Vision Language Model

Large Multi-Modal Model

Contents

- ViT (Vision Transformer) – 2021 Google
- CLIP (Contrastive Learning Image Pre-training – OpenAI 2021
- LLaVA (Large Language and Vision Assistant) – 2023 Open Source
- Post LLaVAs from LLaVA – 2024 Open Source

ViT (Vision Transformer)

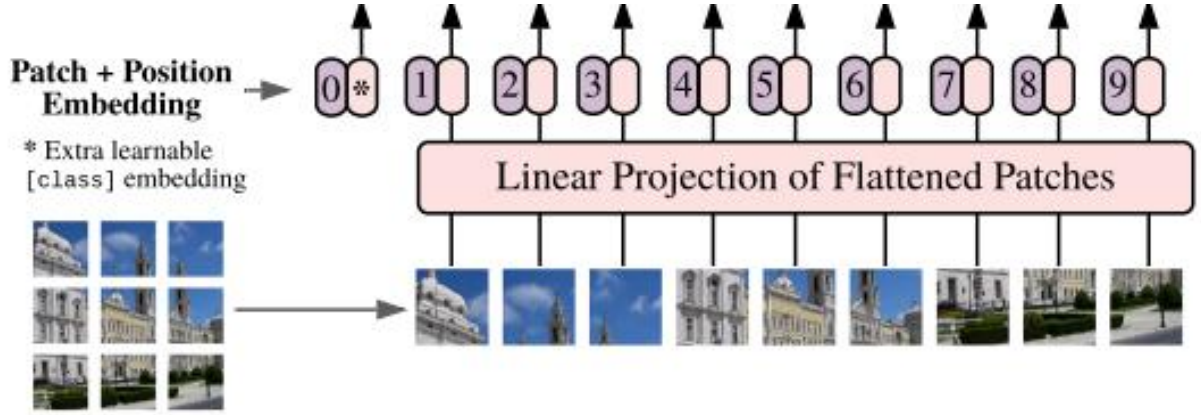
An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale (2021 Google Research)

ViT (Vision Transformer)

- ViT 이전까지는 CNN (Convolutional Neural Network)이 Image, Audio, Video에 적용되는 대표적인 Neural Network이었다.
- Google에서 Transformer network를 image classification에 적용하여 좋은 성적 – Accuracy & Scale 측면 – SOTA 얻음
- ViT 이후로 Transformer Network를 이용한 Image & Video 연구가 많이 진행되었고, 지금은 Transformer Network으로 많이 대체되는 추세

How To Make Sequence of Vectors?

- 이미지를 고정 크기의 패치(patch)로 나눈 후 이를 일종의 토큰으로 간주하여 임베딩
- 이미지 → 패치 분할 → 벡터화 → 선형 변환



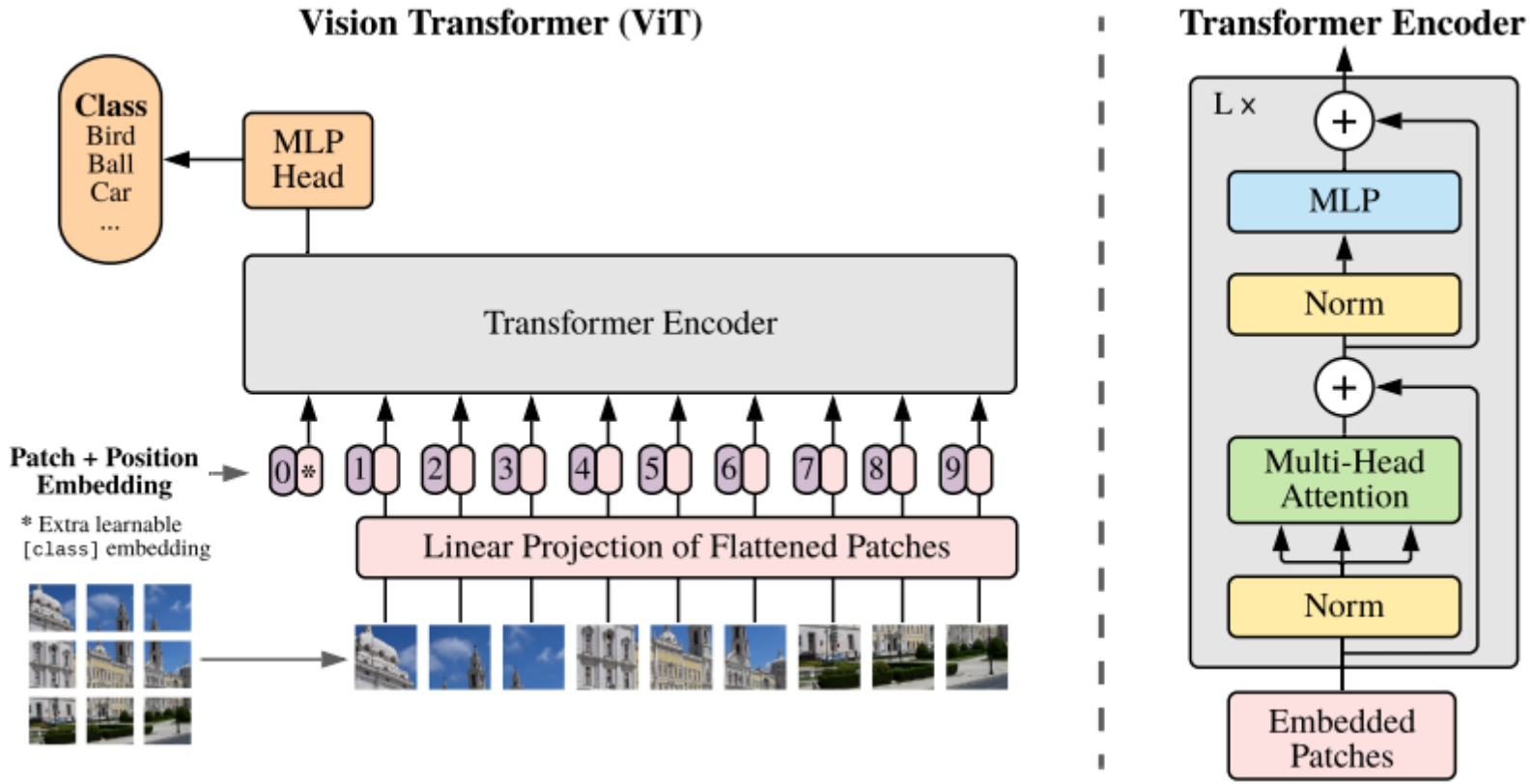
- 구현은

① conv2D (stride를 patch size로 해서) 한 후 flatten

② unfold 함수를 사용한 후 Linear 적용

- 일반적으로 conv2D 방식 선호 – 빠르고 메모리 효율 높음

Model Overview



모델 및 성능

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Model Overview

2

CLIP (Contrastive Language-Image Pre-training)

Learning Transferable Visual Models From Natural Language Supervision (OpenAI 2021)

Learning Image Representation from Natural Language

- **Old Approaches**
  - ImageNet – 1M images (crowd-sourced labeling), 1K classes
  - Weakly Supervised Learning – ImageNet으로 pre-training하고, JFT-300M dataset (weakly supervised dataset)으로 training

- **Recent Approach – Natural Language Supervision**
  - Image와 class label과의 연결을 학습하는 것이 아니라, Image caption의 Natural language를 training signal로 사용

- **Natural Language Supervision Strengths**
  - Dataset를 Web Scale로 확장하기 용이하다.
  - Image representation뿐만 아니라, 언어와의 관계를 배우기 때문에 다양한 Zero-shot Transfer가 가능하다.

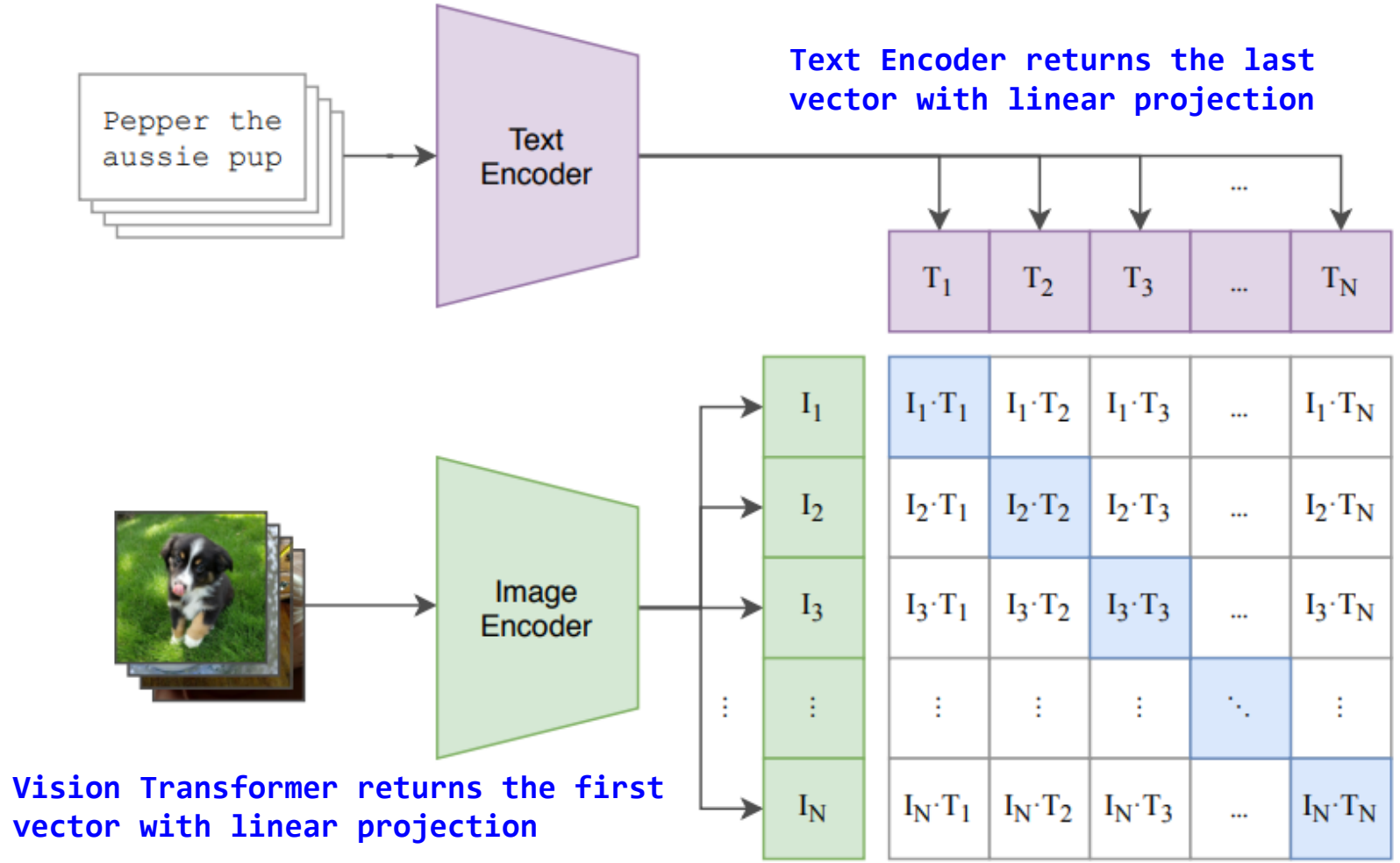
WIT (WebImageText) Dataset

- 400 Million (image, text) Pairs – collected from a variety of publicly available sources on the Internet
  - 500,000 Queries (English version of Wikipedia에서 100번 이상 나온 단어들과 여기서 bi-grams으로 확장하고, WordNet synsets로부터 추가)
  - Query당 최대 20,000 (image, text) pairs를 포함하여, class balance를 유지함

Training Method

- Naïve Approach – 각 image에 대하여 text에 있는 정확한 words를 predict하도록 함 (순서는 상관없는 Bag of words)
  - 어려운 Task – descriptions, comments 가 매우 다양하고 유사한 text가 다른 images에 나타날 수 있음
- Contrastive Representation Learning – 간단하고 더 효과적인 Representation을 학습
  - 어떤 text (전체 text)가 어떤 image와 더 잘 paring되는가? 를 학습하는 것이 더 배우기 쉽다.
- Linear Projection to multi-modal embedding space
  - Image Feature 와 Text Feature를 Contrastive Representation Learning하기 위해서 같은 multi-modal embedding space로 mapping해야함
  - 복잡한 non-linear mapping 대신에 간단한 Linear Projection을 사용

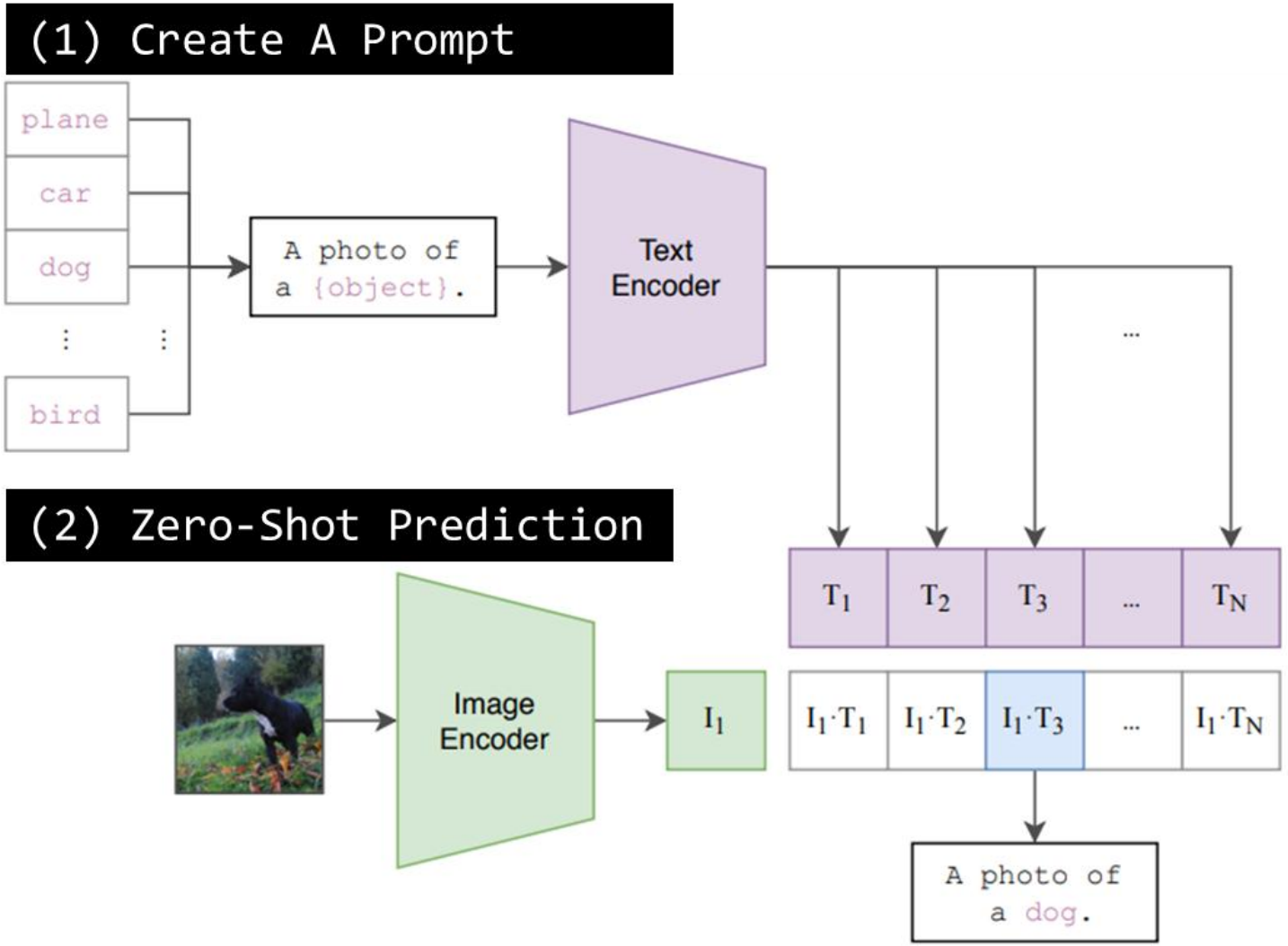
Contrastive Pre-Training



Pseudo Code for Contrastive Pre-Training

```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

Inference in CLIP

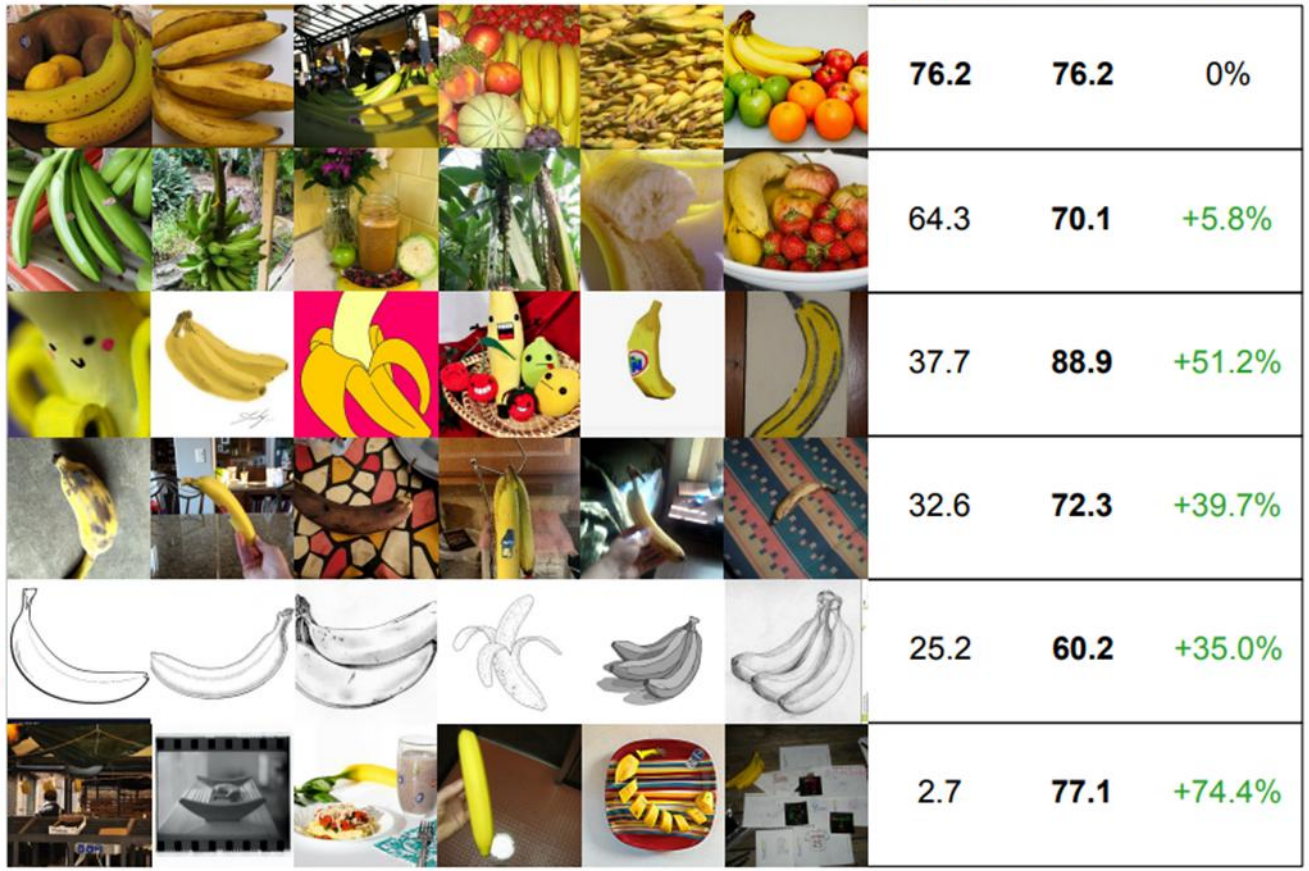


CLIP – Linear Prob Performance

Linear Prob: Freeze pre-trained model and only train linear layer using logits from the model.

		Food101	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech101	Flowers	MNIST	FER2013	STL10*	EuroSAT	RESISC45	GTSRB	KITTI	Country211	PCAM	UCF101	Kinetics700	CLEVR	HatefulMemes	SST	ImageNet
LM RN50		81.3	82.8	61.7	44.2	69.6	74.9	44.9	85.5	71.5	82.8	85.5	91.1	96.6	60.1	95.3	93.4	84.0	73.8	70.2	19.0	82.9	76.4	51.9	51.2	65.2	76.8	65.2
CLIP-RN	50	86.4	88.7	70.3	56.4	73.3	78.3	49.1	87.1	76.4	88.2	89.6	96.1	98.3	64.2	96.6	95.2	87.5	82.4	70.2	25.3	82.7	81.6	57.2	53.6	65.7	72.6	73.3
	101	88.9	91.1	73.5	58.6	75.1	84.0	50.7	88.0	76.3	91.0	92.0	96.4	98.4	65.2	97.8	95.9	89.3	82.4	**73.6**	26.6	82.8	84.0	60.3	50.3	68.2	73.3	75.7
	50x4	91.3	90.5	73.0	65.7	77.0	85.9	57.3	88.4	79.5	91.9	92.5	97.8	98.5	68.1	97.8	96.4	89.7	85.5	59.4	30.3	83.0	85.7	62.6	52.5	68.0	76.6	78.2
	50x16	93.3	92.2	74.9	72.8	79.2	88.7	62.7	**89.0**	79.1	93.5	93.7	98.3	**98.9**	68.7	98.6	97.0	91.4	89.0	69.2	34.8	83.5	88.0	66.3	53.8	71.1	**80.0**	81.5
	50x64	94.8	94.1	78.6	77.2	81.1	90.5	67.7	**88.9**	**82.0**	94.5	95.4	98.9	**98.9**	**71.3**	99.1	97.1	92.8	90.2	69.2	40.7	83.7	89.5	69.1	55.0	**75.0**	**81.2**	83.6
CLIP-ViT	B/32	88.8	95.1	80.5	58.5	76.6	81.8	52.0	87.7	76.5	90.0	93.0	96.9	**99.0**	69.2	98.3	97.0	90.5	85.3	66.2	27.8	83.9	85.5	61.7	52.1	66.7	70.8	76.1
	B/16	92.8	96.2	83.1	67.8	78.4	86.7	59.5	**89.2**	79.2	93.1	94.7	98.1	**99.0**	69.5	99.0	97.1	92.7	86.6	67.8	33.3	83.5	88.4	66.1	**57.1**	70.3	75.5	80.2
	L/14	95.2	98.0	87.5	77.0	**81.8**	**90.9**	69.4	**89.6**	**82.1**	**95.1**	**96.5**	99.2	**99.2**	**72.2**	**99.7**	**98.2**	94.1	**92.5**	64.7	42.9	85.8	**91.5**	72.0	**57.8**	**76.2**	**80.8**	83.9
	L/14-336px	**95.9**	97.9	87.4	**79.9**	**82.2**	**91.5**	**71.6**	**89.9**	**83.0**	**95.1**	**96.0**	99.2	**99.2**	**72.9**	**99.7**	**98.1**	**94.9**	**92.4**	69.2	**46.4**	85.6	**92.0**	**73.0**	**60.3**	**77.3**	**80.5**	85.4

Robustness of Zero-Shot to Distribution Shift



	Dataset Examples	ResNet101	CLIP	Δ Score
ImageNet		**76.2**	**76.2**	0%
ImageNetV2		64.3	**70.1**	+5.8%
ImageNet-R		37.7	**88.9**	+51.2%
ObjectNet		32.6	**72.3**	+39.7%
ImageNet Sketch		25.2	**60.2**	+35.0%
ImageNet-A		2.7	**77.1**	+74.4%

3

LLaVA (Large Language and Vision Assistant)

Visual Instructional Tuning (U. of Wisconsin-Madison, Microsoft Research, Columbia University)

LLaVA Features

① **Multimodal Instruction-Following Data**

- Image-Text Pair Data를 ChatGPT/GPT-4를 이용해서 만드는 방법 제안

② **Large Multimodal Models**:

- 기존의 Visual Encoder CLIP과 LLM Vicuna를 이용해서 Large Multimodal Model 만드는 방법 제안

③ **Open-source**:

- Multimodal Instruction Data, Codebase, Model Checkpoints, and A Visual Chat Demo

④ **Subsequent Researches**:

- LLaVA-1.5, LLaVA-Next, LLaVA-OneVision, LLaVA-Video 등 후속 연구 발표

GPT-assisted Visual Instruction Data Generation

From Context Types

Context type 1: Captions
A group of people standing outside of a black vehicle with various luggage.
Luggage surrounds a vehicle in an underground parking area
People try to fit all of their luggage in an SUV.
The sport utility vehicle is parked in the public garage, being packed for a trip
Some people with luggage near a van that is transporting it.

Context type 2: Boxes
person: [0.681, 0.242, 0.774, 0.694], person: [0.63, 0.222, 0.686, 0.516], person: [0.444, 0.233, 0.487, 0.34], backpack: [0.384, 0.696, 0.485, 0.914], backpack: [0.755, 0.413, 0.846, 0.692], suitcase: [0.758, 0.413, 0.845, 0.69], suitcase: [0.1, 0.497, 0.173, 0.579], bicycle: [0.282, 0.363, 0.327, 0.442], car: [0.786, 0.25, 0.848, 0.322], car: [0.783, 0.27, 0.827, 0.335], car: [0.86, 0.254, 0.891, 0.3], car: [0.261, 0.101, 0.787, 0.626]

Generate Response Types

Response type 1: conversation
Question: What type of vehicle is featured in the image?
Answer: The image features a black sport utility vehicle (SUV).
Question: Where is the vehicle parked?
Answer: The vehicle is parked in an underground parking area, likely in a public garage.
Question: What are the people in the image doing?
Answer: The people are trying to fit all of their luggage into the SUV, likely preparing for a trip.

Response type 2: detailed description
The image is an underground parking area with a black sport utility vehicle (SUV) parked. There are three people in the scene, with one person standing closer to the left side of the vehicle, another person in the middle, and the third person on the right side. They are all working together to pack their luggage into the SUV for a trip.
In the parking area, various luggage items are scattered around the vehicle. There are two backpacks, one located near the left rear wheel and the other closer to the right side of the vehicle. Additionally, there are two suitcases, one on the right side of the car and another further away near the center of the parking area. A bicycle can also be seen on the left side of the vehicle.
Other cars are parked around the main SUV, with one car positioned behind it and slightly to the left, another behind and slightly to the right, and the third car further behind on the right side.

Response type 3: complex reasoning
Question: What challenges do these people face?
Answer: In the image, a group of people is standing outside a black SUV in a parking area, surrounded by various pieces of luggage, including suitcases and backpacks. They are facing the challenge of fitting all their luggage into the black SUV. There are multiple suitcases and backpacks to be packed, which suggests that the group has a significant amount of belongings to accommodate. They might have to strategize and arrange the luggage efficiently to ensure that everything fits properly into the vehicle. Additionally, they need to consider the comfort of the passengers and visibility while driving, so the placement of the luggage must not obstruct the driver's view or make the passengers uncomfortable during the trip.

Prompts for GPT-4

```
messages = [ {"role":"system", "content": f"""You are an AI visual assistant, and you are seeing a single image. What you see are provided with five sentences, describing the same image you are looking at. Answer all questions as you are seeing the image.

Design a conversation between you and a person asking about this photo. The answers should be in a tone that a visual AI assistant is seeing the image and answering the question. Ask diverse questions and give corresponding answers.

Include questions asking about the visual content of the image, including the object types, counting the objects, object actions, object locations, relative positions between objects, etc. Only include questions that have definite answers:
(1) one can see the content in the image that the question asks about and can answer confidently;
(2) one can determine confidently from the image that it is not in the image. Do not ask any question that cannot be answered confidently.

Also include complex questions that are relevant to the content in the image, for example, asking about background knowledge of the objects in the image, asking to discuss about events happening in the image, etc. Again, do not ask about uncertain details. Provide detailed answers when answering complex questions. For example, give detailed examples or reasoning steps to make the content more convincing and well-organized. You can include multiple paragraphs if necessary."""}
]
for sample in fewshot_samples:
    messages.append({"role":"user", "content":sample['context']})
    messages.append({"role":"assistant", "content":sample['response']} )
messages.append({"role":"user", "content":'\n'.join(query)})
```

Example Samples – Conversation

Captions
There is a movie theater that displays the show times above the doors.
A red fire hydrant is deep in the snow.
The fire hydrant is in the snow near a recently plowed sidewalk.
This city has had a very hard winter with snow.
A hotel for dogs in the snow in winter.

Conversation
Question:
What color is the fire hydrant in the image?
===
Answer:
The fire hydrant in the image is red.
===
Question:
Is the fire hydrant fully covered by snow or partially visible?
===
Answer:
The fire hydrant is partially visible, with a significant portion of it covered by the deep snow.
===
Question:
What is the condition of the sidewalk near the fire hydrant?
===
Answer:
The sidewalk near the fire hydrant has been recently plowed, so it is relatively clear of snow compared to the surrounding area.
===
Question:
What does the image suggest about the winter season in the city?
===
Answer:
The image suggests that the winter season in the city has been quite harsh and challenging. The significant snow accumulation seen in the image indicates that the city has experienced heavy snowfall, which can lead to various difficulties and disruptions for its residents.

For instance, the deep snow covering the fire hydrant could potentially hinder its accessibility in case of emergencies, which poses safety concerns. The recently plowed sidewalk implies that the city's maintenance crews have been actively working to keep the walkways clear and safe for pedestrians, but the sheer amount of snowfall might make it difficult to keep up with the continuous snow removal efforts.

Furthermore, such extreme winter conditions can impact transportation, leading to traffic issues, delays in public transit, and increased risks of accidents due to slippery roads. It can also cause problems for businesses and schools, as people might face challenges commuting to work or attending classes. Additionally, the heavy snow can put extra strain on infrastructure, such as roofs and power lines, increasing the likelihood of structural damage or power outages.

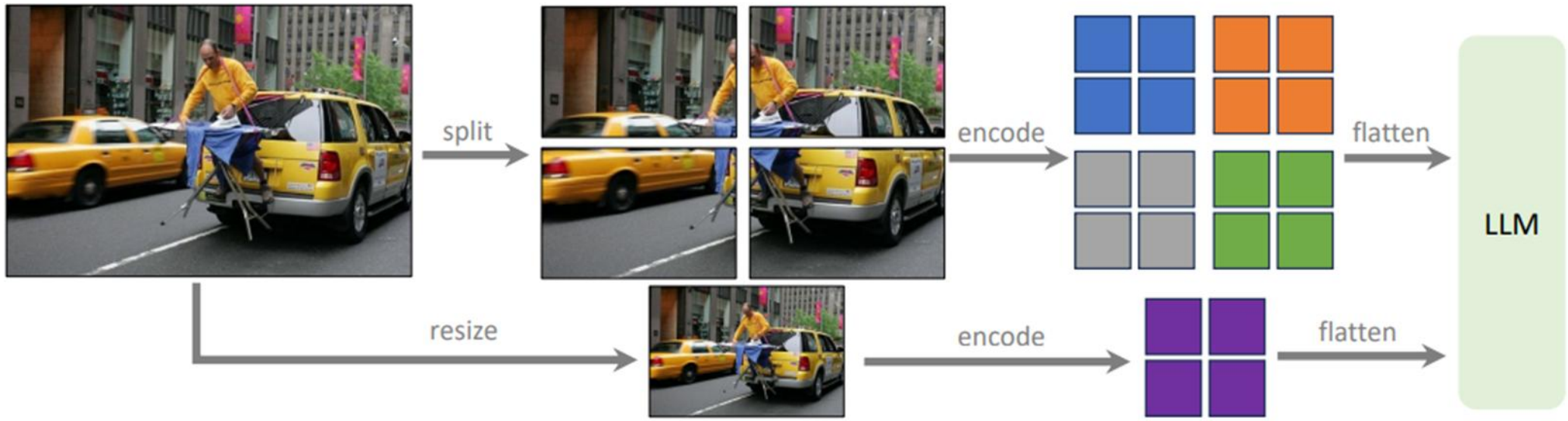
In conclusion, the image of the red fire hydrant deep in the snow and the recently plowed sidewalk suggest that the city has faced a particularly severe winter season, with substantial snowfall that has likely caused various challenges and disruptions for its residents and infrastructure.

Datasets

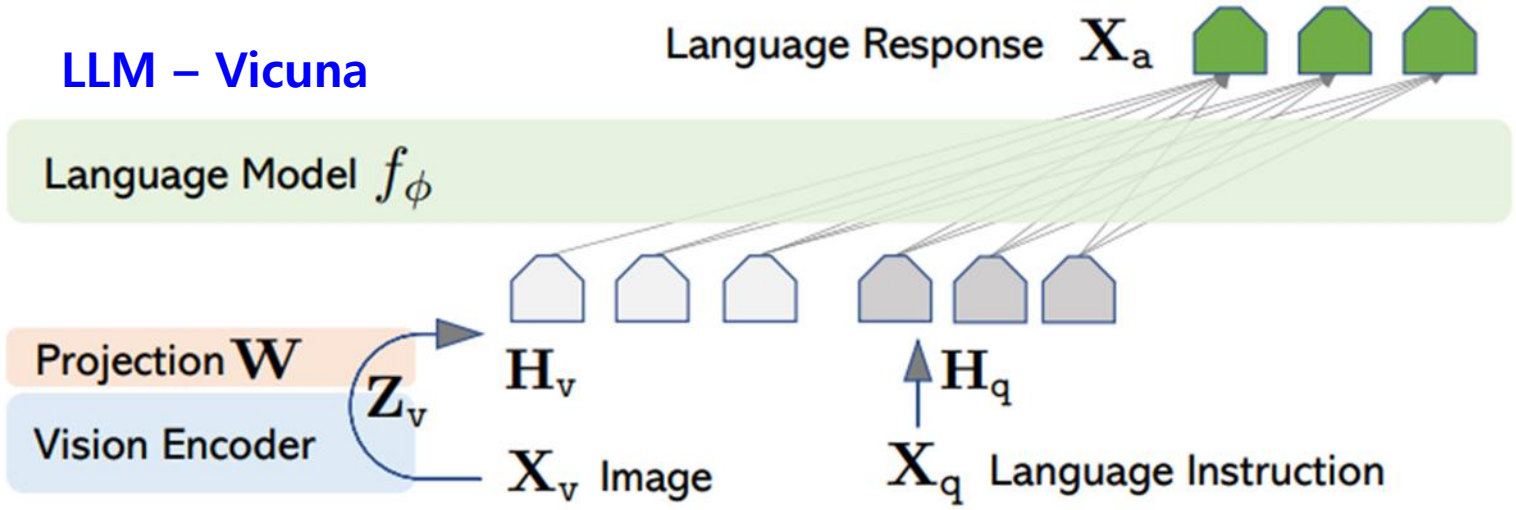
- **Dataset 1: For Fine-tuning Instruction (https://llava-vl.github.io/)**
- 158K language-image instruction-following samples From COCO Dataset
  - 58K – conversations
  - 23K – detailed descriptions
  - 77K – complex reasoning

- **Dataset 2: For Pre-training (https://ai.google.com/research/ConceptualCaptions/)**
- 595K image-text pairs from CC3M (Conceptual Caption 3 Million)

How to Get Image Features?



Large Multimodal Model



Pre-trained CLIP Vision Encoder

Training Sample

- **For each image X_v, generate multi-turn conversation data $(X_q^1, X_a^1, \ldots, X_q^T, X_a^T)$, where T is the total number of turns. X_v 와 나머지 text와의 결합은?**
  - $X_v, X_q^1, X_a^1, \ldots, X_q^T, X_a^T$ **또는,** $X_q^1, X_v, X_a^1, \ldots, X_q^T, X_a^T$
- **Training Sample Example**

$\mathbf{X}_{\texttt{system-message}}$ <STOP>
Human : $\mathbf{X}^1_{\texttt{instruct}}$ <STOP> Assistant: $\mathbf{X}^1_{\texttt{a}}$ <STOP>
Human : $\mathbf{X}^2_{\texttt{instruct}}$ <STOP> Assistant: $\mathbf{X}^2_{\texttt{a}}$ <STOP> ...

Table 2: The input sequence used to train the model. Only two conversation turns are illustrated here; in practice, the number of turns varies based on the instruction-following data. In our current implementation, we follow Vicuna-v0 [9] to set the system message $\mathbf{X}_{\texttt{system-message}}$ and we set <STOP> = ###. The model is trained to predict the assistant answers and where to stop, and thus only green sequence/tokens are used to compute the loss in the auto-regressive model.

Training Phases – 2 Steps

Stage 1: Pre-training for Feature Alignment

- 595K image-text pairs from CC3M
- Vision Encoder CLIP and Vicuna are frozen
- Only Projection Matrix W are learned

Stage 2: Fine-tuning End-to-End

- Generated Datasets 158K from COCO with GPT-4 are used
- Vision Encoder CLIP's weight are frozen
- Projection Matrix W and Vicuna Weights are trained

- **Relatively Simple Training – 8× A100s for one day**

Improvements on LLaVA

- LLaVA-1.5
- LLaVA-NeXt
- LLaVA-OneVision
- LLaVA-Video

- Demo – LLaVA-NeXT Interleave

https://huggingface.co/spaces/lmms-lab/LLaVA-NeXT-Interleave-Demo