# Supplemental Materials: Distilling Object Detectors with Global Knowledge

Sanli Tang[1⋆], Zhongyu Zhang[1⋆], Zhanzhan Cheng[1†], Jing Lu[1], Yunlu Xu[1], Yi Niu[1], and Fan He[2]

[1] Hikvision Research Institute, Hanzhou, China
[2] Institute of Image Processing and Pattern Recognition, Shanghai Jiao Tong University, Shanghai, China
{tangsanli,zhangzhongyu,chengzhanzhan,lujing6,xuyunlu,niuyi}@hikvision.com
hf-inspire@sjtu.edu.cn

## 1 Prototype Selection in PGM

The prototype generation module (PGM) selects the prototypes by minimizing reconstruction errors in two feature spaces and the discrepancy of representations w.r.t. the prototypes as defined in Eq. 2 in the main manuscript. In this section, a solution is given through a variant of matching pursuit [4], which is an effective and efficient approach. For the sake of clarity, we quickly review the notations in the main manuscript. The features of $N$ instances are $\boldsymbol{F}_t = \{\boldsymbol{f}_i^t\}_{i=1}^N$ and $\boldsymbol{F}_s = \{\boldsymbol{f}_i^s\}_{i=1}^N$ in the feature spaces of the teacher and the student detectors, respectively. The $\boldsymbol{G}_t = \{\boldsymbol{g}_k^t\}_{k=1}^K$ and $\boldsymbol{G}_s = \{\boldsymbol{g}_k^s\}_{k=1}^K$ are the features of $K$ prototypes in the two feature spaces, respectively. Thus, the objective in Eq. 2 in the main manuscript can be rewritten as:

$$\begin{aligned}\mathcal{L} &= ||\boldsymbol{F}_t - \boldsymbol{G}_t\boldsymbol{W}_t|| + ||\boldsymbol{F}_s - \boldsymbol{G}_s\boldsymbol{W}_s|| + \lambda||\boldsymbol{W}_s - \boldsymbol{W}_t||_2^2 \\ &= \sum_{i=1}^{N}||\boldsymbol{f}_i^t - \sum_{k=1}^{K}\boldsymbol{g}_k^t w_{k,i}^t||_2^2 + \sum_{i=1}^{N}||\boldsymbol{f}_i^s - \sum_{k=1}^{K}\boldsymbol{g}_k^s w_{k,i}^s||_2^2 \\ &+ \lambda\sum_{i=1}^{N}\sum_{k=1}^{K}(w_{k,i}^t - w_{k,i}^s)^2.\end{aligned} \tag{1}$$

$(\boldsymbol{g}_k^t, \boldsymbol{g}_k^s) \in \{(\boldsymbol{f}_i^t, \boldsymbol{f}_i^s)\}_{i=1}^N$, $\forall k = 1, \cdots, K$. $\boldsymbol{W}_t = \{w_{k,i}^t\}_{K\times N}$ and $\boldsymbol{W}_s = \{w_{k,i}^s\}_{K\times N}$ are coefficient matrices as representations of instances w.r.t. the prototypes. In PGM, prototypes are selected in a greedy manner that only one prototype is selected at each step by minimizing Eq. 1. Concretely, we follow the definition of residuals $\boldsymbol{r}_{n,i}^t$ and $\boldsymbol{r}_{n,i}^s$ as the Eq. 4 of the main manuscripts as follows:

$$\boldsymbol{r}_{n,i}^t \triangleq \boldsymbol{f}_i^t - \sum_{k=1}^{n}\boldsymbol{g}_k^t w_{k,i}^t, \quad \boldsymbol{r}_{n,i}^s \triangleq \boldsymbol{f}_i^s - \sum_{k=1}^{n}\boldsymbol{g}_k^s w_{k,i}^s, \tag{2}$$

⋆ Authors contributed equally. † Corresponding authors.

where $n$ is number of selected prototypes after $n$ steps. Thus, at the $(n+1)^{\text{th}}$ step, the prototype can be selected by minimizing $\mathcal{L}_{n+1}$, where we expand the Eq. 4 in the main manuscripts as follows:

$$\begin{aligned}\mathcal{L}_{n+1} &= \sum_{i=1}^{N} ||\boldsymbol{r}_{n+1,i}^t||_2^2 + \sum_{i=1}^{N} ||\boldsymbol{r}_{n+1,i}^s||_2^2 + \lambda \sum_{i=1}^{N}\sum_{k=1}^{n+1}(w_{k,i}^t - w_{k,i}^s)^2 \\ &= \sum_{i=1}^{N} ||\boldsymbol{r}_{n,i}^t - \boldsymbol{g}_{n+1}^t w_{n+1,i}^t||_2^2 + \sum_{i=1}^{N} ||\boldsymbol{r}_{n,i}^s - \boldsymbol{g}_{n+1}^s w_{n+1,i}^s||_2^2 \\ &+ \lambda \sum_{i=1}^{N}\sum_{k=1}^{n+1}(w_{k,i}^t - w_{k,i}^s)^2,\end{aligned} \tag{3}$$

which is the objective $\mathcal{L}$ in the $(n+1)^{\text{th}}$ iteration. The optimal $w_{n+1,i}^t$ and $w_{n+1,i}^s$ can be calculated by the derivatives:

$$\begin{aligned}\frac{\partial \mathcal{L}_{n+1}}{\partial w_{n+1,i}^t} &= \frac{\partial ||\boldsymbol{r}_{n+1,i}^t||_2^2}{\partial w_{n+1,i}^t} + \frac{\partial \lambda (w_{n+1,i}^t - w_{n+1,i}^s)^2}{\partial w_{n+1,i}^t} \\ &= -2\left\langle \boldsymbol{r}_{n,i}^t, \boldsymbol{g}_{n+1}\right\rangle + 2||\boldsymbol{g}_{n+1}^t||^2 w_{n+1,i}^t \\ &+ 2\lambda(w_{n+1,i}^t - w_{n+1,i}^s),\end{aligned} \tag{4}$$

where the $\langle \cdot, \cdot \rangle$ is the inner product of two vectors. Let the derivative $\frac{\partial \mathcal{L}}{\partial w_{n+1,i}^t}$ be zero and we have

$$w_{n+1,i}^t = \frac{\left\langle \boldsymbol{r}_{n,i}^t, \boldsymbol{g}_{n+1}^t \right\rangle + \lambda w_{n+1,i}^s}{\lambda + ||\boldsymbol{g}_{n+1}^t||_2^2}. \tag{5}$$

Similarly, for $w_{n+1,i}^s$, we have

$$w_{n+1,i}^s = \frac{\left\langle \boldsymbol{r}_{n,i}^s, \boldsymbol{g}_{n+1}^s \right\rangle + \lambda w_{n+1,i}^t}{\lambda + ||\boldsymbol{g}_{n+1}^s||_2^2}. \tag{6}$$

We simplify the notations of the inner-product between the residual and the prototype as $\Delta_n^s \triangleq \left\langle \boldsymbol{r}_{n,i}^s, \boldsymbol{g}_{n+1}^s \right\rangle$ and $\Delta_n^t \triangleq \left\langle \boldsymbol{r}_{n,i}^t, \boldsymbol{g}_{n+1}^t \right\rangle$ in *TS-space*. By considering $w_{n+1,i}^t$ and $w_{n+1,i}^s$ in both Eq. 5 and Eq. 6, they can be calculated w.r.t. $\boldsymbol{g}_{n+1}^t$ and $\boldsymbol{g}_{n+1}^s$ as followings:

$$\begin{aligned}w_{n+1,i}^t &= \frac{\lambda\left(\Delta_n^t + \Delta_n^s\right) + \Delta_n^t ||\boldsymbol{g}_{n+1}^s||_2^2}{\lambda\left(||\boldsymbol{g}_{n+1}^t||_2^2 + ||\boldsymbol{g}_{n+1}^s||_2^2\right) + ||\boldsymbol{g}_{n+1}^t||_2^2 ||\boldsymbol{g}_{n+1}^s||_2^2}, \\ w_{n+1,i}^s &= \frac{\lambda\left(\Delta_n^t + \Delta_n^s\right) + \Delta_n^s ||\boldsymbol{g}_{n+1}^t||_2^2}{\lambda\left(||\boldsymbol{g}_{n+1}^t||_2^2 + ||\boldsymbol{g}_{n+1}^s||_2^2\right) + ||\boldsymbol{g}_{n+1}^t||_2^2 ||\boldsymbol{g}_{n+1}^s||_2^2}.\end{aligned} \tag{7}$$

Therefore, by substituting $w_{n+1,i}^t$ and $w_{n+1,i}^s$ into Eq. 3, the $(n+1)$ prototype can be selected with the minimum value of $\mathcal{L}_{n+1}$. This can be implemented efficiently by parallel computation on GPUs. The above process will be iterated by $K$ times to obtain $K$ prototypes. The overall algorithm for selecting prototypes in PGM is summarized in Alg. 1 as in the main manuscript.

Notice by setting $\lambda = 0$ that means to ignore the representation similarity of instances w.r.t the prototypes in *TS-space*, the calculations of the optimal $w^t_{n+1,i}$ and $w^s_{n+1,i}$ in Eq. 7 can be much simplified as followings:

$$w^t_{n+1,i} = \frac{\Delta^t_n}{||\boldsymbol{g}^t_{n+1}||^2_2}, \quad w^s_{n+1,i} = \frac{\Delta^s_n}{||\boldsymbol{g}^s_{n+1}||^2_2}, \tag{8}$$

which are indeed the relations, *i.e.*, the inner product, between the residuals and the prototypes. In particular, the coefficient of $w_{0,i}$ is the relation between the prototypes and the other instances.

## 2 Implementation Details and Experiments

In this section, we show more implementation details, though codes are appended in the supplemental assets. Besides, we show more experiments and analysis of the proposed method.

### 2.1 More Implementation Details

We implement our distilling framework through the public benchmark MMDetection [1] based on PyTorch[3]. The Faster R-CNN [6] and RetinaNet [3] are well-implemented in MMDetection and used in our experiments. For the VOC dataset, we train the bare student and teacher detectors from the ImageNet [7] pre-trained models. For the COCO dataset, we directly use the public well-trained models from the Model Zoo [1] of MMDetection. Then, the teacher and student detectors are trained by minimizing the objective of knowledge distillation in Eq. 9 in the main manuscript. For various foreground classes and feature maps of multiple resolutions, we separately apply the PGM for each class and each feature map to be distilled, and so is the RDM. The code is available and appended in the supplemental materials.

### 2.2 More Experiments and Analysis

**Prototypes analysis**. In Fig. 1, we illustrate some prototypes of the class *aeroplane*, *car* and *person* as well as some instances with their distilling weights $\sigma$ defined in Eq. 8 and predicted confidence of the teacher detector. Although some of the detection errors are predicted by the high confidence of the teacher detector, the distillation weights are relatively low by measuring the discrepancy of representations w.r.t. the prototypes. Hence, the noisy knowledge is hard to be transferred to the student detector. The quantitative analysis of the noisy knowledge transferring, i.e., the transferring ratios of the true positives and false positives, is illustrated in Table 9 in the main manuscripts.

**Analysis of noisy knowledge transferring**. We find that although some of the detection errors are predicted by the teacher detector with high confidence,

[3] https://pytorch.org

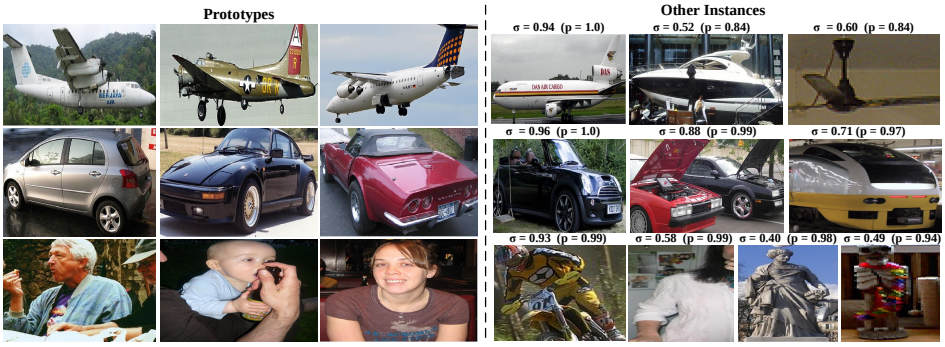


**Fig. 1.** Illustration of the prototypes (left) as well as other instances (right) with the distilling weights $\sigma$ in Eq. 8 in the main manuscript and the predicted confidence $p$ of the teacher detector.

**Table 1.** Noise analysis on different distillation methods. The higher $TR_{\text{TP}}$, the more useful knowledge transferred. The higher $TR_{\text{FP}}$, the more noisy knowledge eliminated.

| Method | FGFI [8] | Mimicking [2] | RKD† [5] | Ours |
|---|---|---|---|---|
| $TR_{\text{TP}}$ | 0.08 | 0.05 | 0.52 | **0.60** |
| $TR_{\text{FP}}$ | 1.96 | 1.89 | 2.01 | **2.20** |

the distillation weights are relatively low by measuring the discrepancy. Hence, the noisy knowledge is hard to be transferred to the student detector. The visualization of prototypes and the qualitative analysis can be found in supplemental materials. Furthermore, we also measure the distillation performance by the knowledge transferring ratio as shown in Table 1 for quantitative analysis. The transferring ratios of true positives and false positives are measured as $\text{TR}_{\text{TP}}$ and $\text{TR}_{\text{FP}}$ in Eq. 9.

$$TR_{\text{TP}} = \frac{\sum_i \sum_{\tilde{b}_{i,j} \cap \boldsymbol{b}_i > \tau} (p^{\text{kd}}_{i,j} - p^{s}_{i,j})}{\sum_i \sum_{\tilde{b}_{i,j} \cap \boldsymbol{b}_i > \tau} (p^{t}_{i,j} - p^{s}_{i,j})}, \quad TR_{\text{FP}} = \frac{\sum_i \sum_{\tilde{b}_{i,j} \cap \boldsymbol{b}_i < \tau} (p^{s}_{i,j} - p^{\text{kd}}_{,i,j})}{\sum_i \sum_{\tilde{b}_{i,j} \cap \boldsymbol{b}_i < \tau} (p^{s}_{i,j} - p^{t}_{i,j})}. \tag{9}$$

$p^{t}_{i,j}$, $p^{s}_{i,j}$ and $p^{\text{kd}}_{i,j}$ are the confidences of the teacher, the base student and the distilled student detector for the predicted box $\tilde{b}_{i,j}$, respectively. $\boldsymbol{b}_i$ are the ground truth boxes of the $i$-th image and $\tilde{b}_{i,j} \cap \boldsymbol{b}_i$ means the maximum IoU between the $\tilde{b}_{i,j}$ and each box in $\boldsymbol{b}_i$. $\tau$ is the IoU threshold and set to 0.5. The higher $TR_{\text{TP}}$, the more useful knowledge of the true positives (TP) from the teacher detector is transferred. The higher $TR_{\text{FP}}$, the more noisy knowledge of the false positives (FP) from the teacher detector is eliminated.

In Table 1, we show that the proposed method achieves the highest $TR_{\text{TP}}$ and $TR_{\text{FP}}$ on PASCAL VOC dataset. Notice that the $TR_{\text{FP}}$ is larger than 1.0, which means there are many false positives predicted by the teacher leading to a lower denominator.

**Further fine-tuning the student without the distillation loss**. Since the proposed distilling method is trained based on detectors being pre-trained from the task-relevant datasets, *e.g.*, the PASCAL VOC or COCO, the performance might be improved due to more training iterations by minimizing the detection loss $\mathcal{L}_{\text{det}}$ rather than distilling loss in Eq. 10. To dissipate such suspicion, we set the weights $\alpha_1 = \alpha_2 = \alpha_3 = 0$ to make the global and local knowledge distilling loss to zero and train the detector by only minimizing the $\mathcal{L}_{\text{det}}$. The same settings are applied, e.g., the learning schedule, as in knowledge distillation. We find that the mAP (81.4) is very closed to the bare student detector (81.3). However, the proposed method shows 82.9% mAP on the VOC dataset, which proves that the performance gain is obtained by the proposed distilling algorithm rather than simply fine-tuning the student with more epochs.

**Analysis on training efficiency**. Generating the prototypes in PGM is a relatively time-consuming process in complexity of $\mathcal{O}(NK)$ for each class. It mainly consists of two steps: extracting the features of instances in *TS-spaces* and applying the Alg.1 in the main manuscript to select the prototypes. Many tricks can be developed to improve the efficiency: (1) The features of instances in the feature space of teacher detector can be extracted once and stored since the teacher detector is unchanged when distilling the detector. (2) All the computations in Alg. 1 in the main manuscript can be implemented efficiently by several parallel matrix operations on GPUs. (3) The features of instances in the feature space of student detector can be bootstrapped and updated per epoch, which shows little harm for the distillation by referring to results in Table 5 in the main manuscript. We evaluate the time consumed by the PGM, which will slight increase about 10% of the overall training process. It could be more efficient through more delicate implementation *e.g.*, selecting the prototypes on each class and feature maps in parallel. Notice that *the computational efficiency exactly remains the same as the bare student detector when inference.*

## References

1. Kai Chen, e.a.: Mmdetection: Open mmlab detection toolbox and benchmark. CoRR **abs/1906.07155** (2019)
2. Li, Q., Jin, S., Yan, J.: Mimicking very efficient network for object detection. In: CVPR. pp. 7341–7349 (2017)
3. Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV. pp. 2999–3007 (2017)
4. Mallat, S., Zhang, Z.: Matching pursuits with time-frequency dictionaries. TIP **41**(12), 3397–3415 (1993)
5. Park, W., Kim, D., Lu, Y., Cho, M.: Relational knowledge distillation. In: CVPR. pp. 3967–3976 (2019)
6. Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NeurIPS. pp. 91–99 (2015)
7. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Li, F.: Imagenet large scale visual recognition challenge. IJCV **115**(3), 211–252 (2015)
8. Wang, T., Yuan, L., Zhang, X., Feng, J.: Distilling object detectors with fine-grained feature imitation. In: CVPR. pp. 4933–4942 (2019)