds are used for training and 1,000 episodes from the target fold $i$ are randomly sampled for evaluation. For every fold, we use the same model with the same hyperparameter setup following the standard cross-validation protocol. FSS-1000 contains mask-annotated images from 1,000 classes divided into training, validation and test splits having 520, 240, and 240 classes respectively.

Evaluation metrics. We adopt mean intersection over union (mIoU) and foreground-background IoU (FB-IoU) as our evaluation metrics. The mIoU metric averages over IoU values of all classes in a fold: $\text{mIoU} = \frac{1}{|C|} \sum_{c=1}^{C} \text{IoU}_c$ where $C$ is the number of classes in the target fold and $\text{IoU}_c$ is the intersection over union of class $c$. FB-IoU ignores object classes and computes average of foreground and background IoUs: $\text{FB-IoU} = \frac{1}{2}(\text{IoU}_P + \text{IoU}_B)$ where $\text{IoU}_P$ and $\text{IoU}_B$ are respectively foreground and background IoU values in the target fold. As mIoU better reflects model generalization capability and prediction quality than FB-IoU does, we mainly focus on mIoU in our experiments.

5.1. Results and analysis

We evaluate the proposed model on PASCAL-5$^i$ [61] in mIoU and FB-IoU. Some results are from [4, 37, 70, 74, 80]. Superscript $^*$ denotes our model without support feature masking (Eqn. 1). Numbers in bold indicate the best performance and underlined ones are the second best.

Table 2: Performance on COCO-20$^i$ [46] in mIoU and FB-IoU. The results of other methods are from [4, 37, 70, 74, 80].

Table 3: Mean IoU comparison on FSS-1000 [33]. Some results are from [2, 74].

We conduct additional experiments without support feature masking (Eqn. 1). Note that this setup is similar to co-segmentation problem [8, 68, 82] with stronger demands for generalizability since the model is evaluated on novel
As seen in Tab. 4, our model, which is trained without support masking still performs remarkably well, achieving 1.4% mIoU improvement over the previous best method [70] in 1-shot setting whereas it rivals [4, 70] in 5-shot setting. This interesting result reveals that our model is also capable of identifying ‘common’ instances across different input images as well as predicting fine-grained segmentation masks.

**Robustness to domain shift.** To demonstrate the robustness of our method to domain shift, we evaluate COCO-trained HSNet on each fold of PASCAL-5′ following the recent work of [4]. We use the same training/test folds as in [4] where object classes in training and testing do not overlap. As seen in Tab. 4, our model, which is trained without any data augmentation methods with 18 times smaller number of trainable parameters compared to [4] (2.6M vs. 46.7M), performs robustly in presence of large domain gaps between COCO-20′ and PASCAL-5′, surpassing [4] by 1.0% in 5-shot setting, and further improves with a larger backbone, e.g., ResNet101. The results clearly show the robustness of our method to domain shift, and may further increase when trained with data augmentations used in [4, 70].

### 5.2. Ablation study

We conduct extensive ablation study to investigate the impacts of major components in our model: hypercorrelations, pyramidal architecture, and center-pivot 4D kernels. We also study how freezing backbone networks prevents overfitting and helps generalization on novel classes. All ablation study experiments are performed with ResNet101 backbone on PASCAL-5′ [61] dataset.

**Ablation study on hypercorrelations.** To study the effect of intermediate correlations \( \{ \mathcal{C}_l \}_{l \in \mathcal{L}_p} \) in hypercorrelation \( \mathcal{C}_p \in \mathbb{R}^{[\mathcal{L}_p \times H_p \times W_p]} \), we form single-channel hypercorrelations using only a **single intermediate correlation**. Specifically, we form two different single-channel hypercorrelations using the smallest (shallow) and largest (deep) layer indices in \( \mathcal{L}_p \) and denote the hypercorrelations as \( \mathcal{C}_p^{\text{shallow}} \) and \( \mathcal{C}_p^{\text{deep}} \) in \( \mathbb{R}^{[1 \times H_p \times W_p]} \), and compare the results with ours (\( \mathcal{C}_p \)) in Fig. 5. The large performance gaps between \( \mathcal{C}_p \) and the single-channel hypercorrelations confirm that capturing diverse correlation patterns from dense intermediate CNN layers is crucial in effective pattern analyses. Performance degradation from \( \mathcal{C}_p^{\text{deep}} \) to \( \mathcal{C}_p^{\text{shallow}} \) indicates that reliable feature representations typically appear at deeper layers of a CNN.

![Figure 5: Ablation study on hypercorrelations on PASCAL-5′ [61] dataset in 1-shot (left) and 5-shot (right) mIoU results.](image)

![Figure 6: Ablation study on pyramid layers on PASCAL-5′ [61] dataset in 1-shot (left) and 5-shot (right) mIoU results.](image)

![Figure 7: Ablation study on hypercorrelation pyramid layers.](image)
Table 5: Comparison between three different 4D conv kernels in model size, per-episode inference time, memory consumption and FLOPs.

For fair comparison, the inference times of all the models are measured on a machine with an Intel i7-7820X and an NVIDIA Titan-XP.

![Image](image.png)

Figure 8: The effect of depths in building blocks: $f_p^{sep}$ and $f_p^{mix}$.

per-episode inference time (1-shot), memory consumption, and floating point operations per second (FLOPs) with ours.

Table 5 summarizes the results. The proposed kernel records the fastest inference time with the smallest memory/FLOPs requirements while being comparably effective than the other two. The results clearly support our claim that a large part of parameters in a high-dimensional kernel can safely be discarded without harming the quality of predictions; only a few relevant parameters are sufficient and even better for the purpose. While both the separable [81] and our center-pivot 4D convolutions operate on two separate 2D convolutions, auxiliary transformation layers with multiple batch normalizations that make the separable 4D conv numerically stable in its sequential design result in twice larger number of parameters (4.4M vs. 2.6M) and slower inference time (28.48ms vs. 25.51ms) than ours.

The number of 4D layers in building blocks. We also perform experiments with varying number of 4D conv layers in the two building blocks: $f_p^{sep}$ and $f_p^{mix}$. Figure 8 plots 1-shot and 5-shot mIoU results on PASCAL-51. In the experiments, appending additional 4D layers (with a group norm and a ReLU activation) in the building blocks provides clear performance improvements up to three layers but the accuracy eventually saturates after all. Hence we use a stack of 3 4D layers for both.

Finetuning backbone networks. To investigate the significance of learning ‘feature correlations’ over learning ‘feature representation’ in few-shot regime, we finetune our backbone network and compare learning processes of the finetuned model and ours (frozen backbone). Figure 9 plots the training/validation curves of the finetuned model and ours on every fold of PASCAL-51. The finetuned model rapidly overfits to the training data, losing generic, comprehensive visual representations learned from large-scale dataset [9]. Meanwhile, our model with frozen backbone provides better generalizability with large trade-offs between training and validation accuracies. The results reveal that learning new appearances under limited supervision requires understanding their ‘relations’ to diverse visual patterns acquired from a vast amount of past experiences, e.g., ImageNet classification. This is quite analogous to human vision perspective in the sense that we generalize novel concepts (what we see) by analyzing their relations to the past observations (what we know) [40].

For additional experimental details, results and analyses, we refer the readers to our supplementary materials.

6. Conclusion

We have presented a novel framework that analyzes complex feature correlations in a fully-convolutional manner using light-weight 4D convolutions. The significant performance improvements on three standard benchmarks demonstrate that learning patterns of feature relations from multiple visual aspects is effective in fine-grained segmentation under limited supervision. We also demonstrated a unique way of discarding insignificant weights leads to an efficient decomposition of a 4D kernel into a pair of 2D kernels, thus allowing extensive use of 4D conv layers at a significantly small cost. We believe our investigation will further facilitate the use of 4D convolutions in other domains that require learning to analyze high-dimensional correlations.

Acknowledgements. This work was supported by Samsung Advanced Institute of Technology (SAIT), the NRF grant (NRF-2017R1E1A1A01077999), and the IITP grant (No.2019-0-01906, AI Graduate School Program - POSTECH) funded by Ministry of Science and ICT, Korea.
References


[22] Shuaiyi Huang, Qiuyue Wang, Songyang Zhang, Shipeng Yan, and Xuming He. Dynamic context correspondence network for semantic alignment. In Proc. IEEE International Conference on Computer Vision (ICCV), 2019. 1, 2, 4


