

TextRefiner: Internal Visual Feature as Efficient Refiner for Vision-Language Models Prompt Tuning

Xie, Jingjing, et al. *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 39. No. 8. **2025**.

발표자 : 홍권

Contents

1. **Introduction**
2. **Related Works**
3. **Method**
4. **Experiment**
5. **Conclusion**

Introduction

- 최근 Vision-Language Model이 발달함에 따라 downstream task에 효율적으로 적응시키기 위한 **PEFT(Parameter-Efficient Fine-Tuning)** 기법들이 다수 등장
- 본 논문에서는 PEFT의 다양한 기법 중 **Prompt Learning**을 개선
- 본 논문은 Prompt Learning 중 외부 지식 없이 **세부적인(Fine-Grained)** 정보를 반영하도록 인코더 내부의 표현을 활용하여 텍스트 임베딩을 Fine-tuning 하는 **TextRefiner**를 제안

Contents

1. **Introduction**
2. Related Works
3. Method
4. Experiment
5. Conclusion

Contents

1. Introduction

2. Related Works

- Vision-Language Model
- Prompt Learning in VLMs

3. Method

4. Experiment

5. Conclusion

Vision-Language Model

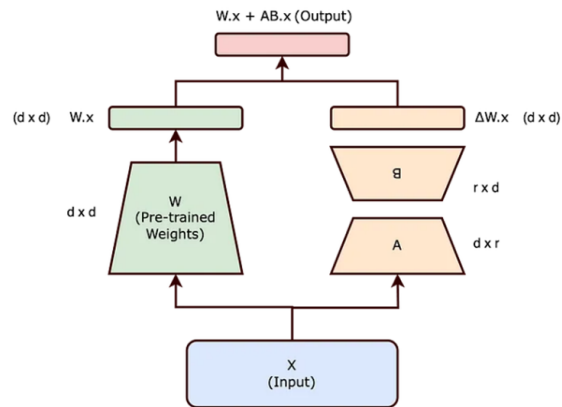
- 전통적인 지도학습은 사전에 정의된 클래스(Label)에 의존 → 보지 못했던 클래스(Label)로의 확장이 제한
- 이런 한계를 해결하기 위해 VLM은 인터넷에서 수집한 방대한 양의 Image-Text Pair를 이용한 **Self-Supervised** 방식으로 학습
- CLIP, ALIGN 등은 **대조손실(Contrastive Loss)**을 활용하여 학습
- 대규모의 데이터셋 + 대조 학습 목표 덕분에 다양한 Downstream Vision task(Object Detection, Semantic Segmentation 등)에 활용
- **Few-Shot** 환경의 downstream task로 전이하는 것은 사용 가능한 데이터의 양이 현저히 적어 여전히 어려움.

Prompt Learning in VLMs

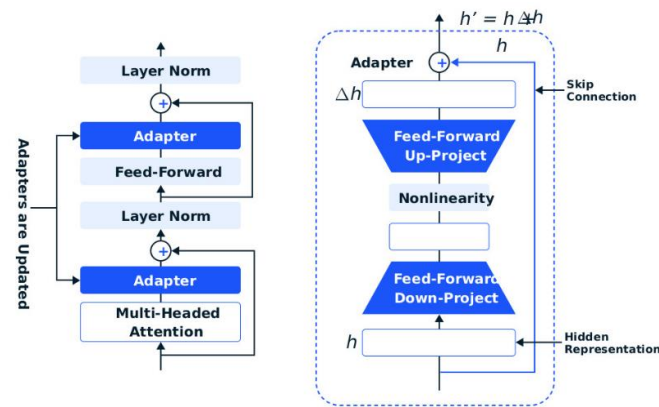
- **Prompt Learning**은 VLM을 downstream task에 적응시키는 데 있어 점점 선호되는 방식
- VLM에서 Prompt Learning은 단순한 **하드 프롬프트**를 넘어서 **학습 가능한 프롬프트**를 도입하는 것
- 기존 사전 학습 모델을 파라미터 효율적으로 Fine-tuning(**PEFT**)하는 방법에는 다양한 방법이 존재
 - LoRA(Low-Rank Adaptation) : $W' = W + BA$,
 - Adapter : $Adapter(x) = W_{down} \cdot ReLU(W_{up} \cdot x)$
 - Prompt Learning : $Input = [P_1, P_2, \dots, P_n, S]$
- **Prompt Learning**은 입력 시퀀스에 **Learnable Parameter**를 도입하는 방식으로 구현이 간단하고 모델 아키텍처에 의존적이지 않음

Prompt Learning in VLMs

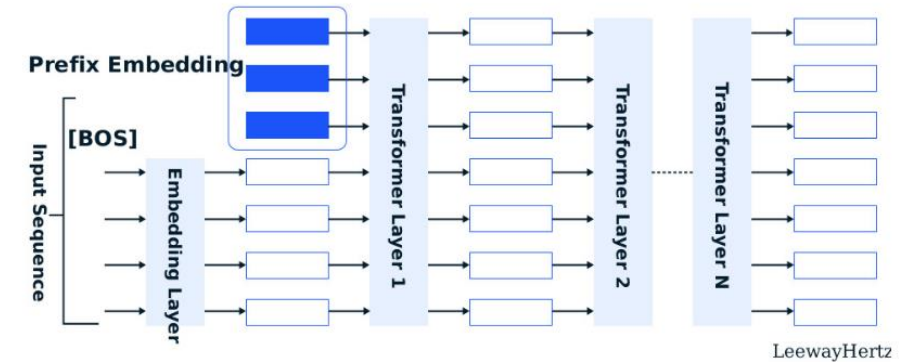
- LoRA(Low-Rank Adaptation)



- Adapter



- Prompt Learning

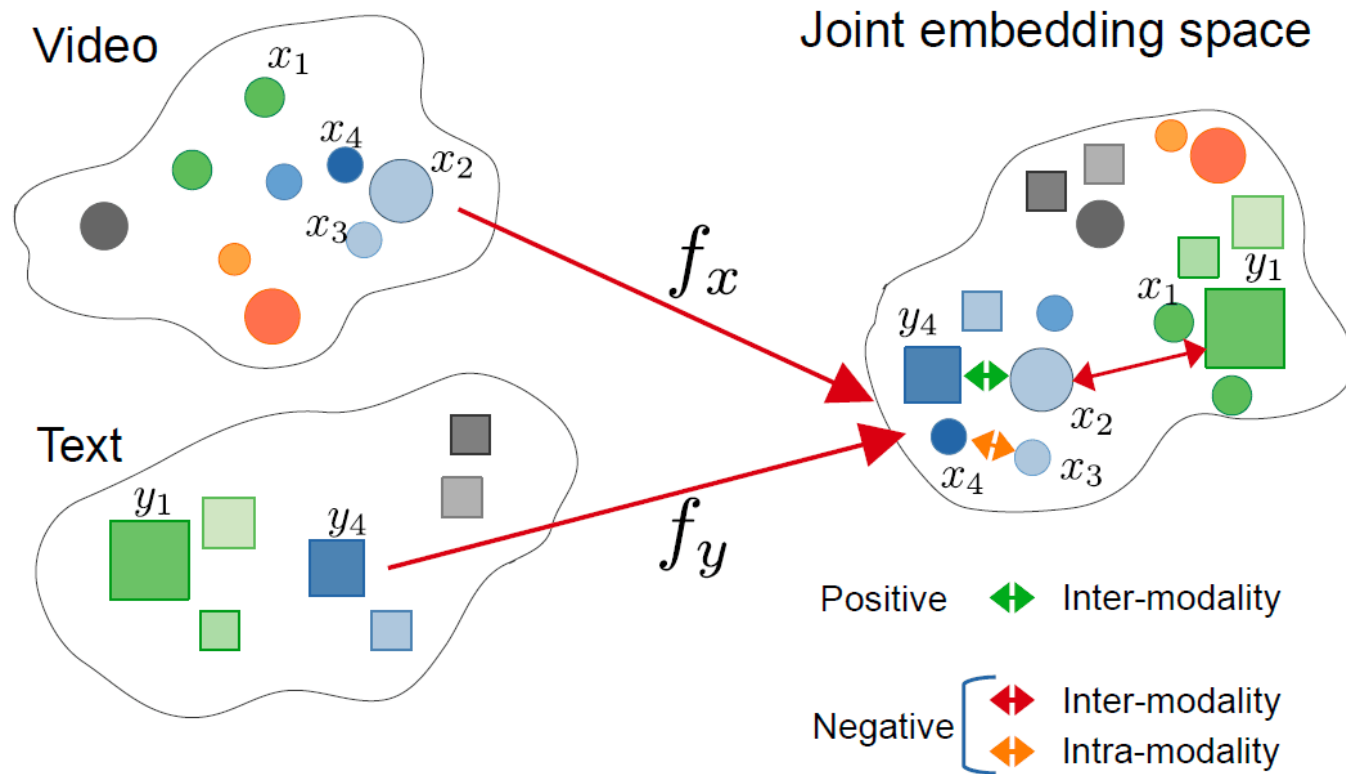


Contents

1. Introduction
2. Related Works
- 3. Method**
 - Background
 - TextRefiner
4. Experiment
5. Conclusion

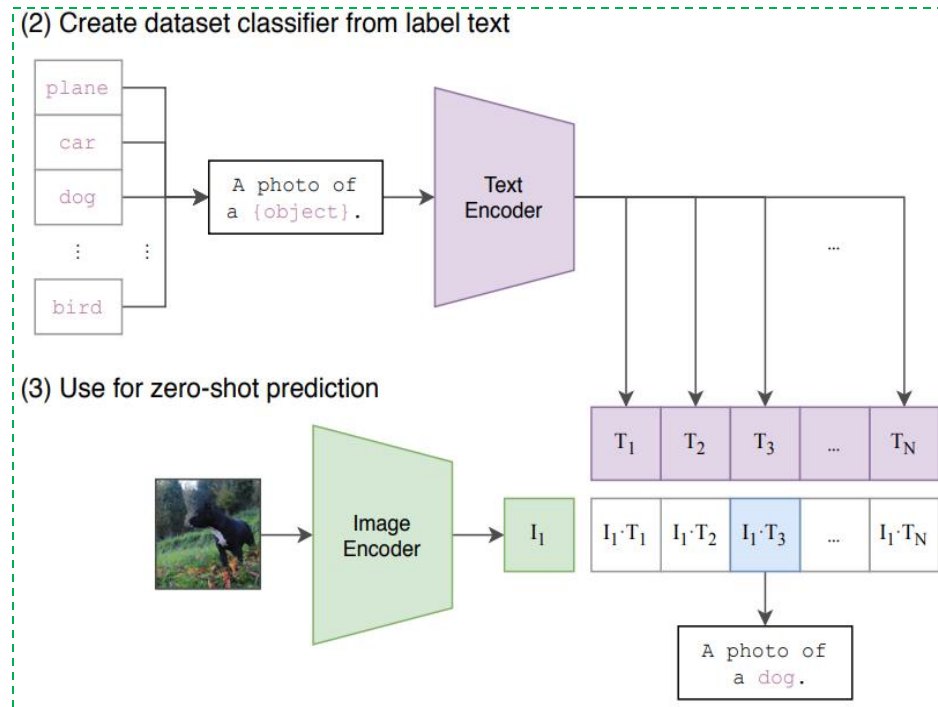
Preliminaries

- **CLIP, ALIGN** [1, 2]과 같은 Vision-Language Model들은 이미지와 텍스트를 **joint embedding space**에 정렬시킴으로써 놀라운 발전을 거듭



Preliminaries

- CLIP



- Image Encoder : f_I
- Text Encoder : f_T
- 대조 손실(Contrastive Loss)이 학습 과정에 사용되며, f_I 와 f_T 를 통과한 임베딩을 사용.

$$P(c) = \frac{\exp(\cos(v, E_c)/\tau)}{\sum_{c=1}^C \exp(\cos(v, E_c)/\tau)}$$


Preliminaries

- **VLMs Prompt Tuning**


CLIP을 통한 추론 시 "A photo of a"와 같은 하드 프롬프트를 사용

Downstream task 적용 시, **최적의 하드 프롬프트**를 찾는 것에 대한 어려움이 존재

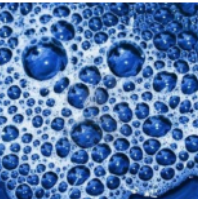
CoOp[\[link\]](#)와 같은 연구들은 하드 프롬프트 대신 **학습 가능한 매개변수**를 도입함으로 학습 과정 중 최적의 프롬프트를 찾도록 함.

Caltech101	Prompt	Accuracy
	a [CLASS].	82.68
	a photo of [CLASS].	80.81
	a photo of a [CLASS].	86.29
	$[V]_1 [V]_2 \dots [V]_M$ [CLASS].	91.83

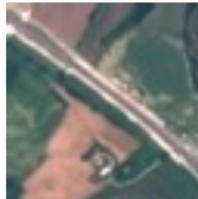
(a)

Flowers102	Prompt	Accuracy
	a photo of a [CLASS].	60.86
	a flower photo of a [CLASS].	65.81
	a photo of a [CLASS], a type of flower.	66.14
	$[V]_1 [V]_2 \dots [V]_M$ [CLASS].	94.51

(b)

Describable Textures (DTD)	Prompt	Accuracy
	a photo of a [CLASS].	39.83
	a photo of a [CLASS] texture.	40.25
	[CLASS] texture.	42.32
	$[V]_1 [V]_2 \dots [V]_M$ [CLASS].	63.58

(c)

EuroSAT	Prompt	Accuracy
	a photo of a [CLASS].	24.17
	a satellite photo of [CLASS].	37.46
	a centered satellite photo of [CLASS].	37.56
	$[V]_1 [V]_2 \dots [V]_M$ [CLASS].	83.53

(d)

Preliminaries

- **VLMs Prompt Tuning**

위와 같은 방식도 여전히 모든 클래스에 적용될 수 있게 **Coarse-grained**하게 임베딩을 강화한다는 단점이 존재.

> **Localized Region(Fine-grained)**, 세밀한 속성 정보나 지역적인 속성을 반영하는 것에 실패.

최근 연구들이 **LLM의 외부 지식을 활용**하여 이러한 문제를 개선

> 그러나 **추가적인 추론 비용**이 발생

따라서 본 논문은 외부 LLM을 이용하지 않고, **아키텍처 개선**을 통해 **Fine-grained**한 속성을 반영할 수 있도록 하겠다!

TextRefiner

- **TextRefiner** : 외부 LLM 없이 VLM 프롬프트 튜닝이 가능한 Plug-and-Play 방식 제안
- **Local Cache, Feature Aggregation, Feature Alignment** 로 구성된 세 가지 요소를 통해 **세밀한 시각 개념(Fine-grained visual concepts)**을 풍부하게 포착할 수 있음

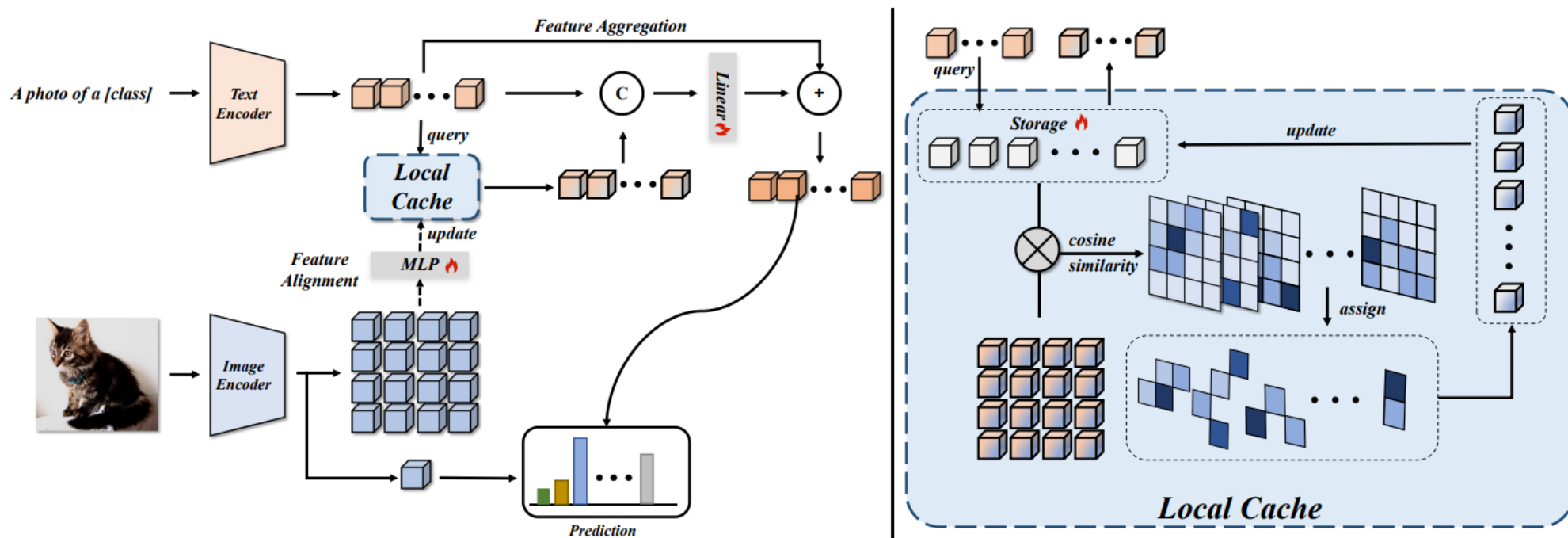
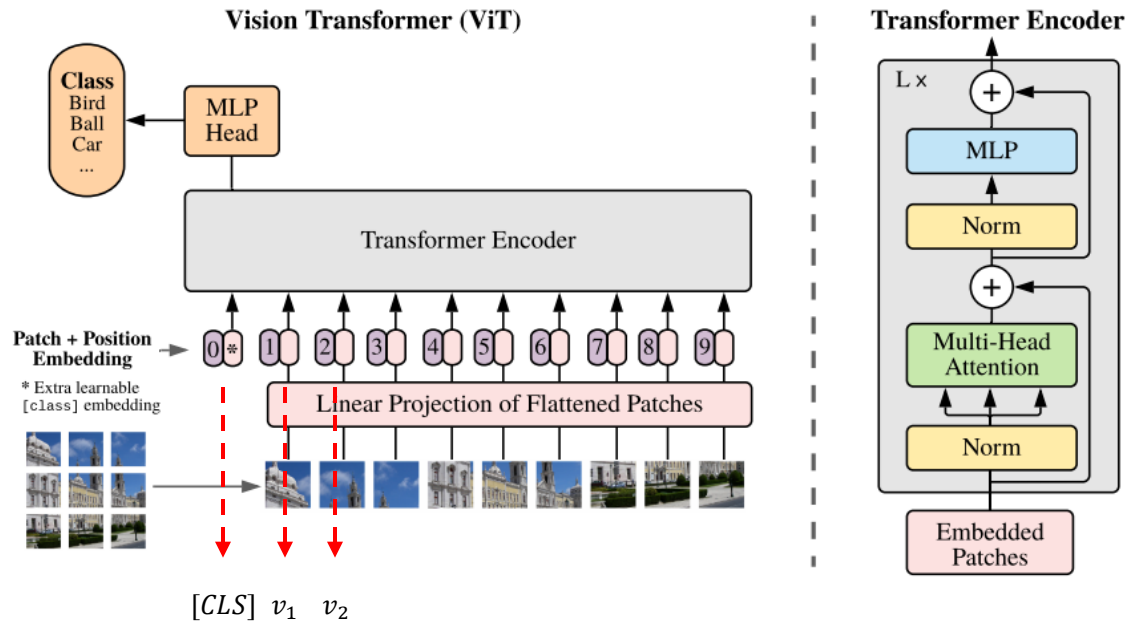


Figure 2: The framework of TextRefiner, which is composed of local cache, feature aggregation and feature alignment. Here, each item in the local cache can be considered as an attribute prior which will be updated by local tokens from the image branch. Therefore, textual class embedding can obtain corresponding linguistic visual attributes by querying this cache.

TextRefiner : Local Cache



- ViT(Vision Transformer)[[link](#)]는 입력 이미지를 **patch** 단위로 분할
- 각 patch를 Embedding Layer에 통과하여 **로컬 토큰** 생성

$$V = \{v_1, v_2, \dots, v_n\}$$

- 로컬 토큰은 **Edge**나 **Texture**, 개념적인 범주를 식별이 가능

- 전체 로컬 토큰을 그냥 저장하는 대신, **고정된 개수의 캐시 항목을 정의**하고, 유사한 토큰들을 클러스터링

- **얼룩말** : 뚜렷한 **흑백 줄무늬**라는 텍스처 정보를 하나의 캐시에 저장
- 초기 캐시의 벡터는 **랜덤 초기화**

$$A \in \mathbb{R}^{M \times d}$$

- 그 다음, 캐시의 정보와 로컬 토큰의 **코사인 유사도** 측정

$$D_{i,j} = \frac{\exp(\cos(v_i, A_j))}{\sum_{j=1}^M \exp(\cos(v_i, A_j))}$$

- 가장 유사도가 높은 인덱스를 구한 후, 이 정보를 기반으로 업데이트 진행

$$G_j = \{i \mid \operatorname{argmax}_k D_{i,k} = j\}$$

$$A_j = \gamma \cdot A_j + (1 - \gamma) \sum_{i \in G_j} D_{i,j} \cdot v_i$$

TextRefiner : Feature Aggregation

- Fine-Grained 정보를 담고 있는 로컬 캐시를 활용하기 위해 **Feature Aggregation** 도입
- 업데이트 된 **로컬 캐시**와 **CLIP의 텍스트 임베딩** E_i 를 잔차 연결 (Residual Connection) 형식으로 연결
 - 텍스트 임베딩과 캐시 항목 간 유사도

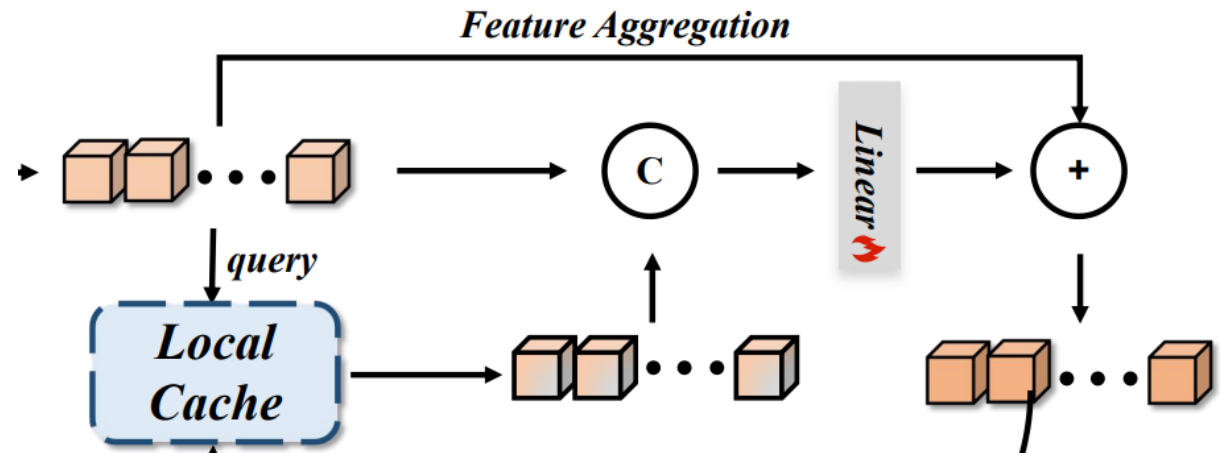
$$W_{i,j} = \frac{\exp(\cos(E_i, A_j))}{\sum_{j=1}^M \exp(\cos(E_i, A_j))}$$

- 집계된 시각 정보 생성

$$\bar{E}_i = \sum_{j=1}^M W_{i,j} \cdot A_j$$

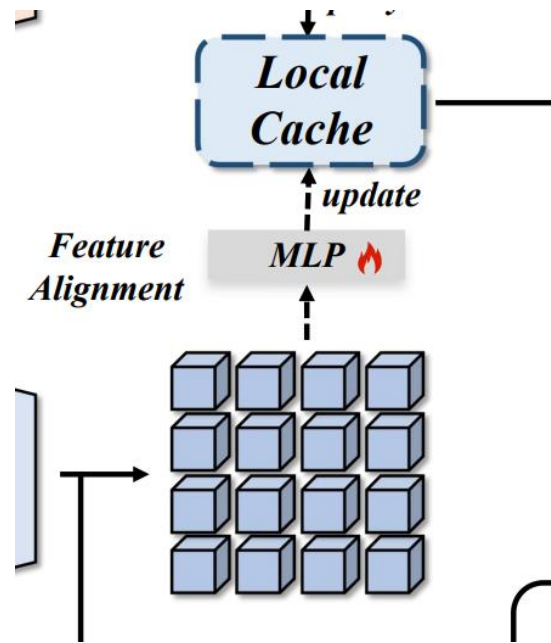
- 텍스트 임베딩과 시각 정보 결합

$$\hat{E}_i = \alpha \cdot \text{Linear}(|E_i, \bar{E}_i|) + E_i$$



TextRefiner : Feature Alignment

- CLIP은 이미지와 텍스트로부터 추출된 전역(Global)적인 특성을 사용
- 그러나, 이미지의 로컬(Local) 특성은 명시적인 정렬이 부족 → 로컬 이미지와 텍스트 임베딩 사이 **모달리티 갭** 존재
- 이러한 간극을 해결하기 위한 **Feature Alignment** 모듈 추가
 - 간단한 2-Layer MLP로 구성
 $\tilde{v} = W_2 \sigma(\text{norm}(W_1, V))$



TextRefiner : Training

- 학습 과정은 기본적인 CLIP 프레임워크를 따르며, **Contrastive Loss**를 주요 손실로 사용
- 추가적으로, TextRefiner는 **Semantic Loss**와 **Regularization Loss** 도 사용

Semantic Loss

- **Feature Alignment** 모듈에서 로컬 이미지 특징(Local visual features)와 대응되는 텍스트 임베딩 E_i 이 정렬되도록 유도하는 손실

$$L_{sem} = \frac{1}{k+1} \sum_{i=1}^{k+1} \log \frac{\exp(\cos(S_i, \hat{E}_c)/\tau)}{\sum_{j=1}^C \exp(\cos(S_i, \hat{E}_j)/\tau)}$$

Regularization Loss

- 제한된 학습 이미지(Few-Shot)으로 인한 과적합 방지를 위한 손실

$$L_{reg} = \|E - \hat{E}\|$$

최종 Loss

$$L = L_{cls} + \lambda_1 \cdot L_{sem} + \lambda_2 \cdot L_{reg}$$

Contents

1. Introduction
2. Related Works
3. Method
- 4. Experiment**
5. Conclusion

Datasets

- **Base-to-novel**

학습된 적 없는 신규 클래스로의 평가

ImageNet, Caltech101, OxfordPets, StanfordCars, Flowers102, Food101 등 다양한 데이터셋 사용

- **Cross-domain**

학습 데이터 : ImageNet

검증 데이터 : ImageNetV2, ImageNet-Sketch, ImageNet-A, ImageNet-R 등

학습은 각 클래스에 대해서 **16-shot**만을 사용, 검증은 모든 데이터셋에 대해 평가

Main Results – Base to Novel

Method	Average			ImageNet			Caltech101			OxfordPets		
	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM
CLIP	69.34	74.22	71.70	72.43	68.14	70.22	96.84	94.00	95.40	91.17	97.26	94.12
CoOp	82.69	63.22	71.66	76.47	67.88	71.92	98.00	89.81	93.73	93.67	95.29	94.47
CoCoOp	80.47	71.69	75.83	75.98	70.43	73.10	97.96	93.81	95.84	95.20	97.69	96.43
PromptSRC	84.26	76.10	79.97	77.60	70.73	74.01	98.10	94.03	96.02	95.33	97.30	96.30
MaPLe	82.28	75.14	78.55	76.66	70.54	73.47	97.74	94.36	96.02	95.43	97.76	96.58
PromptKD	84.11	78.28	81.09	77.63	70.96	74.15	98.31	96.29	97.29	93.42	97.44	95.39
LLaMP	85.16	77.71	81.27	77.99	71.27	74.48	98.45	95.85	97.13	96.31	97.74	97.02
CoOp w/TextRefiner	79.74	74.32	76.94	76.84	70.54	73.56	98.13	94.43	96.24	95.27	97.65	96.45
PromptKD w/TextRefiner	85.22	79.64	82.33	77.51	71.43	74.35	98.52	96.52	97.51	95.60	97.90	96.74

Method	StanfordCars			Flowers102			Food101			FGVCAircraft		
	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM
CLIP	63.37	74.89	68.65	72.08	77.80	74.83	90.10	91.22	90.66	27.19	36.29	31.09
CoOp	78.12	60.40	68.13	97.60	59.67	74.06	88.33	82.26	85.19	40.44	22.30	28.75
CoCoOp	70.49	73.59	72.01	94.87	71.75	81.71	90.70	91.29	90.99	33.41	23.71	27.74
PromptSRC	78.27	74.97	76.58	98.07	76.50	85.95	90.67	91.53	91.10	42.73	37.87	40.15
MaPLe	72.94	74.00	73.47	95.92	72.46	82.56	90.71	92.05	91.38	37.44	35.61	36.50
PromptKD	80.48	81.78	81.12	98.69	81.91	89.52	89.43	91.27	90.34	43.61	39.68	41.55
LLaMP	81.56	74.54	77.89	97.82	77.40	86.42	91.05	91.93	91.49	47.30	37.61	41.90
CoOp w/TextRefiner	71.40	70.90	71.15	95.92	74.33	83.76	90.88	91.43	91.15	35.35	35.87	35.61
PromptKD w/TextRefiner	80.91	81.83	81.37	99.30	82.91	90.37	91.42	92.71	92.06	45.01	40.12	42.42

Method	SUN397			DTD			EuroSAT			UCF101		
	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM
CLIP	69.36	75.35	72.23	53.24	59.90	56.37	56.48	64.05	60.03	70.53	77.50	73.85
CoOp	80.60	65.89	72.51	79.44	41.18	54.24	92.19	54.74	68.69	84.69	56.05	67.46
CoCoOp	79.74	76.86	78.27	77.01	56.00	64.85	87.49	60.04	71.21	82.33	73.45	77.64
PromptSRC	82.67	78.47	80.52	83.37	62.97	71.75	92.90	73.90	82.32	87.10	78.80	82.74
MaPLe	80.82	78.70	79.75	80.36	59.18	68.16	94.07	73.23	82.35	83.00	78.66	80.77
PromptKD	82.53	80.88	81.70	82.86	69.15	75.39	92.04	71.59	80.54	86.23	80.11	83.06
LLaMP	83.41	79.90	81.62	83.49	64.49	72.77	91.93	83.66	87.60	87.13	80.66	83.77
CoOp w/TextRefiner	80.96	76.49	78.66	75.35	58.09	65.60	74.57	72.82	73.68	82.52	75.01	78.59
PromptKD w/TextRefiner	83.02	80.50	81.74	83.91	71.01	76.92	92.99	79.22	85.55	89.20	81.90	85.39

11개의 벤치마크로 실험

- CoOp : 신규 클래스에 대한 일반화 성능 크게 증가
63.22% → 74.32%
- PromptKD[link] : 본 방법론 적용 시 성능 향상
DTD : 69.15% → 71.01%
EuroSAT : 71.59% → 79.22%
- LLaMP[link] : 클래스 명에 따라 LLM을 활용하여 상세한 설명 제공
PromptKD + TextRefiner > LLaMP

Main Results – Cross Domain

Method	Source	Target			
	ImageNet	-V2	-Sketch	-A	-R
CLIP	66.73	60.83	46.15	47.77	73.96
CoOpOp	71.02	64.07	48.75	50.63	76.18
PromptSRC	71.27	64.35	49.55	50.90	77.80
CoOp	71.51	64.20	47.99	49.71	75.21
+ TextRefiner	72.06	65.02	48.58	49.77	76.30
MaPLe	70.72	64.07	49.15	50.90	76.98
+ TextRefiner	71.13	64.54	49.08	51.49	77.71

TextRefiner의 도메인 간 일반화 성능을 평가

CoOp, MaPLe[\[link\]](#) 방법론에 TextRefiner 적용 시 도메인 간 일반화 성능이 향상 되는 것을 확인

Table 2: Comparison between our method and other existing methods on cross-domain generalization. Models will be trained with 16-shot ImageNet and test in out-of-distribution datasets.

Contents

1. Introduction
2. Related Works
3. Method
4. Experiment
- 5. Conclusion**

Conclusion

- 본 논문에서는 **로컬 토큰**에 내제된 시각 개념을 활용하여, 텍스트 프롬프트에 Fine-grained한 정보를 더함으로 표현력을 강화하는 방법을 제안
- **TextRefiner**는 **Plug-and-Play** 방식으로 기존의 다양한 방법론에 병합 가능하며, 추가적인 연산 부담이 거의 없이 적용 가능
- **TextRefiner**의 세 가지 구성 요소
 - **Local Cache** : 로컬 토큰에서 얻은 세밀한 정보를 저장
 - **Feature Aggregation** : 전역 정보와 로컬 정보를 통합
 - **Feature Alignment** : 로컬 토큰과 텍스트 임베딩의 Modality Gap 완화
- 본 연구는 VLM의 표현 능력을 향상시키고, 적은 데이터 환경에서 효율적인 전이 학습을 할 수 있는 통찰을 제공