Detect Any Keypoints: An Efficient Light-Weight Few-Shot Keypoint Detector (Supplementary Material)

Changsheng Lu[1], **Piotr Koniusz**[2,1]

[1]The Australian National University [2]Data61/CSIRO
changshengluu@gmail.com, piotr.koniusz@data61.csiro.au

A. Summary of Methodology

A.1 Architecture Overview

The goal of few-shot keypoint detection (FSKD) is to predict the corresponding keypoints in query image given the support image & support keypoints. Via such setting, the FSKD model becomes very general and being able to detect arbitrary kinds of keypoints as long as providing the supports.

The overview of our lightweight FSKD model is shown in Fig. 1. As we can see, our model is comprised of five components for keypoint inference, which are weight-shared encoder $\mathcal{F}$, keypoint feature aggregator $\mathcal{A}$, non-linear kernel generator (KG) $\mathbf{\Pi}$, non-parametric detection module $\mathcal{D}$, and upsampling modules $\mathcal{U}$. Each module is meticulously designed to endow the model to be lightweight while extremely efficient. In training stage, the mean feature based contrastive learning (MFCL) is applied as a regularization loss, to enforce the feature learning and align the keypoint representation distributions. A glance of keypoint feature distribution with and without MFCL in training can be found in Fig. **??**.

We instantiate the non-linear kernel generator (KG) as *space-channel disentangled* refinement network, as shown in Fig. 1(b). The non-linear KG is responsible to map the support keypoint prototype (SKP) into multi-group kernels with diverse resolutions, which aims to improve *correlation window* during simultaneous modulation and detection. The SRM and CRM in Fig. 1(b) refers to *space refinement module* (SRM) and *channel refinement module* (CRM), respectively. Via this disentangled way of generating kernels, our model could save parameters and achieve high-efficiency. From the Table 1 (in main paper), we can observe that the model parameters for generating kernels with resolution $S = \{1, 3, 5\}$ are similar, only consuming 27.5M, 27.6M and 27.6M, which manifests the effective design of non-linear KG.

A.2 Improving FSKD by Contrasting Keypoints

To encourage our lightweight FSKD model to learn representative keypoint representations, we propose to equip our model with mean feature based contrastive learning (MFCL). Even though we discovered that the MFCL could help reach higher performance, we try to investigate and understand following questions:

Q1: *Why mean feature based contrastive learning is important for FSKD?* Foremost, contrastive learning (CL) can help align the distribution of same type of keypoint features across difference species. By introducing the CL loss, the model is not only learning towards the goal of class-agnostic keypoint localization, but also takes care of feature representation learning and encourages better manifold. This is the benefit of CL. However, when applying CL to the task of few-shot learning, one major concern is the limited number of support samples, which would cause instance-level keypoint features to be noisy and less representative, thus being sub-optimal for contrastive learning. Instead, using the mean feature over an episode, and then performing CL in episode level will benefit FSKD as mean keypoint features are more representative and more stable than instance-level features.

Q2: *How to control hardness of negative keypoints for contrastive learning? Why we can control?* The hardness of negative keypoints can be tuned by setting $(\alpha, \rho, N_{\text{neg}})$, as shown in Fig. 2. If the bounding box scale α is small, then bounding box shrinks towards the center of object, which would lead bounding box falling more on the foreground. Then sampling negatives within bounding box would lead most of the negatives coming from object foreground region, thus increasing the hardness. If setting α to big, bounding box becomes larger and more background region is incorporated, thus lowering the hardness, as background usually is regarded as the source coming easy negatives. For distance threshold ρ, it is easy to understand that if ρ is small, then negative keypoints are closer to anchors, thus increasing the hardness. If ρ is big, vice versa. The N_{neg} controls the number of negative keypoints sampled from bounding box. Obviously, it controls density. By setting appropriate $(\alpha, \rho, N_{\text{neg}})$, it would boost performance with MFCL, as shown in Table 4 (ablation study in main paper).

A.3 Proposition 1 & Proof

The goal of Proposition 1 is to give an insight that our simultaneous modulation and detection (SMD) has potential to achieve equal feature modeling ability compared to modulation-detection separate (MDS) design. It indicates our model may not lose performance even if combining modulation and detection into one step for lightweight.

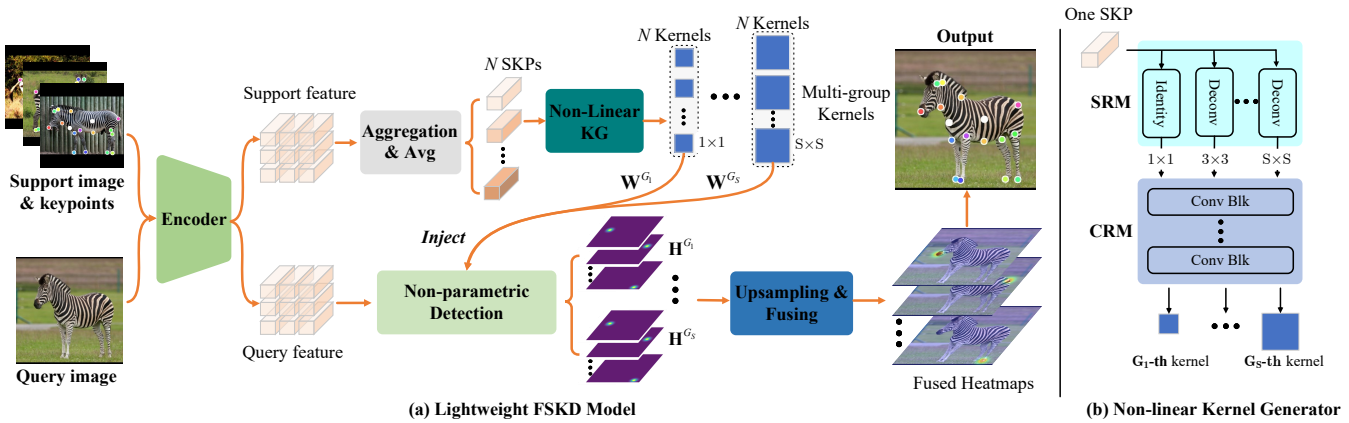

Figure 1: The pipeline of proposed lightweight few-shot keypoint detection. Our model takes both support prompts and query image as input, and outputs the predicted keypoints for query image. The support keypoints are used to aggregate the keypoint features and build support keypoint prototypes (SKPs). Via the non-linear kernel generator (KG), the SKPs are refined and converted into multi-group kernels which will be injected into detection module for simultaneous modulation and detection (SMD). The non-linear KG includes SRM and CRM which are responsible to refine the *space* and *channel* of the generated kernels, respectively. The design of our model could significantly reduce the memory consumption while keeping FSKD performance.

Figure 2: Examples of negative keypoints control. White circles are negative keypoints while red ones are anchors. (a) Bounding box scale ranging in 0.8, 1.15, 1.5, 2.0; (b) Distance threshold ranging in 10, 30, 40, 60 pixels; (c) Number of negative keypoints ranging in 5, 10, 20, 30. We can clearly see that the hardness are controlled by bounding box scale and distance threshold, while the density is controlled by number of keypoints.

Since neural network is non-linear yet complex, it is very hard to analyze its properties inside the black box. However, one can use Taylor expansion to probe its properties by focusing on tiny intervals at some points, *e.g.* the works (Balduzzi, McWilliams, and Butler-Yeoman 2017; Alain and Bengio 2014), which use first-order or second-order Taylor approximation. Analogously, we give the proof as follows.

Proof. Recall that the formulation of softplus is $\phi(x) = \frac{1}{T}\ln(1+e^{Tx})$ (*i.e.*, soft ReLU) which is continuously differentiable and approximates ReLU with arbitrary precision based on temperature T. By using Taylor expansion, one can rewrite the difference between $\mathbf{H}_f$ and $\mathbf{H}_g$ as follows

$$\|\mathbf{H}_f - \mathbf{H}_g\|_2 = \|\mathbf{W}_{1,2}\phi(\mathbf{W}_{1,1}\mathbf{Z}) - \frac{1}{c}g(\mathbf{a})^\intercal\mathbf{X}\|_2$$
$$= \|\mathbf{W}_{1,2}\phi(\varepsilon+\mathbf{W}_{1,1}\mathbf{Z}-\varepsilon) - \frac{1}{c}\big(\mathbf{W}_{2,2}\phi(\varepsilon+\mathbf{W}_{2,1}\mathbf{a}-\varepsilon)\big)^\intercal\mathbf{X}\|_2$$
$$\approx \|\mathbf{W}_{1,2}\big(\phi(\varepsilon)+\phi'(\varepsilon)(\mathbf{W}_{1,1}\mathbf{Z}-\varepsilon)+\frac{1}{2}\phi''(\varepsilon)(\mathbf{W}_{1,1}\mathbf{Z}-\varepsilon)^2\big) -$$
$$\frac{1}{c}\big(\mathbf{W}_{2,2}\big(\phi(\varepsilon)+\phi'(\varepsilon)(\mathbf{W}_{2,1}\mathbf{a}-\varepsilon)+\frac{1}{2}\phi''(\varepsilon)(\mathbf{W}_{2,1}\mathbf{a}-\varepsilon)^2\big)\big)^\intercal\mathbf{X}\|_2$$
$$= \|\mathbf{a}^\intercal\big(\mathrm{Diag}(\mathbf{W}_{\text{mds}}) - \frac{1}{c}\mathbf{W}_{\text{smd}}^\intercal\big)\mathbf{X}\|_2$$

For soft ReLU at $\varepsilon = 0$ and $T\to\infty$, the first-order derivative equals 0.5, and first-order Taylor expansion holds. Moreover, when expanding at $\varepsilon > 0$ for $T\to\infty$, then $\phi(\varepsilon) = \varepsilon$, $\phi(\varepsilon)' = 1$, and n-th order derivative $\phi(\varepsilon)^{(n)} = 0$ ($n \geq 2$). The high-order Taylor expansion holds. □

Remark: Our SMD design can approximate MDS closely, and can correlate support keypoint prototype $\mathbf{a}$ with query feature $\mathbf{X}$ via $\mathbf{W}_{\text{smd}}$, which benefits keypoint location inference. We also conduct a toy experiment to observe the heatmap difference (*i.e.*, $\|\mathbf{H}_f - \mathbf{H}_g\|_2$) when *respectively fitting* SMD and MDS networks to randomly sampled heatmap given same inputs (1000 trials). Fig. 3 shows that the difference of heatmaps dramatically reduces and limits to zero over a few optimization iterations, which shows that SMD can approximate MDS well.

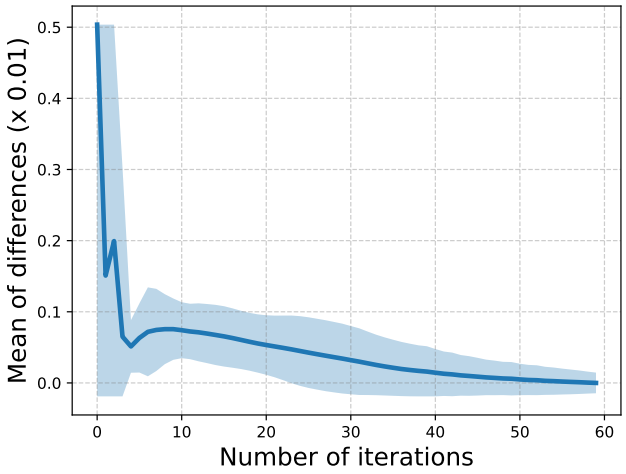

Figure 3: Toy experiment on the observation of heatmap difference between SMD and MDS networks.

References

Alain, G.; and Bengio, Y. 2014. What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 15(1): 3563–3593.

Balduzzi, D.; McWilliams, B.; and Butler-Yeoman, T. 2017. Neural taylor approximations: Convergence and exploration in rectifier networks. In *International conference on machine learning*, 351–360. PMLR.