

Learning to prompt for vision-language models

Zhou, K., Yang, J., Loy, C. C., & Liu, Z. (2022). Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9), 2337-2348.

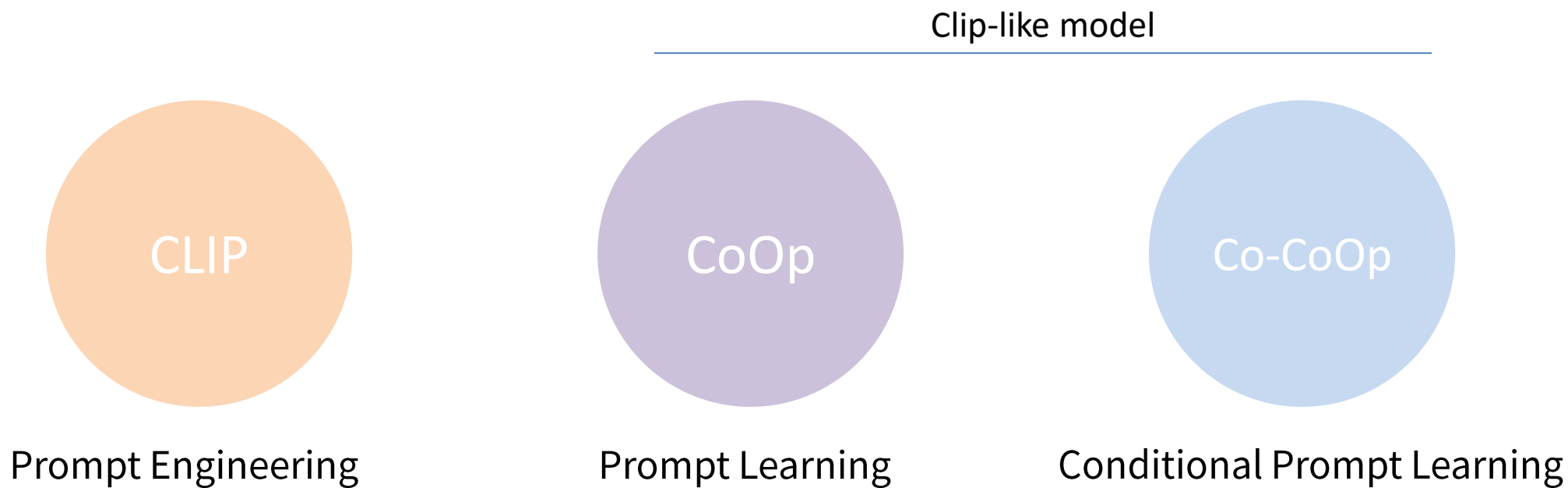
Conditional prompt learning for vision-language models

Zhou, K., Yang, J., Loy, C. C., & Liu, Z. (2022). Conditional prompt learning for vision-language models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 16816-16825).

Test-Time Prompt Tuning for Zero-Shot Generalization in Vision-Language Models

Shu, M., Nie, W., Huang, D. A., Yu, Z., Goldstein, T., Anandkumar, A., & Xiao, C. (2022). Test-time prompt tuning for zero-shot generalization in vision-language models. arXiv preprint arXiv:2209.07511.

Prompt Learning



Zero-Shot Transfer

- Transfer Learning : 특정 태스크를 학습한 모델을 다른 태스크 수행에 재사용하는 기법
- Transfer Learning 한계점
 - Fine-tuning 없이 새로운 downstream task에 적용하기 어려운 일반화 문제가 발생
 - 새로운 태스크에 적합한 다량의 이미지와 라벨링 작업을 요구
 - 벤치마크 데이터셋 성능과 실제 데이터셋과 도메인의 차이로 인해 성능 차이 존재 가능성
- ▶ pre-training을 할 때 Fine-tuning이 필요 없는 일반화 된 모델 또는 이미지 수집 및 정답 레이블 생성에 적은 노력을 기울이며, 여러 현실 데이터셋에도 좋은 성능을 보이는 강건한 모델 필요

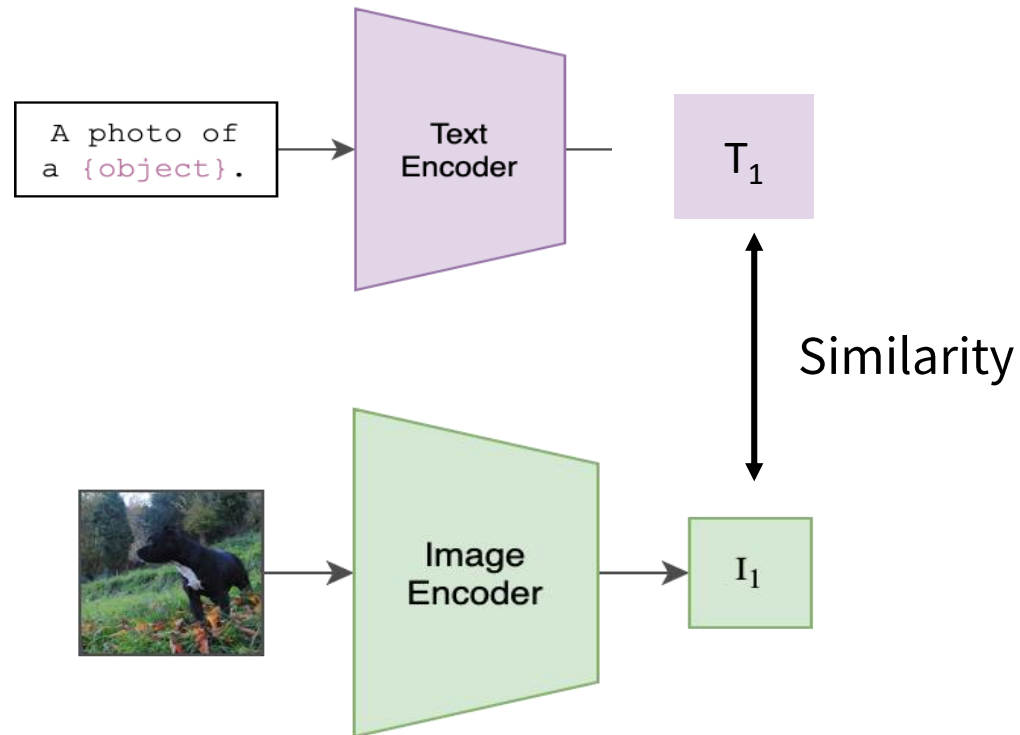
Prompt learning in Vision-Language Pretrained model

- 백본 아키텍처로 VLM은 visual과 textual data를 input으로 사용, LM은 텍스트 데이터만 사용
- 목표 함수로 clip-like model은 contrastive learning을 통해 학습하고, LM은 autoregressive learning

	Clip-like model(coop, co-coop)	Language Model
Backbone architecture	Visual and textual data as input	Text data
Pre-training objective	Contrastive Learning	Autoregressive Learning

Prompt learning in Vision-Language Pretrained model

- Vision-language models은 이미지와 텍스트 모델들이 각각 데이터로 사전 학습
- 각 task에 맞는 프롬프트 텍스트를 직접 사용자가 지정하여 zero-shot 성능을 향상



Prompt engineering in NLP

- Prompt Engineering : 자연어 처리 모델의 성능을 향상시키기 위해 사용자 입력에 대한 적절한 응답을 생성하는 데 필요한 **최적의 프롬프트를 찾아내는** 기술
- Prompt Learning : 모델 **학습 과정**에 있어서 적절한 프롬프트를 사용하여 더 나은 결과와 이해력을 얻기 위한 자연어 처리 학습 방법론

Example

Prompt Function : [X] Overall, it was a [Z] movie.




Input(X) : “가오갤3”, “드림”, “슈퍼마리오”

Answer(Z) : “Good”, “Fantastic”, “Boring”

Template : Input [X] 뒤에 [Z]를 가진 문장이 덧붙여진 하나의 구조

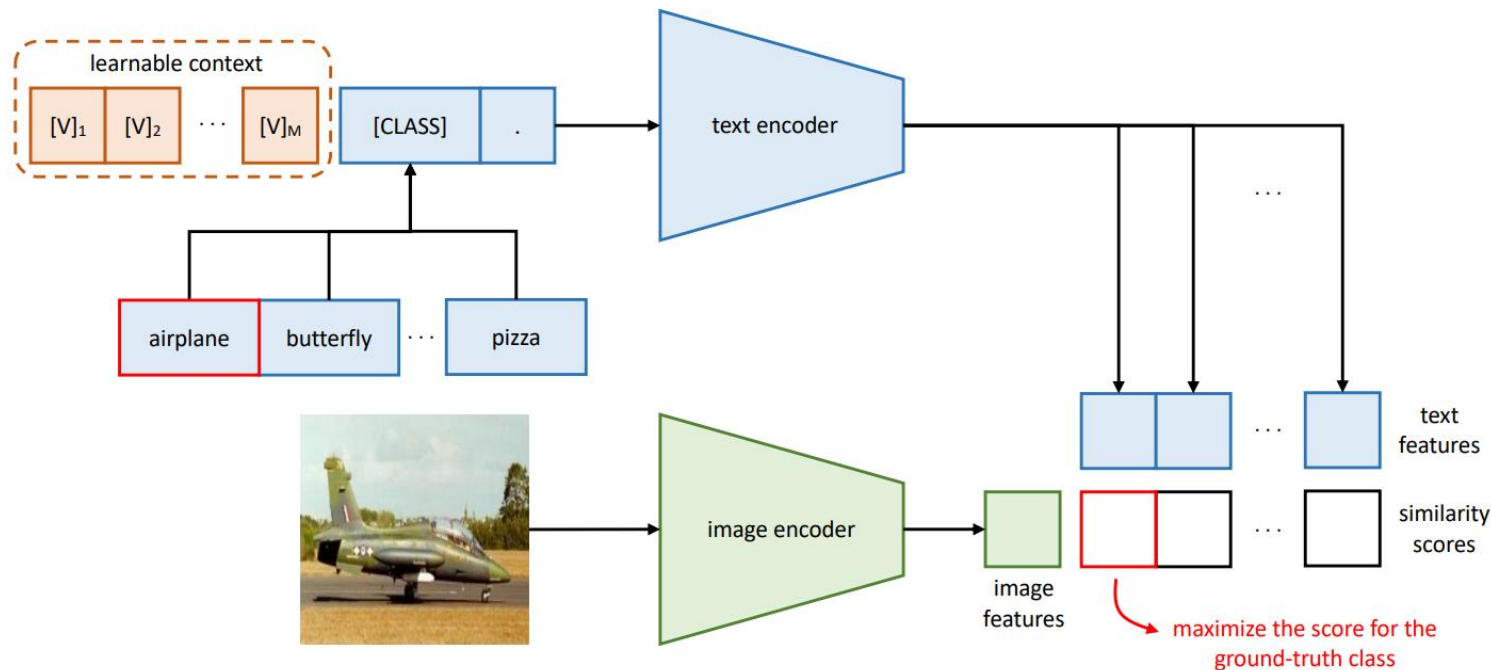
CoOp(21 CVPR)

- 적절한 프롬프트를 찾는 것은 non-trivial task라 볼 수 있음(많은 전문 지식 요구 & 단어 튜닝에 많은 시간 요구)
Ex) {class}가 오기 전에 “a”를 더하는 것과 더하지 않는 것을 비교했을 때 5%의 차이가 발생
- 광범위한 조정이 있더라도 결과적으로 프롬프트가 다운스트림 태스크 작업에 최적을 보장해주지는 않음

Caltech101	Prompt	Accuracy	Describable Textures (DTD)	Prompt	Accuracy	EuroSAT	Prompt	Accuracy
	a [CLASS].	82.68		a photo of a [CLASS].	39.83		a photo of a [CLASS].	24.17
	a photo of [CLASS].	80.81		a photo of a [CLASS] texture.	40.25		a satellite photo of [CLASS].	37.46
	a photo of a [CLASS].	86.29		[CLASS] texture.	42.32		a centered satellite photo of [CLASS].	37.56
	[V] ₁ [V] ₂ ... [V] _M [CLASS].	91.83		[V] ₁ [V] ₂ ... [V] _M [CLASS].	63.58		[V] ₁ [V] ₂ ... [V] _M [CLASS].	83.53

CoOp(21 CVPR)

- Prompt learning을 pre-train된 CLIP에 도입
- CoOp: “적절한 prompt를 찾는 과정을 자동화” → 데이터에서 end-to-end로 학습할 수 있는 continuous vector(랜덤값 혹은 pretrained된 워드임베딩으로 초기화 시킨 learnable vector)를 사용하여 각 context token을 모델링 하는 것



Method of CoOp

Unified Context

- 모든 class에 동일한 context 공유

$$t = [V]_1[V]_2 \dots [V]_M[\text{CLASS}]$$

$$t = [V]_1 \dots [V]_{\frac{M}{2}}[\text{CLASS}][V]_{\frac{M}{2}+1} \dots [V]_M$$

Class token의 위치
prompt의 중간 또는 끝에 위치

Class-specific Context

- 각 class에 대해 특정 set의 context vector 학습

$$[V]_1^i[V]_2^i \dots [V]_M^i \neq [V]_1^j[V]_2^j \dots [V]_M^j : \\ \text{for } i \neq j \text{ and } i, j \in \{1, \dots, K\}$$

- $[V]_M$: 워드 임베딩과 같은 차원의 벡터 / M : context token의 수를 지정하는 하이퍼 파라미터
- Unified Context : 모든 클래스와 동일한 컨텍스트를 공유 & 대부분의 범주에서 잘 작동하는 context를 기반
- Class-Specific Context : 각 클래스에 대한 특정 컨텍스트 토큰 집합을 학습하고 일부 세분화된 범주에 더 적합한 방법인 클래스 별 컨텍스트를 기반
- 학습 loss : Cross Entropy

예측 확률

$$p(y = i | x) = \frac{\exp(\cos(g(t_i), f) / \tau)}{\sum_{j=1}^K \exp(\cos(g(t_j), f) / \tau)}$$

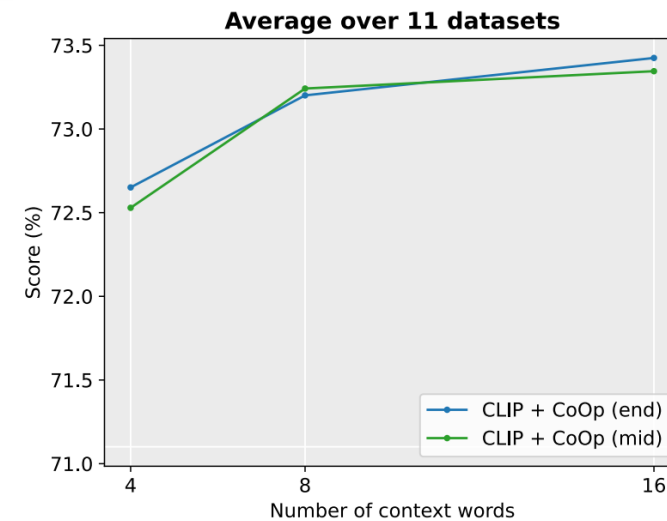
CoOp Experiment – CLIP Prompt와의 비교

■ 컨텍스트 토큰의 길이의 실험 결과

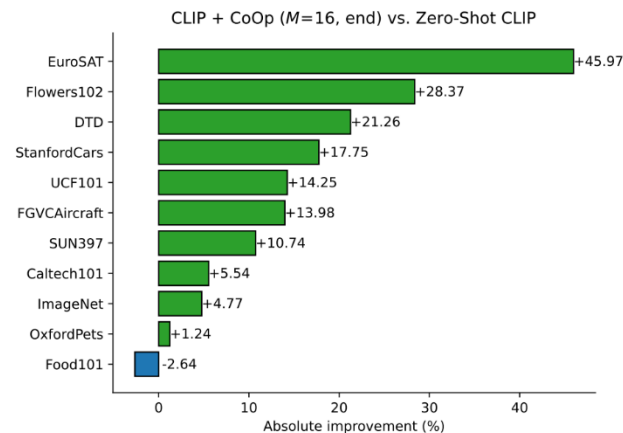
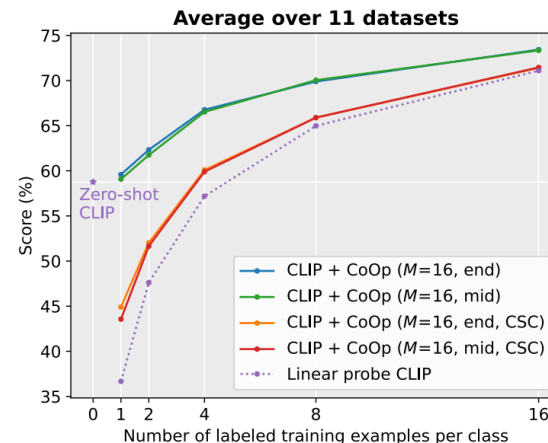
- Domain shift 관점에서는 컨텍스트 길이가 짧을수록 더 도움이 되었음
- 더 작은 매개변수가 학습됨에 따라 과적합이 줄어들기 때문이라 예상
- Context token이 많을수록 성능이 향상되고, 프롬프트 중간에 있는 클래스 토큰은 더 긴 텍스트 길이를 필요

■ CLIP vs CoOp

- 베스트끼리 비교 했을때 성능 향상이 존재
- zero-shot CLIP에 대해 여러 dataset의 fine-tuning(Linear probing) 과정에서 4-shot 이전까지는 few-shot 성능이 zero-shot 성능 이상으로 보장되지 않았던 것이 CLIP의 limitation이었지만, prompt learning을 통해 few-shot 성능이 열추 zero-shot 성능 이상으로 올라가는 경향성을 확인할 수 있음



(a) Context length



CoOp Experiment – 결론

- Fine-Grained
 - Oxford-Pets, StanfordCars, Flowers102, Food101과 같은 Fine-Grained용 데이터셋을 사용해 실험한 결과 ‘Class-specific context’가 좋은 성능을 보임
- Few-shot Classification
 - Zero-shot CLIP과 비교했을 때 16shot 기준 적게는 1.24%, 많게는 45.97%의 성능 향상을 보임
- Domain Generalization
 - CoOp을 통한 학습은 Domain Shifting 상황에 대해서 보다 Robustness를 가짐

Method	Source	Target			
	ImageNet	-V2	-Sketch	-A	-R
ResNet-50					
Zero-Shot CLIP	58.18	51.34	33.32	21.65	56.00
Linear Probe CLIP	55.87	45.97	19.07	12.74	34.86
CLIP + CoOp ($M=16$)	62.95	55.11	32.74	22.12	54.96
CLIP + CoOp ($M=4$)	63.33	55.40	34.67	23.06	56.60
ResNet-101					
Zero-Shot CLIP	61.62	54.81	38.71	28.05	64.38
Linear Probe CLIP	59.75	50.05	26.80	19.44	47.19
CLIP + CoOp ($M=16$)	66.60	58.66	39.08	28.89	63.00
CLIP + CoOp ($M=4$)	65.98	58.60	40.40	29.60	64.98
ViT-B/32					
Zero-Shot CLIP	62.05	54.79	40.82	29.57	65.99
Linear Probe CLIP	59.58	49.73	28.06	19.67	47.20
CLIP + CoOp ($M=16$)	66.85	58.08	40.44	30.62	64.45
CLIP + CoOp ($M=4$)	66.34	58.24	41.48	31.34	65.78
ViT-B/16					
Zero-Shot CLIP	66.73	60.83	46.15	47.77	73.96
Linear Probe CLIP	65.85	56.26	34.77	35.68	58.43
CLIP + CoOp ($M=16$)	71.92	64.18	46.71	48.41	74.32
CLIP + CoOp ($M=4$)	71.73	64.56	47.89	49.93	75.14

CoOp Contribution & Limitation

■ Contribution

- ① 기존의 VLP의 manual한 비효율적인 문제점을 지적하며 downstream application을 연구
- ② Pre-trained VL model을 통해 Prompt engineering 자동화
- ③ 기존 CLIP의 prompt와 linear probe model 보다 더 좋은 성능을 보임
- ④ Domain shift에 있어서 더 robustness함

■ Limitation

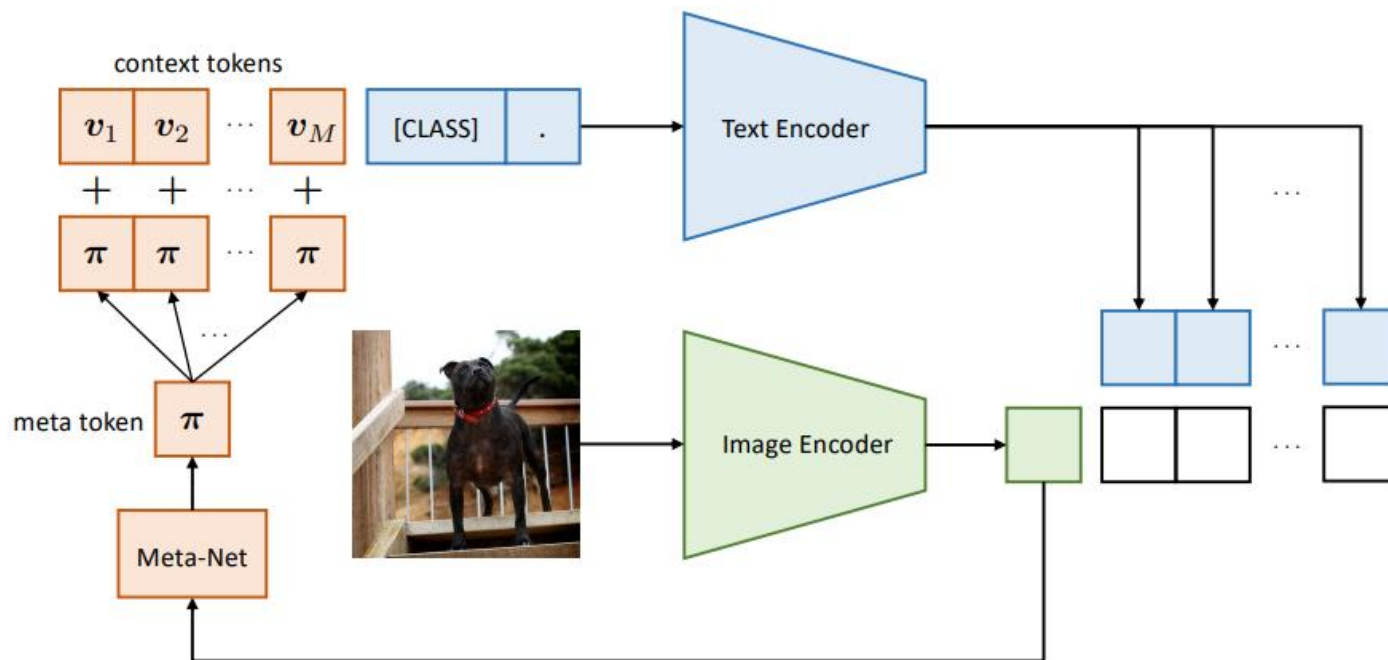
- 학습 과정에서 context가 downstream task에 overfitting이 되다 보니, in-domain class에 대해서는 좋은 성능을 보이지만 비슷한 distribution을 가지는 out-of domain class에 대해서는 낮은 성능을 보임

Ex) 'Wind farm' 이나 'Train railway' 와 같이 비슷한 distribution(scene understating이라는 관점에서) CLIP prompt를 사용하는 zero-shot baseline에 비해 오히려 성능이 나빠짐

<p>New classes</p>  <p>Wind farm Train railway</p>	<p>Zero-shot</p> <p>[a] [photo] [of] [a] [wind farm].</p> <p>⋮</p> <p>[a] [photo] [of] [a] [train railway].</p> <p>Accuracy: 75.35 😊</p>	<p>CoOp</p> <p>[v₁] [v₂] ... [v_M] [wind farm].</p> <p>⋮</p> <p>[v₁] [v₂] ... [v_M] [train railway].</p> <p>Accuracy: 65.89 😞</p>
<p>Base classes</p>  <p>Arrival gate Cathedral</p>	<p>Zero-shot</p> <p>[a] [photo] [of] [a] [arrival gate].</p> <p>⋮</p> <p>[a] [photo] [of] [a] [cathedral].</p> <p>Accuracy: 69.36 😞</p>	<p>CoOp</p> <p>[v₁] [v₂] ... [v_M] [arrival gate].</p> <p>⋮</p> <p>[v₁] [v₂] ... [v_M] [cathedral].</p> <p>Accuracy: 80.60 😊</p>

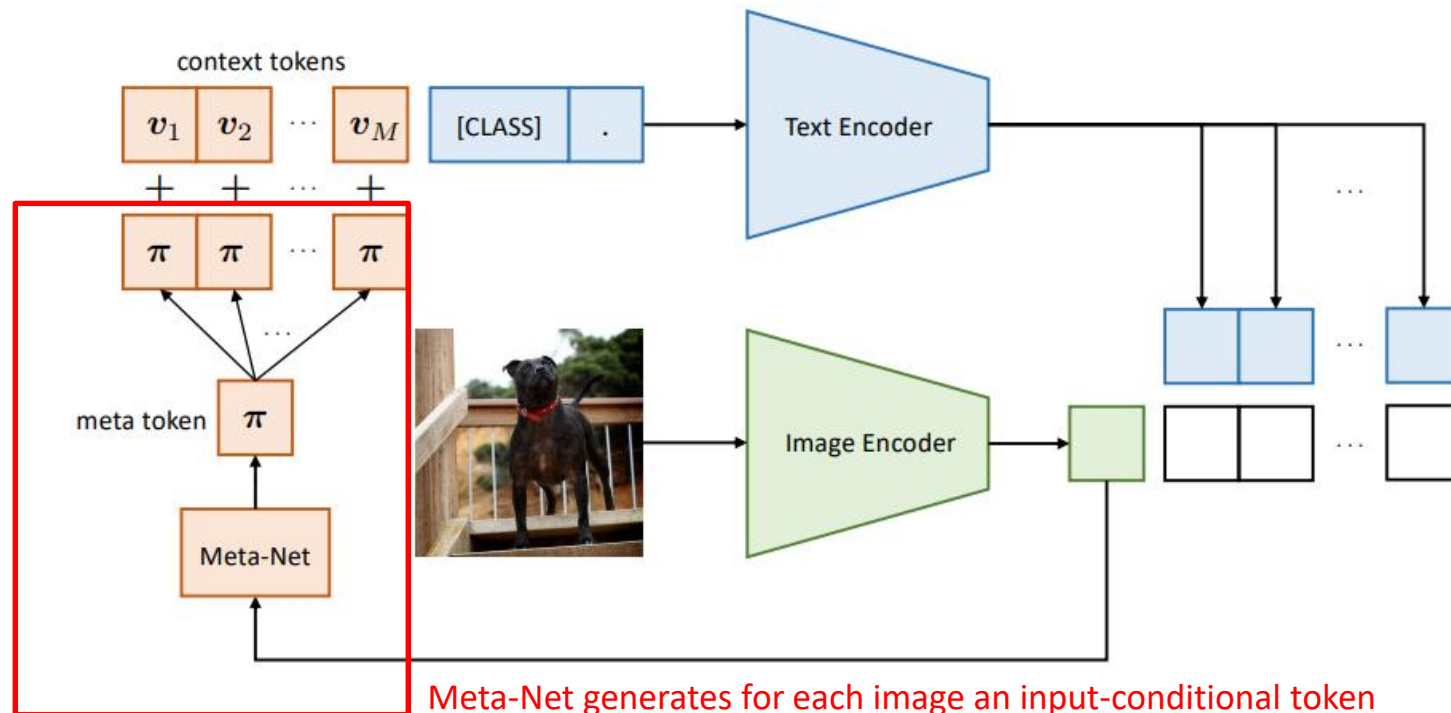
Co-CoOp(22 CVPR)

- 프롬프트가 클래스에 과적합되는 약한 일반화 문제를 해결하고자 조건부 프롬프트 학습 개념을 도입
- 핵심 아이디어는 일단 학습 후 고정되지 않고, 각 입력 인스턴스에 따라 조건이 지정된 프롬프트를 만드는 것
- 학습 가능한 컨텍스트 벡터와 결합되는 input conditional token을 각 이미지에 대해 생성하는 lightweight neural network(Meta-Net)를 추가로 학습하여 CoOp을 확장



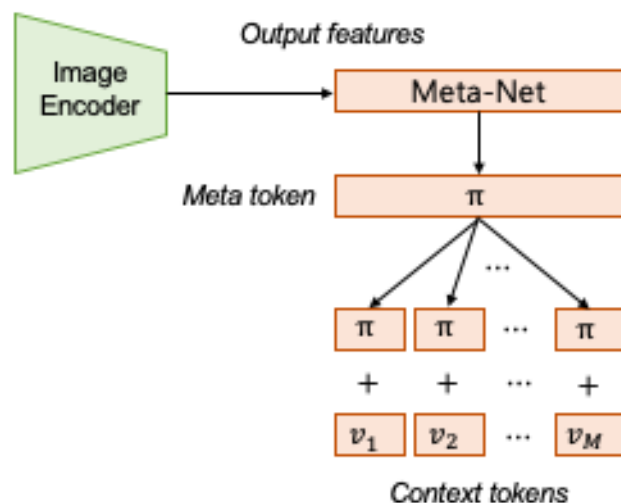
Co-CoOp(22 CVPR)

- Meta-Net : Linear-ReLU-Linear의 two-layer 구조 & 입력 차원을 16배로 축소
- 인풋 : 이미지의 feature vector / 각 이미지에 대해 입력 조건부 토큰을 생성
- Co-CoOp은 context vector & 메타넷 2군데를 학습
- 입력에 대한 conditional token을 생성하는 메타넷은 컨텍스트 벡터와 결합 & 컨텍스트 벡터는 메타넷의 파라미터 세타와 함께 업데이트



Co-CoOp(22 CVPR)

- 그림과 같이 각 입력이 들어갔을 때 나오는 토큰이 각 컨텍스트 토큰과 결합
- 각 클래스에 대한 프롬프트는 결국 입력 인스턴스에 따라 조건이 지정되어, 각 i 번째 클래스에 대한 프롬프트는 $t_i(x)$
- 메타 토큰을 통해 기존 static한 디자인에서 조금 더ダイナ믹하게 변경되어 특정 클래스에 집중하는 것을 방지



$h_\theta(\cdot)$: Meta-Net parameterized by θ
(Linear - ReLU - Linear)

$$\pi = h_\theta(x)$$

$$V_m(x) = V_m + \pi, \quad m \in \{1, 2, \dots, M\}$$

$$t_i(x) = \{v_1(x), v_2(x), \dots, v_M(x), c_i\}$$

c_i : word embedding for the class name

$$p(y|x) = \frac{\exp(\text{sim}(x, g(t_y(x))))/\tau}{\sum_{j=1}^k \exp(\exp(\text{sim}(x, g(t_i(x))))/\tau)}$$

CLIP, CoOp and Co-CoOp

CLIP

$$p(y|x) = \frac{\exp(\text{sim}(x, \mathbf{w}_y)/\tau)}{\sum_{j=1}^k \exp(\exp(\text{sim}(x, \mathbf{w}_j))/\tau)}$$

CoOp

$$p(y|x) = \frac{\exp(\text{sim}(x, \mathbf{g}(\mathbf{t}_y))/\tau)}{\sum_{j=1}^k \exp(\exp(\text{sim}(x, \mathbf{g}(\mathbf{t}_j))/\tau)}$$

Co-CoOp

$$p(y|x) = \frac{\exp(\text{sim}(x, \mathbf{g}(\mathbf{t}_y(x)))/\tau)}{\sum_{j=1}^k \exp(\exp(\text{sim}(x, \mathbf{g}(\mathbf{t}_j(x)))/\tau)}$$

Base classes



Arrival gate



Cathedral

Zero-shot

[a] [photo] [of] [a] [arrival gate].

⋮

[a] [photo] [of] [a] [cathedral].

Accuracy: 69.36 😞

CoOp

[v₁] [v₂] ... [v_M] [arrival gate].

⋮

[v₁] [v₂] ... [v_M] [cathedral].

Accuracy: 80.60 😊

CoCoOp

[v₁(x)] [v₂(x)] ... [v_M(x)] [arrival gate].

⋮

[v₁(x)] [v₂(x)] ... [v_M(x)] [cathedral].

Accuracy: 79.74 😊

New classes



Wind farm



Train railway

Zero-shot

[a] [photo] [of] [a] [wind farm].

⋮

[a] [photo] [of] [a] [train railway].

Accuracy: 75.35 😊

CoOp

[v₁] [v₂] ... [v_M] [wind farm].

⋮

[v₁] [v₂] ... [v_M] [train railway].

Accuracy: 65.89 😞

CoCoOp

[v₁(x)] [v₂(x)] ... [v_M(x)] [wind farm].

⋮

[v₁(x)] [v₂(x)] ... [v_M(x)] [train railway].

Accuracy: 76.86 😊

Co-CoOp Experiment

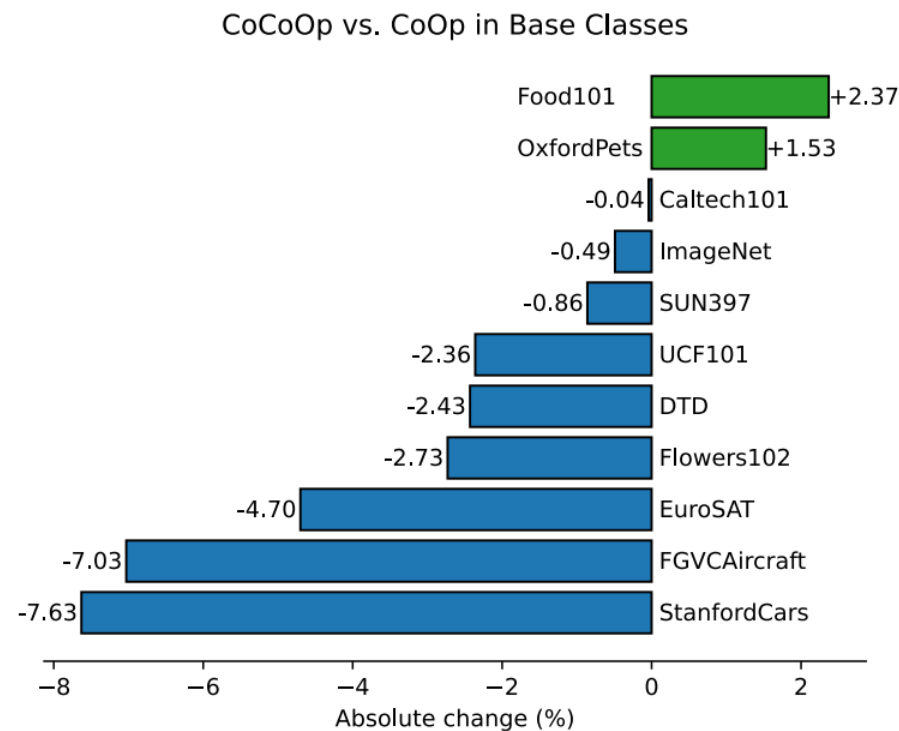
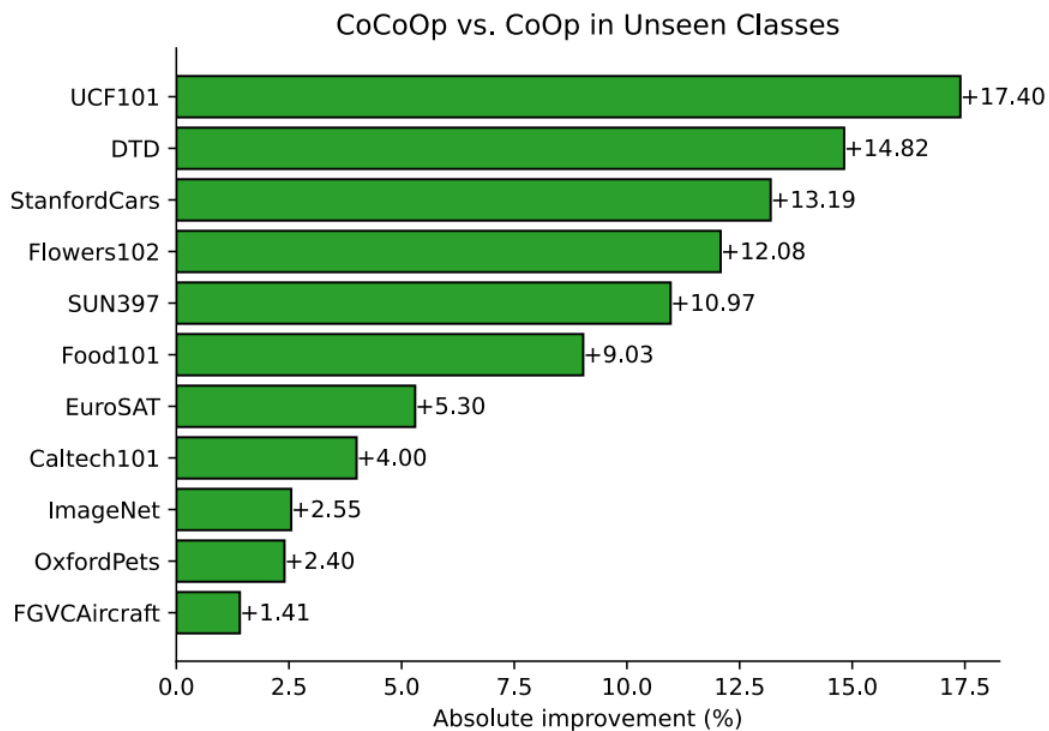
- 11개의 데이터셋에서 클래스를 base class와 new class 두 그룹으로 균등하게 분할
- 학습 기반의 coop과 co-coop은 기본 클래스만 사용하여 학습하고, 기본 클래스와 새 클래스에 대해 별도로 평가를 수행하여 일반화 가능성을 테스트
- new class에 대해서 CLIP이 더 좋은 성능을 보이는 데이터셋이 많이 존재
- CLIP이 unseen class에 대한 zero-shot 성능은 제일 좋았으나, base dataset에 대한 성능이 11% 차이가 난다는 점에서 CoCoOp 방법이 seen class와 unseen class 모두에 적용될 수 있는 방법이라는 것을 주장

(a) Average over 11 datasets.				(b) ImageNet.				(c) Caltech101.			
	Base	New	H		Base	New	H		Base	New	H
CLIP	69.34	74.22	71.70	CLIP	72.43	68.14	70.22	CLIP	96.84	94.00	95.40
CoOp	82.69	63.22	71.66	CoOp	76.47	67.88	71.92	CoOp	98.00	89.81	93.73
CoCoOp	80.47	71.69	75.83	CoCoOp	75.98	70.43	73.10	CoCoOp	97.96	93.81	95.84
(d) OxfordPets.				(e) StanfordCars.				(f) Flowers102.			
	Base	New	H		Base	New	H		Base	New	H
CLIP	91.17	97.26	94.12	CLIP	63.37	74.89	68.65	CLIP	72.08	77.80	74.83
CoOp	93.67	95.29	94.47	CoOp	78.12	60.40	68.13	CoOp	97.60	59.67	74.06
CoCoOp	95.20	97.69	96.43	CoCoOp	70.49	73.59	72.01	CoCoOp	94.87	71.75	81.71
(g) Food101.				(h) FGVCAircraft.				(i) SUN397.			
	Base	New	H		Base	New	H		Base	New	H
CLIP	90.10	91.22	90.66	CLIP	27.19	36.29	31.09	CLIP	69.36	75.35	72.23
CoOp	88.33	82.26	85.19	CoOp	40.44	22.30	28.75	CoOp	80.60	65.89	72.51
CoCoOp	90.70	91.29	90.99	CoCoOp	33.41	23.71	27.74	CoCoOp	79.74	76.86	78.27
(j) DTD.				(k) EuroSAT.				(l) UCF101.			
	Base	New	H		Base	New	H		Base	New	H
CLIP	53.24	59.90	56.37	CLIP	56.48	64.05	60.03	CLIP	70.53	77.50	73.85
CoOp	79.44	41.18	54.24	CoOp	92.19	54.74	68.69	CoOp	84.69	56.05	67.46
CoCoOp	77.01	56.00	64.85	CoCoOp	87.49	60.04	71.21	CoCoOp	82.33	73.45	77.64

	Base	New	H
CLIP	69.34	74.22	71.70
CoOp	82.69	63.22	71.66
CoCoOp	80.47	71.69	75.83

Co-CoOp Experiment

- 왼쪽 : 모든 데이터 세트의 unseen classes에서는 쿵에 비해 일관되게 개선한 모습
- 오른쪽 : 정확도도 3% 미만으로 향상시켰으며, 반대로도 아래 세 개의 데이터셋을 제외하고도 3% 미만의 감소



Co-CoOp Experiment – Domain Generalization

- ImageNet을 소스 데이터로 학습을 하고 다른 10개의 데이터셋 & 4개의 벤치마크 데이터셋에 transfer하여 coop과 co-coop을 DG 성능 비교
- Co-CoOp이 source domain과 target domain의 차이가 커짐에도 CoOp보다 robust한 성능을 보임

	Source	Target										
	ImageNet	Caltech101	OxfordPets	StanfordCars	Flowers102	Food101	FGVCAircraft	SUN397	DTD	EuroSAT	UCF101	Average
CoOp [63]	71.51	93.70	89.14	64.51	68.71	85.30	18.47	64.15	41.92	46.39	66.55	63.88
CoCoOp	71.02	94.43	90.14	65.32	71.88	86.06	22.94	67.36	45.73	45.37	68.21	65.74
Δ	-0.49	+0.73	+1.00	+0.81	+3.17	+0.76	+4.47	+3.21	+3.81	-1.02	+1.66	+1.86

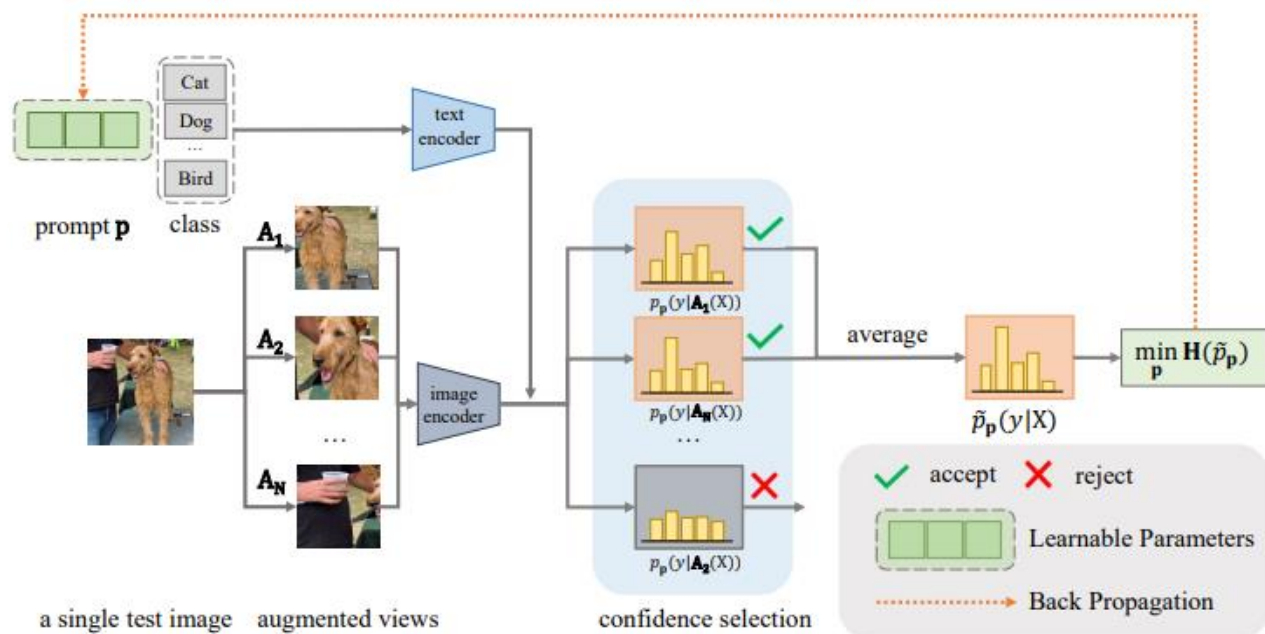
		Source	Target			
	Learnable?	ImageNet	ImageNetV2	ImageNet-Sketch	ImageNet-A	ImageNet-R
CLIP [40]		66.73	60.83	46.15	47.77	73.96
CoOp [63]	✓	71.51	64.20	47.99	49.71	75.21
CoCoOp	✓	71.02	64.07	48.75	50.63	76.18

Co-CoOp Conclusion

- Limitation
 - 학습하는데 GPU가 많이 필요로 함
 - CLIP보다 안좋은 경우도 존재
- Contribution
 - 조건부 prompt를 활용한 일반화 문제 해결
 - 사전 연구 CoOp의 인스턴스에 대한 과적합 문제 해결
 - Domain Shift 문제에 강건
 - CoOp보다 general하게 사용할 수 있음

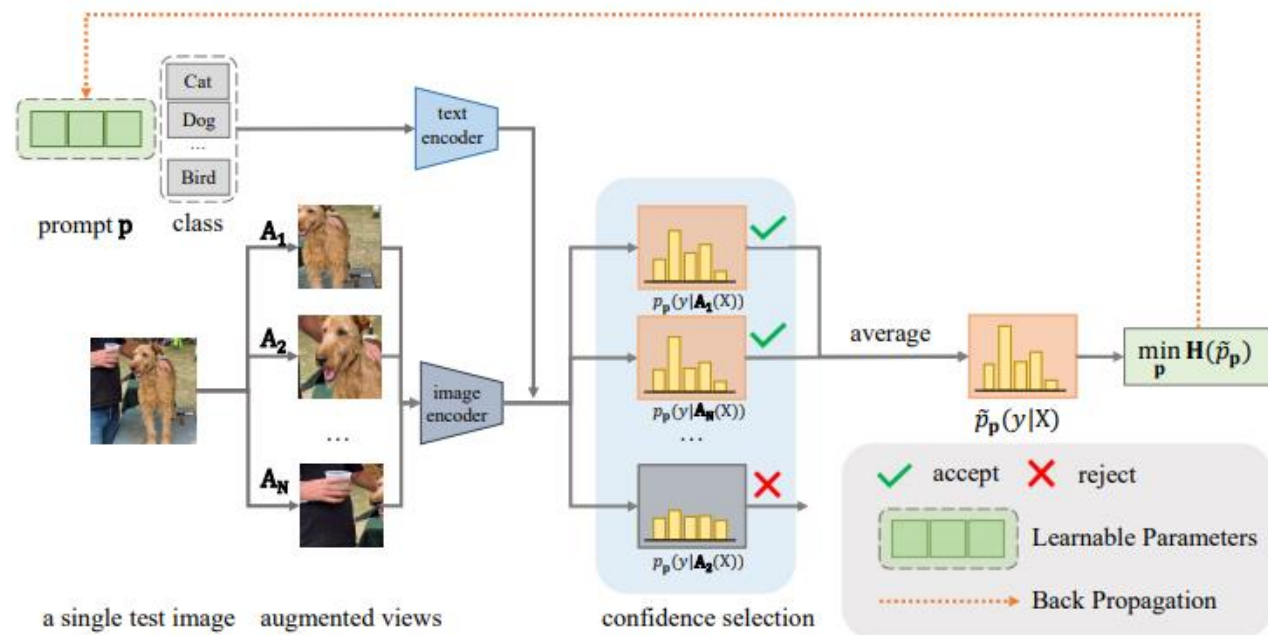
Test-Time Prompt Tuning(22 NuerIPS)

- Test Phase에서 하나의 인스턴스에 대해 학습된 모델을 shift할 수 있는 방법(No additional training data or annotations)
- 1개의 test instance를 여러 방법을 통해 데이터 증강 진행 → 증강한 test image를 CLIP 기반의 모델에 넣고 output(확률 분포) 추출



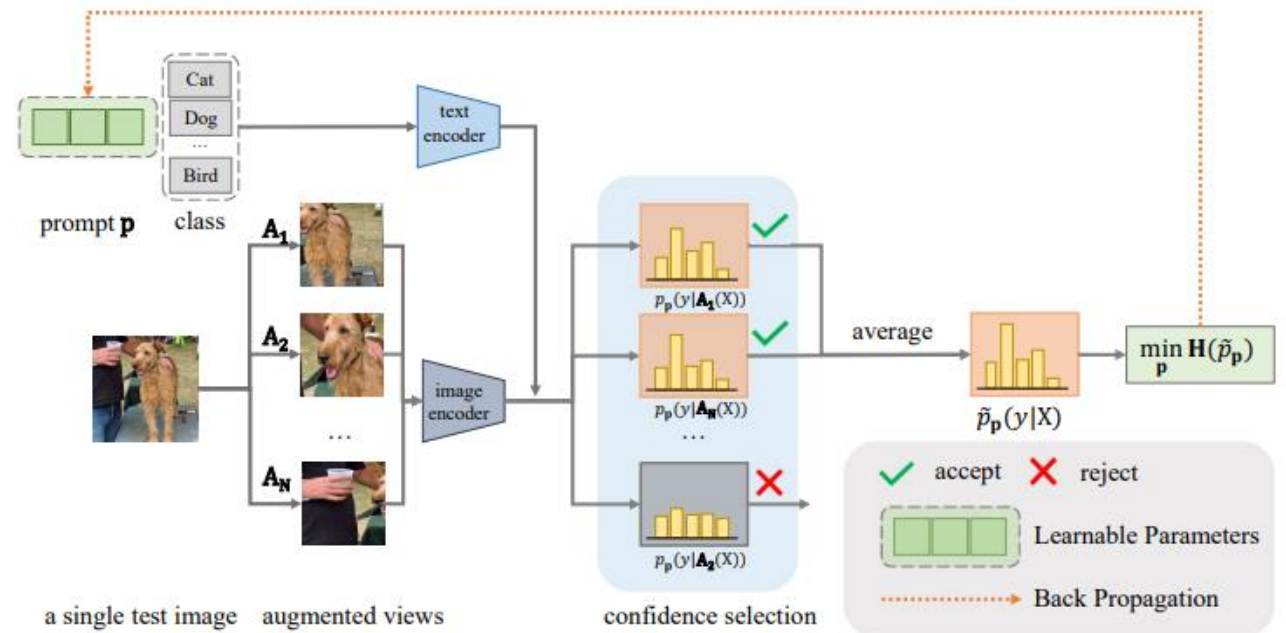
Test-Time Prompt Tuning(22 NuerIPS)

- Confidence selection을 진행하여 무작위로 증강된 image view에서 발생하는 noise 제거
 - Confidence selection은 예측 엔트로피가 임계값 τ 이하인 샘플을 선택하여 제거
 - T는 각 test image에 대해 N개의 증강된 이미지의 self-entropy 중 ρ -percentile의 엔트로피 값으로 설정(신뢰도가 높은 것에서 낮은 것으로 순위가 매겨진 것)
 - self-entropy : 특정 확률 분포의 불확실성이나 정보의 양을 측정하는 척도 / 분포가 더 평평할수록(즉, 가능한 사건들이 대체로 같은 확률을 가질수록) 증가



Test-Time Prompt Tuning(22 NuerIPS)

- Select된 확률들의 평균을 이용해 최종 확률 분포를 구하고 구한 분포를 통해 loss를 계산하고 다시 업데이트
- 조정 후 하나의 test instance에 대해 다시 예측 진행
- CLIP 모델을 freeze 하여 zero-shot 성능을 유지하면서 prompt만 수정해 test instance에 대해 맞춤화 하여 튜닝 가능



Test-Time Prompt Tuning Experiment

- Ensemble : CLIP 반복해 앙상블 / CoOp & Co-CoOp : 4개의 토큰 사용
 - ※Marginal entropy
 - 결합 엔트로피(joint entropy)를 계산할 때 사용되는 개념으로, **단일 확률 변수의 엔트로피**를 나타냄
- Random resize / crop과 같은 단순 augmentation으로 총 64개의 augmentation 진행
- Top 10%($p=0.1$) confidence samples(lowest 10% in self-entropy)만 선택
- 선택한 값의 평균 확률을 marginal entropy 이용하여 최종 class 예측
- 실험 결과 TPT + CoOp이 가장 좋은 강건한 성능을 보임
- CoOp 또는 Co-CoOp으로 학습한 프롬프트에 TPT를 적용함으로써 도메인 내 ImageNet 데이터의 정확도와 OOD 데이터에 대한 일반화 능력이 더욱 향상됨

Method	ImageNet Top1 acc. ↑	ImageNet-A Top1 acc. ↑	ImageNet-V2 Top1 acc. ↑	ImageNet-R Top1 acc. ↑	ImageNet-Sketch Top1 acc. ↑	Average	OOD Average
CLIP-RN50	58.16	21.83	51.41	56.15	33.37	44.18	40.69
Ensemble	59.81	23.24	52.91	60.72	35.48	46.43	43.09
CoOp	63.33	23.06	55.40	56.60	34.67	46.61	42.43
CoCoOp	62.81	23.32	55.72	57.74	34.48	46.81	42.82
TPT	60.74	26.67	54.70	59.11	35.09	47.26	43.89
TPT + CoOp	64.73	30.32	57.83	58.99	35.86	49.55	45.75
TPT + CoCoOp	62.93	27.40	56.60	59.88	35.43	48.45	44.83
CLIP-ViT-B/16	66.73	47.87	60.86	73.98	46.09	59.11	57.2
Ensemble	68.34	49.89	61.88	77.65	48.24	61.20	59.42
CoOp	71.51	49.71	64.20	75.21	47.99	61.72	59.28
CoCoOp	71.02	50.63	64.07	76.18	48.75	62.13	59.91
TPT	68.98	54.77	63.45	77.06	47.94	62.44	60.81
TPT + CoOp	73.61	57.95	66.83	77.27	49.29	64.99	62.83
TPT + CoCoOp	71.07	58.47	64.85	78.65	48.47	64.30	62.61

Test-Time Prompt Tuning Experiment

- Cross-Datasets Generalization에 대해 실험 진행 (이미지넷(소스) / 실험데이터(타겟))
- AugMix를 사용하여 이미지 증강
- 실험 결과 ImageNet을 통해 fine-tuning 되었던 CoOp & Co-CoOp 보다 비슷하거나 좋으며 전반적으로 fine-grained dataset에 대해 좋은 성능을 보이고 있음

Method	Flower102	DTD	Pets	Cars	UCF101	Caltech101	Food101	SUN397	Aircraft	EuroSAT	Average
CLIP-RN50	61.75	40.37	83.57	55.70	58.84	85.88	73.97	58.80	15.66	23.69	55.82
Ensemble	62.77	40.37	82.97	55.89	59.48	87.26	74.82	60.85	16.11	25.79	56.63
CoOp	61.55	37.29	87.00	55.32	59.05	86.53	75.59	58.15	15.12	26.20	56.18
CoCoOp	65.57	38.53	88.39	56.22	57.10	87.38	76.2	59.61	14.61	28.73	57.23
TPT	62.69	40.84	84.49	58.46	60.82	87.02	74.88	61.46	17.58	28.33	57.66
CLIP-ViT-B/16	67.44	44.27	88.25	65.48	65.13	93.35	83.65	62.59	23.67	42.01	63.58
Ensemble	66.99	45.04	86.92	66.11	65.16	93.55	82.86	65.63	23.22	50.42	64.59
CoOp	68.71	41.92	89.14	64.51	66.55	93.70	85.30	64.15	18.47	46.39	63.88
CoCoOp	70.85	45.45	90.46	64.90	68.44	93.79	83.97	66.89	22.29	39.23	64.63
TPT	68.98	47.75	87.79	66.87	68.04	94.16	84.67	65.5	24.78	42.44	65.10

Test-Time Prompt Tuning Conclusion

▪ Limitation

- CoOp과 Co-CoOp의 best token 개수를 사용하지 않음
- 앙상블을 baseline으로 하는 이유가 명확하지 않음
- 두번째 실험에서 평균적으로는 TPT가 높지만 다른 방법이 결과가 더 좋은 경우가 더 많음(CLIP : 2번, CoOp : 1번, Co-CoOp : 9번, TPT+CoCoop : 8번)

▪ Contribution

- Test phase에서의 튜닝을 통해 제로샷 일반화를 유지하면서 도메인 일반화 성능을 개선
- TPT는 특히 자연 데이터 분포 변화에 잘 일반화되며, 다양한 데이터셋 간의 일반화 능력을 측정할 때 전반적으로 높은 성능을 보임
- 모델에 대한 강력한 일반화 능력을 제공하며 이미지 분류와 같은 다양한 비전 작업에서 효과적으로 활용 가능