

Vision-language는 큰 성능을 보여줌.

text class는 VLM의 text encoder의 prompt를 사용하여 생성

그 후 detector의 훈련을 위한 region classifier로 사용됨

↳ prompt가 중요함.

∴ DetPro 제안 → VLM 기반

1) image background를 prompt로 포함하는 background interpretation.

2) image foreground를 prompt training을 위한 분리

* Introduction

Object Detection(OD)는 closed-set의 categories 성능을 이룸.

Detection vocab을 늘리는 데, AI labeling 등 많은 연구가 필요하다.

다른 방법으로는 open-Vocabulary Object Detection(OVOD)가 있다.

VILDM에서 도입.

↳ class와 대한 설명을 CLIP이 도입. (=Prompt)

↳ CLIP Embedding으로 계층에 템즈 훈련

Prompt design은 성능에 중요.

∴ vocab tuning을 자동으로 학습할 수 있도록 함

논문에서는 이를 위해 OVOD-VLM을 사용한 DetPro 제안

Prompts의 특징은

1) negative proposal도 detection에 중요한지 영향을 주지 못하는 문제

2) classification은 디렉시 컨텍스트 연관. ∴ 여러 level을 배울 수도 있는 문제.

CLIPM 논문에서는

1) negative proposal 학습이 포함.

2) 다양화된 positive proposal 학습.

* Related Work.

- Prompt learning

CoOpM classification을 위한 방법

- OVOD

* Problem Setting.

OVOD-VLM의 prompt를 배우는 것은 기본

Positive, negative의 대안 loss를 찾.

↳ multi-content ↳ negative proposal

- Data split

Detection의 대안 것을 base class C_B et novel class C_N 으로 분리

X_T, X_I 는 training, Inference dataset

- Pre-trained VLM

Text encoder $T(\cdot)$ 과 Image encoder $I(\cdot)$ 은 구조된 CLIP을 사용

$T(\cdot)$ 은 prompt representation을 input으로 받고

$I(\cdot)$ 은 image를 input으로 받고, image embedding을 output.

- Detection Framework

Faster-RCNN At8

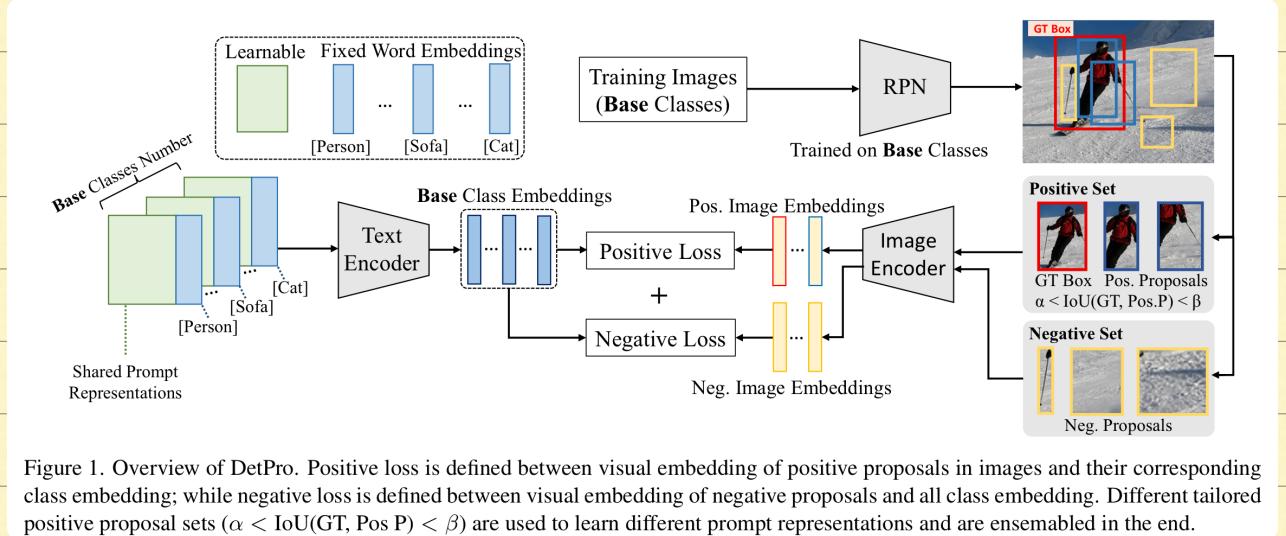


Figure 1. Overview of DetPro. Positive loss is defined between visual embedding of positive proposals in images and their corresponding class embedding; while negative loss is defined between visual embedding of negative proposals and all class embedding. Different tailored positive proposal sets ($\alpha < \text{IoU}(\text{GT}, \text{Pos.P}) < \beta$) are used to learn different prompt representations and are ensembled in the end.

* Method

- Preliminaries : Prompt

T이 들어갈 prompt를 만드는 것은 쉽지 않다. 따라서 학습을 할 때

given class $c \in C_B$ 의 대응 prompt representation V_c 는

$$V_c = [v_1, v_2 \dots v_L, w_c],$$

ex) $v_i = \text{'a'}$, 'photos' , 'of' ; $w_c = [\text{class}]$

Text encoding t_c

$$t_c = \mathcal{T}(V_c).$$

Image encoding f

$$f = I(I_{\text{img}})$$

$$p_c = \frac{\exp(\cos(f, t_c)/\tau)}{\sum_{i \in C_B} \exp(\cos(f, t_i)/\tau)}, \quad (3)$$

$$\mathcal{L}_p = -\log p_c.$$

- Detection prompt

- Naive Solution.

OD는 bounding box를 있고 class label을 예측하는 힌트로 classification과 다르다.

가장 간단한 방법은 GT bounding box를 $I(\cdot)$ 이 넓어 각각 f 를 얻는 것이다.

이런 naive solution은 시스템이 더 좋은 방법을 탐색해야 한다.

- Fine-grained Solution.

C_B 의 일부 x_T 로부터 RPN을 훈련시켜, foreground proposal F 와 background proposal B 를 얻을

ground truth $g + F = \text{Positive set } P$, negative set $N = B$

P 의 경우 배경도 많이 포함

$\therefore I(\cdot)$ 을 사용할 때 문제 발생

↳ context grading scheme.

N 의 경우 대체로 배경

Background (negative proposal) 추적의 경우 모호하기 때문에 embedding function text embedding의 어떤 부위도 빼지 않아야 함

f_n 이 특정 class c 가 될 확률 p_{nc} 는 작아야 한다.

↳ $\frac{1}{|C_B|}$ 은 gt로 균등하게 맞춤.

$$\mathcal{L}_n = - \sum_{c=1}^{|C_B|} w \log p_{nc}, \quad w = \frac{1}{|C_B|}. \quad (5)$$

다른 방법으로는 따로 bg token을 배우는 것

$$V_{bg} = [v_1^{bg}, v_2^{bg}, \dots, v_L^{bg}]. \quad \mathcal{L}_n = -\log p_{nbg}.$$

$$p_{nbg} = \frac{\exp(\cos(f_n, t_{bg})/\tau)}{\sum_{c=1}^{|C_B|} \exp(\cos(f_n, t_c)/\tau) + \exp(\cos(f_n, t_{bg})/\tau)}.$$

보통 첫번째 방법이 더 강력.

↳ 배경이 포함될 수도 있고, 첫번째는 배경이 각 class와 밀어치도록 학습하기 때문

- Context grading with tailored positive proposal.

positive IoU 전에 object가 포함된 사건의 경우 “~의 사건”이라고 할 수 있지만,

부분만 포함된 경우, “~의 부분 사건”이라고 할 수 있다.

∴ prompt level을 도입

IoU를 흡수하여 사건이 object가 포함된 정도의 group으로 나눔

이제 대신 f_p 의 loss를 사용하여 optimize = $-\log P_{pc}$

negative IoU 대비에도 비슷하게 동작

$$\mathcal{L} = \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} \mathcal{L}_n + \frac{1}{|\mathcal{P}^k|} \sum_{p \in \mathcal{P}^k} \mathcal{L}_p.$$

- Assembling DetPro onto ViLD

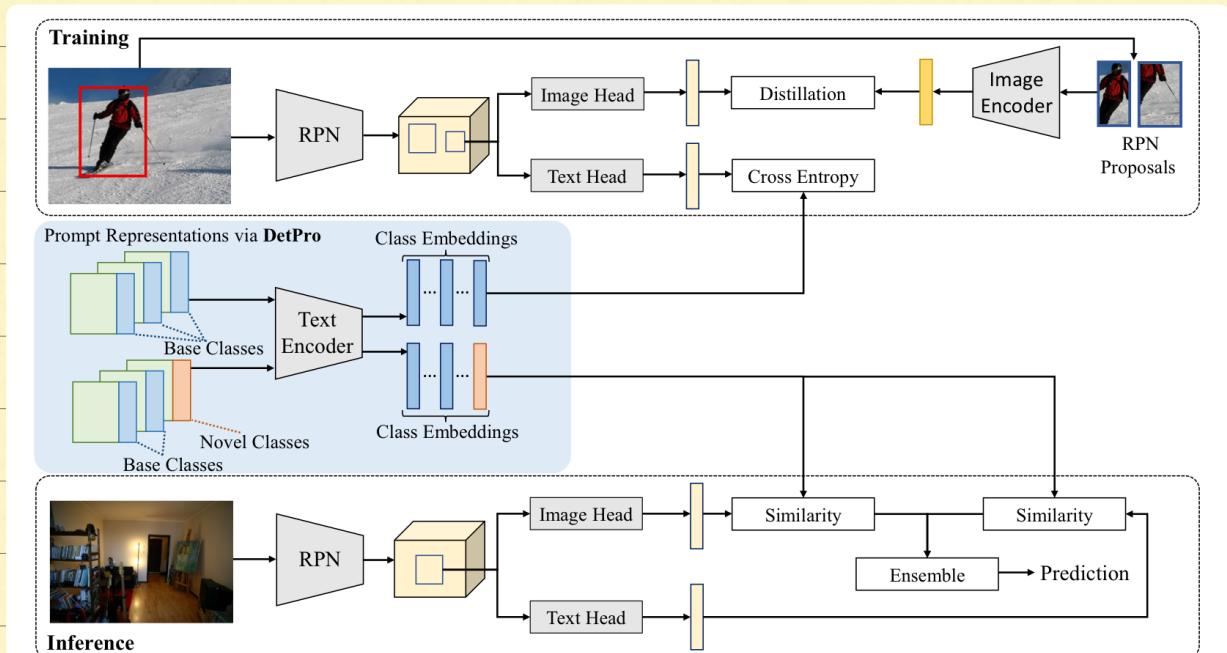


Figure 2. Assembling DetPro with ViLD. DetPro is highlighted with azure background. We omit the class-agnostic bounding box regression branch and mask prediction branch in both training and testing pipelines.

- Training ViLD with DetPro

$V_c = [v_1, v_2 \dots v_L, w_c]$, 은 prompt 생성, T(,)와 feeding하여 embedding 생성

Embedding은 proposal classifier로 사용됨 (detector)

ViLD의 Image RCNN head를 통하여 사용

Image RCNN head는 CLIP의 encoder를 distillation하고,

Text RCNN은 기본 class embedding, background embedding을 사용

RPN을 통해 생성된 ROI로 손실 계산을 하는 feature 추출

L_{txt} 의 경우 ROI feature와 text embedding 사이의 cosine 유사도 사용

L_{img} 의 경우 image embedding과 ROI feature 사이의 차이 (L1)

- Inference ViLD with DetPro

Inference 시 $V_{bg} = [v_1^{bg}, v_2^{bg}, \dots, v_L^{bg}]$. base+ novel class의 prompt 추출

Text image head에서 ROI feature 추출.

↳ cosine 유사도 계산 후 confidence score 획득

* Experiments

Method	Epoch	Detection				Instance segmentation			
		AP _r	AP _c	AP _f	AP	AP _r	AP _c	AP _f	AP
Supervised (base)	20	0.0	26.1	34.0	24.7	0.0	24.7	29.8	22.4
Supervised (base+novel)	20	15.5	25.5	33.6	27.0	16.4	24.6	30.6	25.5
ViLD (base) [7]	460	16.7	26.5	34.2	27.8	16.6	24.6	30.3	25.5
ViLD* (base) [7]	20	17.4	27.5	31.9	27.5	16.8	25.6	28.5	25.2
DetPro (base)	20	20.8	27.8	32.4	28.4	19.8	25.6	28.9	25.9

Table 1. Comparison with ViLD on LVIS v1 dataset. * denotes our re-implementation version, see Section 5.2 for the details. The frequent and common classes are used as the base classes, while the rare classes are held out as the novel classes. AP_r is the main evaluation metric for open-world object detection.

Method	Pascal VOC		COCO						Objects365					
	AP ₅₀	AP ₇₅	AP	AP ₅₀	AP ₇₅	AP _s	AP _m	AP _l	AP	AP ₅₀	AP ₇₅	AP _s	AP _m	AP _l
Supervised	78.5	49.0	46.5	67.6	50.9	27.1	67.6	77.7	25.6	38.6	28.0	16.0	28.1	36.7
ViLD* [7]	73.9	57.9	34.1	52.3	36.5	21.6	38.9	46.1	11.5	17.8	12.3	4.2	11.1	17.8
DetPro	74.6	57.9	34.9	53.8	37.4	22.5	39.6	46.3	12.1	18.8	12.9	4.5	11.5	18.6

Table 2. We evaluate the LVIS-trained model on Pascal VOC test set, COCO validation set and Object365 validation set.

Strategy	AP _r	AP _c	AP _f	AP
DetPro w/o BG	16.9	25.1	27.7	24.7
DetPro-LearnableBG	15.3	25.4	27.9	24.6
DetPro-SoftBG	19.1	25.4	28.2	25.4

Table 3. Ablation study on different strategies of involving negative proposals into our DetPro.

Background proposals	AP _r	AP _c	AP _f	AP
10%	19.1	25.4	28.2	25.4
30%	18.3	25.6	28.4	25.4
50%	17.8	25.6	28.4	25.4
100%	17.6	25.1	28.2	25.0

Table 4. Ablation on number of background proposals involved in DetPro training.

GT	FG	BG	AP _r	AP _c	AP _f	AP
✓			15.3	25.4	27.9	24.6
✓	✓		16.9	25.1	27.7	24.7
✓		✓	17.7	25.3	28.2	25.1
✓	✓	✓	19.1	25.4	28.2	25.4

Table 5. Ablation study on the involvement of different training data. ‘GT’: ground-truth; ‘FG’: foreground; ‘BG’: background.

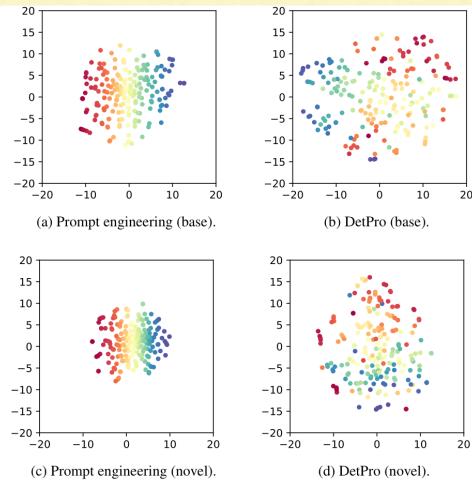


Figure 3. We randomly select 200 base classes and 200 novel classes from LVIS dataset and use t-SNE to visualize the class embedding generated by our DetPro and the classical prompt engineering. (a) base class embedding generated by prompt engineering; (b) base class embedding generated by DetPro; (c) novel class embedding generated by prompt engineering; (d) novel class embedding generated by DetPro. Each point denotes a category. Class embedding generated by our method is more discriminative in the embedding space, which attributes to the involvement of background proposals during training.

IoU range	AP _r	AP _c	AP _f	AP
0.5-0.6	17.3	25.3	28.2	25.0
0.6-0.7	18.0	25.4	28.1	25.4
0.7-0.8	17.2	25.4	28.3	25.1
0.8-0.9	17.3	24.9	28.2	24.9
0.9-1.0	17.2	25.2	28.3	25.0
0.5-1.0	16.1	25.7	28.3	25.1
0.6-1.0	17.2	25.4	28.9	25.3
0.7-1.0	16.8	25.0	28.3	25.1
0.8-1.0	17.2	25.2	28.4	25.1
Ensemble (0.5:1.0:0.1)	19.1	25.4	28.2	25.4
Ensemble (0.6:1.0:0.1)	18.4	25.2	28.2	25.2
Ensemble (0.7:1.0:0.1)	18.7	25.8	28.3	25.5
Ensemble (0.8:1.0:0.1)	18.2	25.3	28.1	25.2

Table 6. The effects of prompt representation ensemble. ‘Ensemble (0.5:1.0:0.1)’ represents we divide the positive proposals of the IoU range [0.5-1.0] into 5 disjoint groups with an IoU interval of 0.1. Then we use each group to train a separate DetPro and perform ensemble on 5 learned models.

Length	AP _r	AP _c	AP _f	AP
4	18.7	24.9	28.2	25.1
8	19.1	25.6	28.3	25.2
16	17.7	25.6	28.3	25.3

Table 7. Ablation study on context lengths.

Position	AP _r	AP _c	AP _f	AP
Front	16.4	24.5	28.3	24.6
Middle	18.0	25.1	28.3	25.1
End	19.1	25.4	28.2	25.4

Table 8. Ablation study of inserting class token into different positions of prompt representation.