Few-Shot Object Detection by Second-order Pooling (Supplementary Material)

Shan Zhang[1], Dawei Luo[2], Lei Wang[3], Piotr Koniusz[4,1](✉)

[1] Australian National University
[2] Beijing University of Posts and Telecommunications
[3] University of Wollongong
[4] Data61/CSIRO

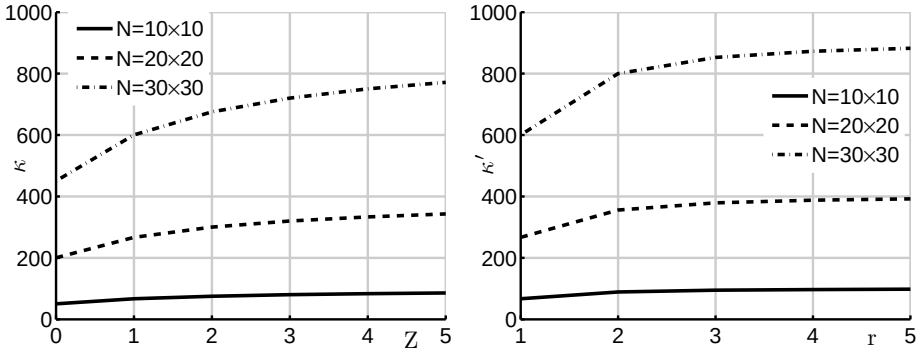
Power Normalization in Similarity Learning. Relation descriptors between query-support matrices $\boldsymbol{M}^*$ and $\boldsymbol{M}$ are formed by concatenation of $\boldsymbol{M}^*$ with $\boldsymbol{M}$ along the channel mode *e.g.* $\text{cat}(\boldsymbol{M}^*, \boldsymbol{M}) \in \mathbb{R}^{2\times K\times K}$ with the goal of similarity learning by SN. We also note that $\boldsymbol{M}$ may be obtained by for instance the mean between $\boldsymbol{M}^1, ..., \boldsymbol{M}^Z$ belonging to the same episode and class ($Z>1$ for few-shot case).

It is known from [1] that the Power Normalization in Eq. (1) (the main submission) performs a co-occurrence detection rather than counting. For classification problems, assume a probability mass function $p_{X_{ij}}(x) = 1/(N+1)$ if $x = 0, ..., N$, $p_{X_{ij}}(x) = 0$ otherwise, that tells the probability that co-occurrence between $\boldsymbol{\Phi}_{in}$ and $\boldsymbol{\Phi}_{jn}$ happened $x = 0, ..., N$ times in some chosen image region with $N = WH$ feature vectors. Note that classification often depends on detecting a co-occurrence (*e.g.*, is there a flower co-occurring with a pot?) rather than counts (*e.g.*, how many flowers and pots co-occur?). Using second-order pooling without PN requires a classifier to observe $N+1$ tr. samples of *flower and pot* occurring in quantities $0, ..., N$ to memorise all possible occurrence configurations. For relation learning, we stack pairs of samples to compare, thus a comparator now has to deal with a probability mass function of $R_{ij} = X_{ij} + Y_{ij}$ depicting *flowers and pots* whose $\text{support}(p_{R_{ij}}) = 2N+1 > \text{support}(p_{X_{ij}}) = N+1$ if random variable $X = Y$ (same class). For Z-shot learning, the support equals $(Z+1)N+1$ and the variance grows further indicating that the comparator has to memorize more configurations of co-occurrence (i, j) as Z grows.

However, this situation is alleviated by Power Normalization (the SigmE operator), whose probability mass function can be modeled as $p_{X^\eta_{ij}}(x) = 1/2$ if $x = \{0, 1\}$, $p_{X^\eta_{ij}}(x) = 0$ otherwise, as PN detects a co-occurrence (or its lack). For Z-shot learning, $\text{support}(p_{R^\eta_{ij}}) = Z+2 \ll \text{support}(p_{R_{ij}}) = (Z+1)N+1$. The following ratio

$$\kappa = \frac{\text{support}(p_{R_{ij}})}{\text{support}(p_{R^\eta_{ij}})} = \frac{(Z+1)N+1}{Z+2} \tag{1}$$

shows that the comparator has to memorize many more configurations of co-occurrence (i, j) for naive pooling compared to PN as Z and/or N grow (N depends on the width and height H and W of a chosen region). Figure 1a shows the ratio of required memorization of no-PN case divided by the PN case (in



(a) Memorization in RD (b) Memorization in HA

Fig. 1: Evaluations of the effect of PN on required memorization for Relationship Descriptors (RD) which in our case are a simple concatenation, and Hyper Attention (HR) in HARPN.

similarity learning) as a function of Z-shot number given various region sizes $N=WH$. If $Z=0$, the case illustrates regular classification. Clearly, the effect of required memorization is exacerbated more in FSL than classification especially for larger $Z \gg 1$. For the PN-based variant, reduction in required memorization is equivalent of limiting the family of functions during similarity learning.

Our modeling assumptions are very basic *e.g.*, the assumption on mass functions with uniform probabilities, the use of the support of mass functions rather than variances to describe variability of co-occurrence (i, j). Yet, substituting these modeling choices with more sophisticated ones does not affect theoretical conclusions that: (i) PN benefits few-shot learning ($Z \geq 1$) more than the regular classification ($Z=0$) in terms of reducing possible configurations of (i, j), and (ii) for variable size regions (varying N), PN reduces the number of configurations of (i, j) irrespective of value of N which is beneficial for forming relations between query-support ROIs of different sizes. While classifiers and comparators do not learn exhaustively all configurations of co-occurrence (i, j) as they have some generalization ability, they should learn quicker if the number of configurations of (i, j) is limited given the low-sample few-shot learning regime.

Power Normalization in Hyper Attention RPN. Firstly, let us explain the role of Second-order Self Correlation (SOSC) from Eq. (4) (the main submission) given by $\boldsymbol{a}_{\text{SOSC+PN}} = \mathcal{G}_{\text{SigmE}}(\mathbf{M}\cdot\mathbf{1}/K; \eta)$. Consider the effect of row-wise averaging:

$$\mathbf{M}\cdot\mathbf{1}/K = \frac{1}{NK}\sum_{n=1}^{N}\sum_{k=1}^{K}\boldsymbol{\phi}_n\phi_{kn} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}_n\mu_n, \tag{2}$$

where $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_N]$ and $\mu_n = \sum_{k=1}^{K}\phi_{kn}$. Eq. (2) captures correlation of each feature i in ϕ_{in} with itself and other channels, as expressed by μ_n. Thus, one

can think of Eq. (2) as capturing self-correlation of feature i together with its spread to other channels.

Below we show that not using PN in HARPN has a detrimental effect on learning in RPN due to larger number of feature variations which require more memorization capacity from network and thus a larger set of functions a classifier realizes. Increasing the set of functions while keeping the fixed number of training samples is a conceptually bad idea. Assume that each feature in the support and query maps of HARP can take a value in $\{0,1\}$ (to simplify assumptions). To describe the support feature i (spatial positions are factored out by the average pooling), let a probability mass function $p_{X_i}(x) = 1/(N+1)$ if $x = 0, ..., N$, $p_{X_i}(x) = 0$ otherwise, that tells the probability of co-occurrence between Φ_{in} and $(\Phi_{1n} + ... + \Phi_{Kn})/K = \mu_n,\ \forall n \in \{1, ..., N\}$. Note that we just assume here naively that $\mu_n \in \{0,1\}$ for simplicity. To describe the query feature i (at some spatial location), assume $p_{Y_i}(x) = 1/2$ as $x \in \{0,1\}$. Then, each spatial location in the cross-correlated attention map is described by a probability mass function of $A_i = X_i \cdot Y_i$ which results in $\text{support}(p_{A_i}) = \text{support}(p_{X_i}) = N+1$.

However, using PN on Eq. (2) turns this equation into a feature detector with $p_{X'_i}(x) = 1/2$ as $x \in \{0,1\}$ (co-occurrence detection or lack of it). Then, each spatial location in the cross-correlated attention map is described by a probability mass function of $A'_i = X'_i \cdot Y_i$ which results in $\text{support}(p_{A'_i}) = \text{support}(p_{X'_i}) = 1/2$.

Finally, assume that the cross-correlated query against the support feature map is convolved with a filter of side size r which can take values in $\{-1, 0, 1\}$ (also to simplify the argumentation) before being passed through a non-linearity (the form of an intermediate decision boundary) to take some intermediate decision regarding a region proposal. Each location of the filter can be described by $p_{F_i}(x) = 1/3$ for $x \in \{-1, 0, 1\}$ and $\text{support}(p_{F_i}) = 1/3$. Not surprisingly, a probability mass function of $D_i = A_i + \sum_{i \in \mathcal{I}_{r^2}} F_i$ yields $\text{support}(p_{D_i}) = 2Nr^2 + 1$ while for $D'_i = A'_i + \sum_{i \in \mathcal{I}_{r^2}} F_i$, we have $\text{support}(p_{D'_i}) = 2r^2 + 1$. The following ratio

$$\kappa' = \frac{\text{support}(p_{D_i})}{\text{support}(p_{D'_i})} = \frac{2Nr^2 + 1}{2r^2 + 1} \tag{3}$$

shows that the covolutional comparator with a filter of side size r has to memorize many more configurations if PN is not used by the Hyper Attention. Figure 1b shows the ratio of required memorization of no-PN case divided by the PN case (for RPN learning for a single conolution) as a function of filter side size r and support crop sizes $N = WH$. Clearly, the effect of required memorization is exacerbated for typical filter sizes $r = 3$. For the PN-based variant, reduction in the required memorization is equivalent of limiting the family of functions during learning region proposals which has a regularization effect on the learner.

Furthermore, to describe the support feature i, one may consider that each μ_n in fact may have a probability mass function $p_Z(x) = 1/(K+1)$ if $x = 0, ..., K$, $p_Z(x) = 0$ otherwise. Therefore, the probability mass $X''_i = X_i \cdot Z$ has $\text{support}(p_{X''_i}) = NK + 1$ which is a more realistic modeling of set support for co-occurrence between Φ_{in} and $(\Phi_{1n} + ... + \Phi_{Kn})/K = \mu_n,\ \forall n \in \{1, ..., N\}$. Having $A''_i = X''_i \cdot Y_i$ and $D''_i = A''_i + \sum_{i \in \mathcal{I}_{r^2}} F_i$, we get $\text{support}(p_{D''_i}) = 2NKr^2 + 1$ which

yields the following ratio

$$\kappa'' = \frac{\text{support}(p_{D_i''})}{\text{support}(p_{D_i'})} = \frac{2NKr^2+1}{2r^2+1}, \tag{4}$$

which shows that as $k \gg 1$, not using PN in HARPN would result in even more need for memorization than Eq. (3) suggests.

HARPN network/difference with ARPN. HARPN/ARPN use RPN from the Faster R-CNN detector. As Fig. 3 and 4 (main paper) show, Hyper Attention uses our SOSD or SOSC to form attention-modified Query Feature Map (QFM). SOSC acts as 'feature detector' that also captures the feature spread between channels (operation $\mathbf{M}\cdot\mathbf{1}$). ARPN uses average pooling–a mere 'feature counter' that increases learning uncertainty. Eq. (1) and (3) show that SOSC limits uncertainty of cross-correlating support against query, thus improving the quality of QFM and region proposals.

References

1. Koniusz, P., Zhang, H., Porikli, F.: A deeper look at power normalizations. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, IEEE Computer Society (2018) 5774–5783