Language and Vision

Project Progress (+1 participation)

Language and Vision

Language and Vision: Joint understanding of both image/video and text data

Popular Tasks

- Image Captioning
- Visual Question Answering
- Visual Entailment
- Visual Storytelling
- Visual Reasoning
- Image-Text Retrieval
- Vision and Language Navigation
- Video Understanding
- ….

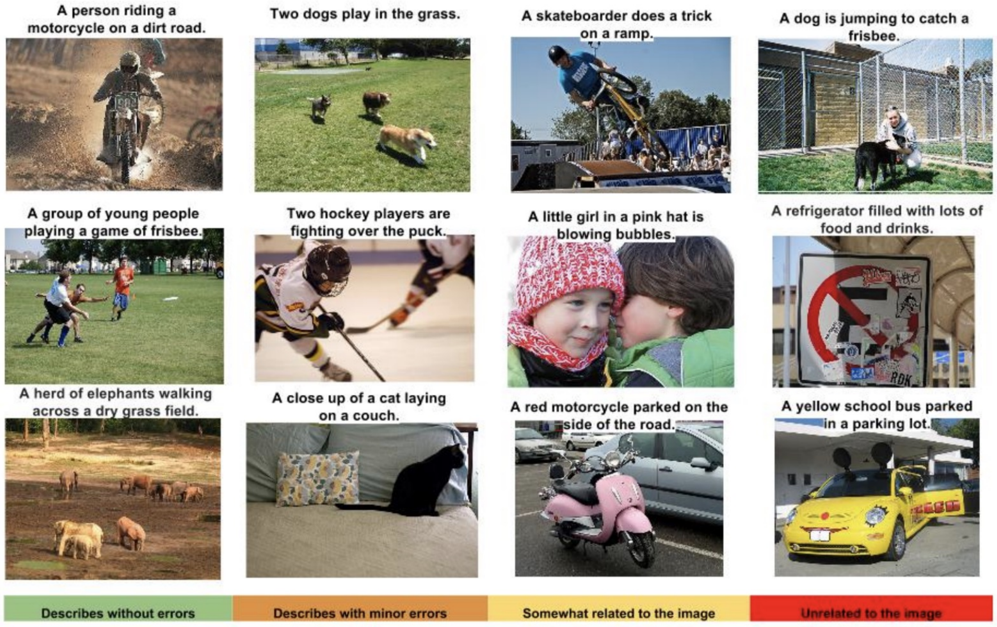
Image Captioning

Every picture tells a story: Generating sentences from images (ECCV 2010)

Baby talk: Understanding and generating simple image descriptions (CVPR 2011)

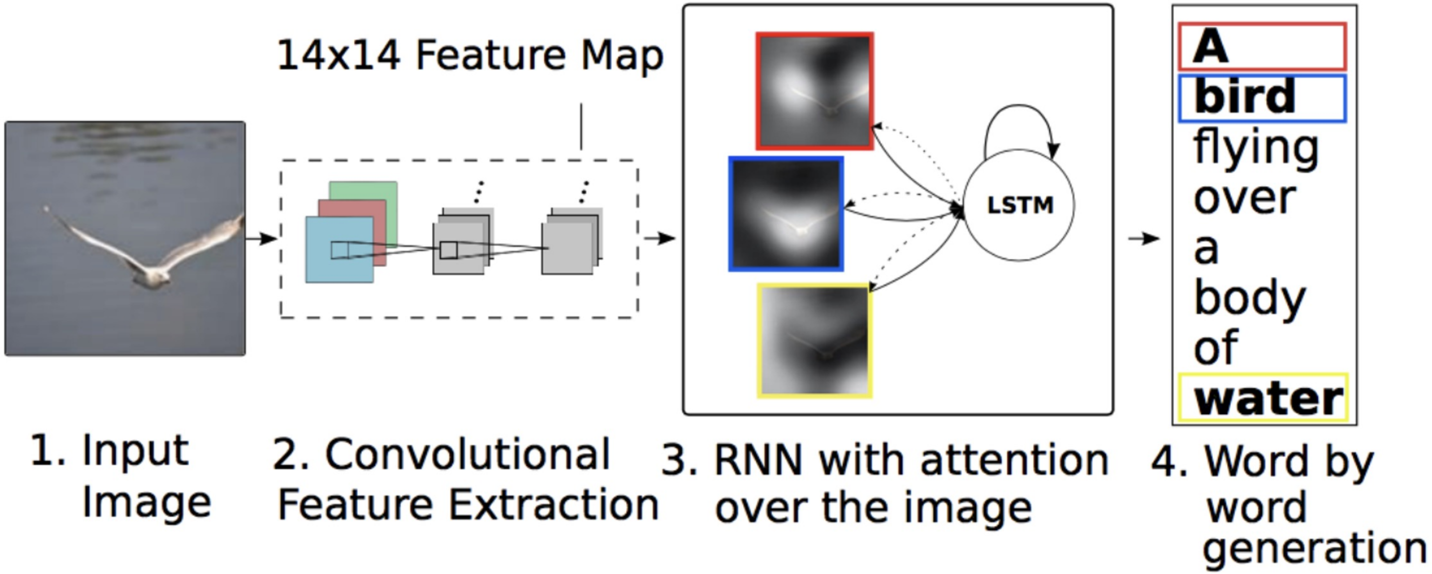
Show and Tell: A Neural Image Caption Generator (CVPR 2015)

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention (ICML 2015)


A person riding a motorcycle on a dirt road.
Two dogs play in the grass.
A skateboarder does a trick on a ramp.
A dog is jumping to catch a frisbee.
A group of young people playing a game of frisbee.
Two hockey players are fighting over the puck.
A little girl in a pink hat is blowing bubbles.
A refrigerator filled with lots of food and drinks.
A herd of elephants walking across a dry grass field.
A close up of a cat laying on a couch.
A red motorcycle parked on the side of the road.
A yellow school bus parked in a parking lot.
Describes without errors
Describes with minor errors
Somewhat related to the image
Unrelated to the image

Attention in Image Captioning

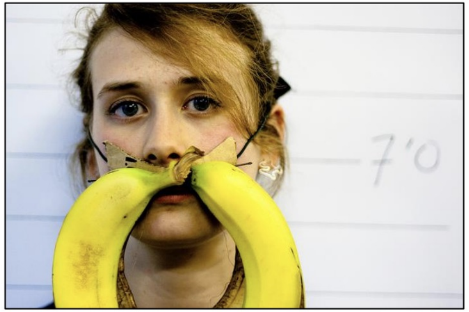
Show, Attend and Tell: Neural Image Caption Generation with Visual Attention


14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

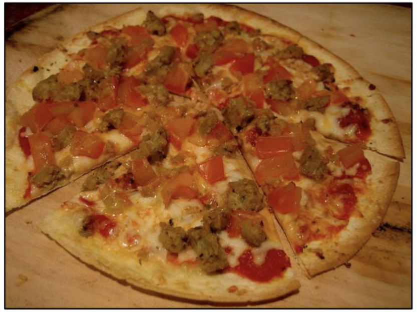
Visual Question Answering

VQA: Visual Question Answering (ICCV 2015)

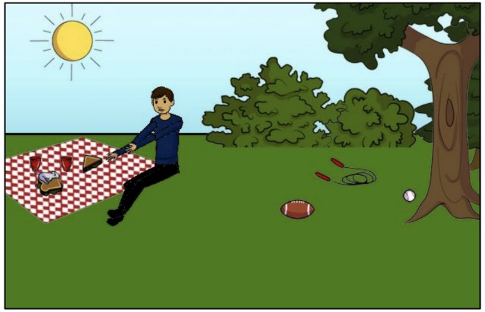
https://visualqa.org/



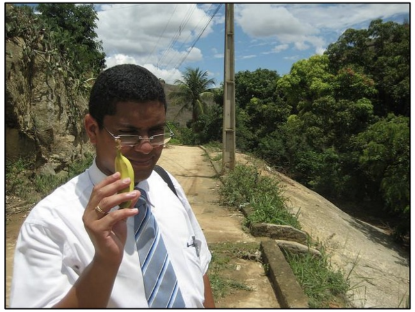
What color are her eyes?
What is the mustache made of?



How many slices of pizza are there?
Is this a vegetarian pizza?



Is this person expecting company?
What is just under the tree?

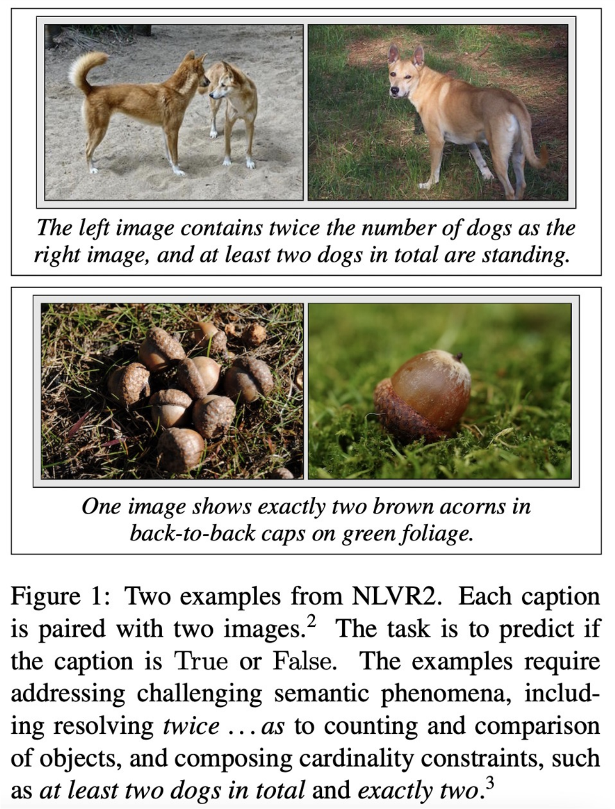


Does it appear to be rainy?
Does this person have 20/20 vision?

Natural Language for Visual Reasoning

A corpus for reasoning about natural language grounded in photographs ACL 2019

A corpus of natural language for visual reasoning. ACL 2017

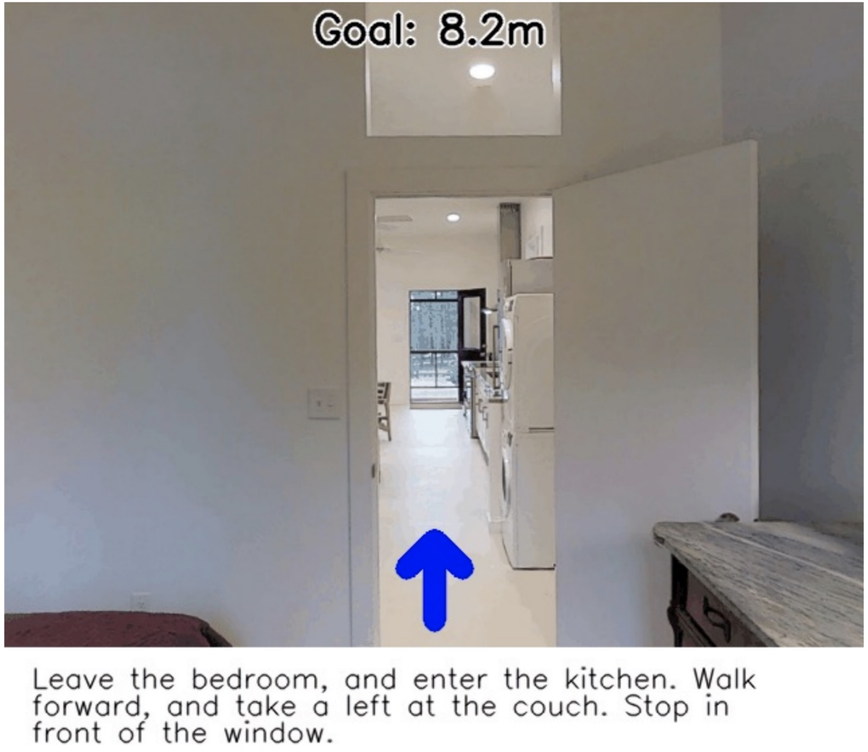


The left image contains twice the number of dogs as the right image, and at least two dogs in total are standing.

One image shows exactly two brown acorns in back-to-back caps on green foliage.

Figure 1: Two examples from NLVR2. Each caption is paired with two images.[2] The task is to predict if the caption is True or False. The examples require addressing challenging semantic phenomena, including resolving *twice* ... *as* to counting and comparison of objects, and composing cardinality constraints, such as *at least two dogs in total* and *exactly two*.[3]

Vision and Language Navigation

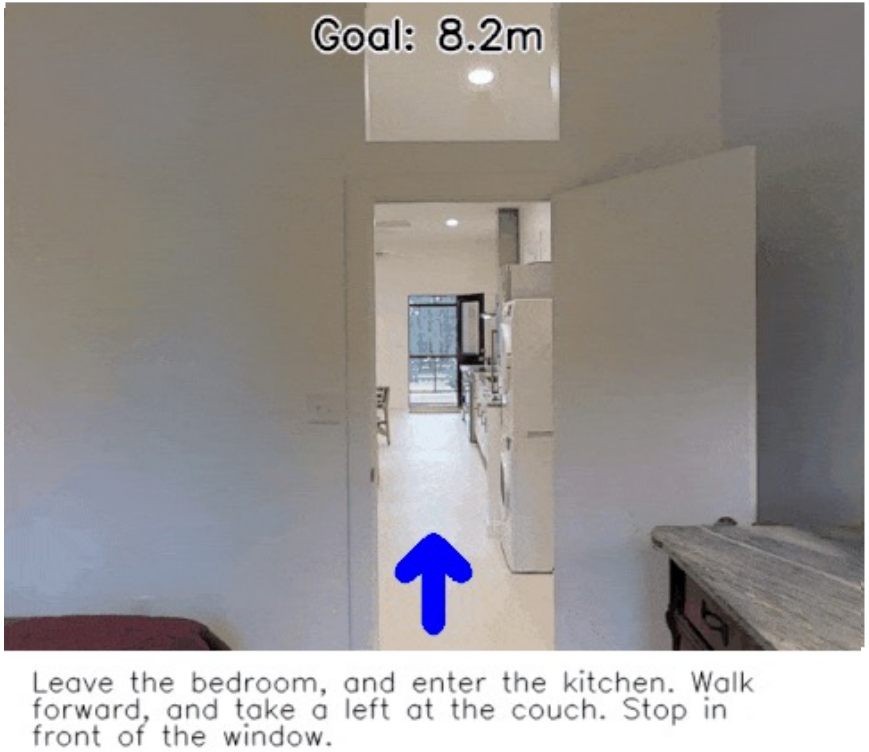


Leave the bedroom, and enter the kitchen. Walk forward, and take a left at the couch. Stop in front of the window.

Vision-and-Language Navigation (VLN): an embodied agent is placed at a spot in a photo-realistic environment;

https://datarelease.blob.core.windows.net/tutorial/VQA2VLN2021/VLN_part1.pdf

Vision and Language Navigation



Leave the bedroom, and enter the kitchen. Walk forward, and take a left at the couch. Stop in front of the window.

Vision-and-Language Navigation (VLN): an embodied agent is placed at a spot in a photo-realistic environment;

The agent is called to navigate to a specific spot based on given natural language instructions.

https://bringmeaspoon.org

Video Understanding

MSR-VTT (Microsoft Research Video to Text) is a large-scale dataset for the open domain video captioning, which consists of 10,000 video clips from 20 categories, and each video clip is annotated with 20 English sentences by Amazon Mechanical Turks.

- Video Captioning
- Text-to-Video Retrieval
- Video QA

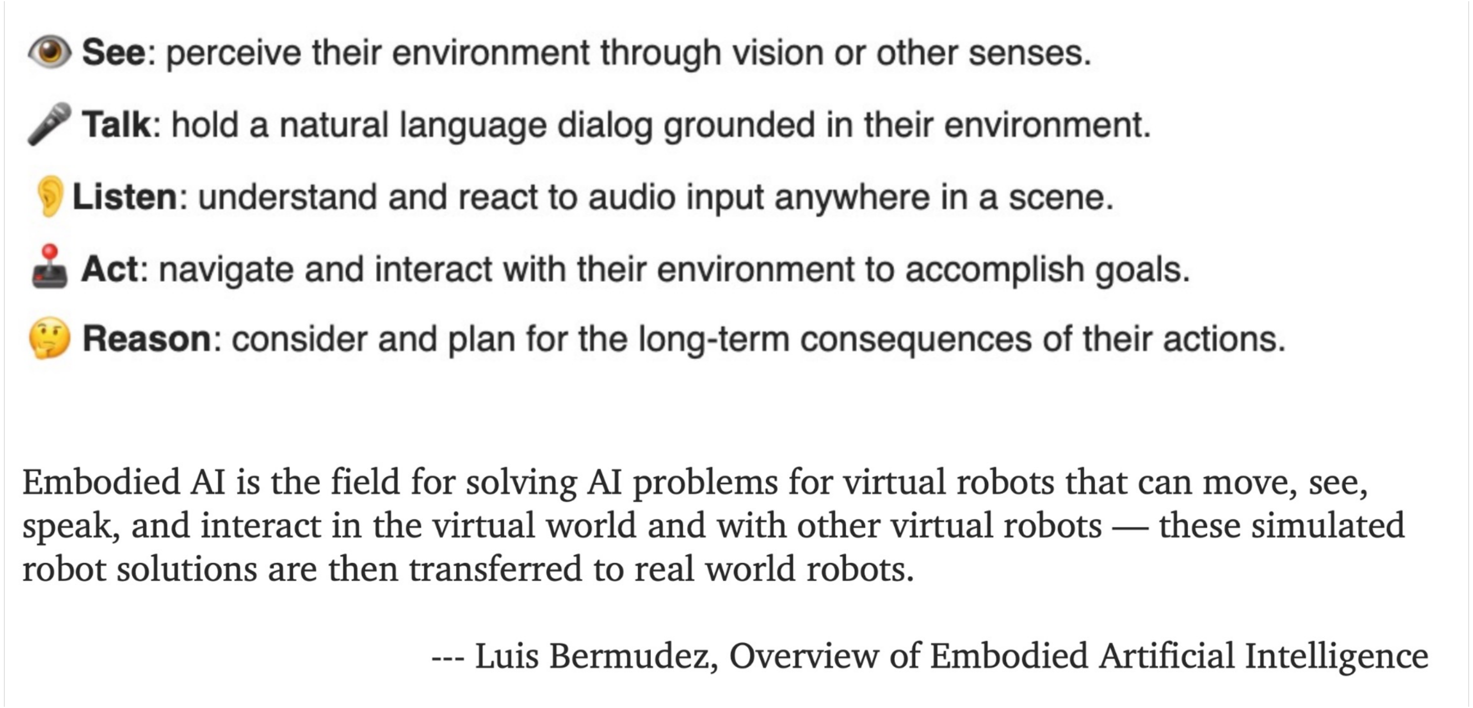
1. A child is cooking in the kitchen.
2. A girl is putting her finger into a plastic cup containing an egg.
3. Children boil water and get egg whites ready.
4. People make food in a kitchen.
5. A group of people are making food in a kitchen.

Why Language and Vision

https://datarelease.blob.core.windows.net/tutorial/VQA2VLN2021/VLN_part1.pdf

Why Language and Vision

An AI system should perform well on both.
"Embodied AI"



👁️ **See**: perceive their environment through vision or other senses.

🎤 **Talk**: hold a natural language dialog grounded in their environment.

👂 **Listen**: understand and react to audio input anywhere in a scene.

🕹️ **Act**: navigate and interact with their environment to accomplish goals.

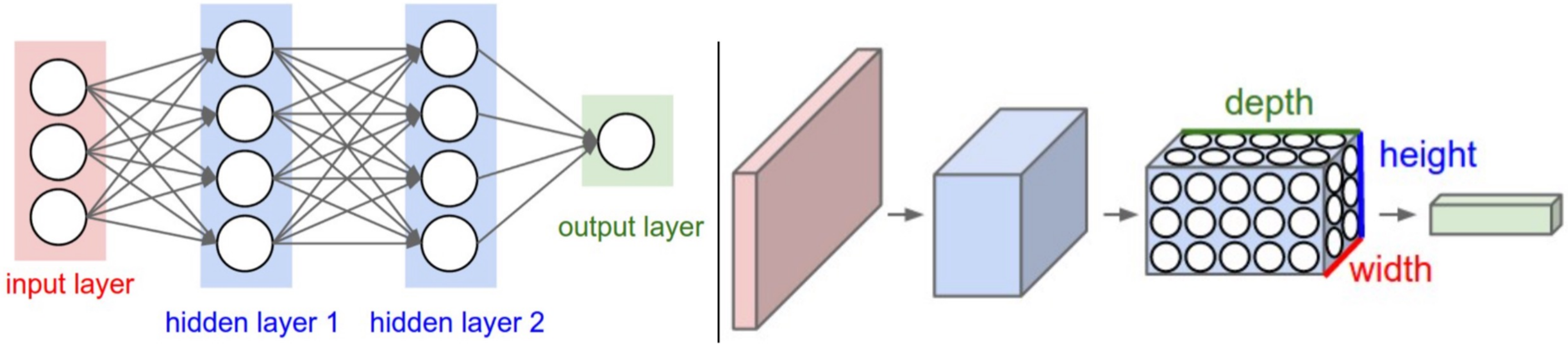
🤔 **Reason**: consider and plan for the long-term consequences of their actions.

Embodied AI is the field for solving AI problems for virtual robots that can move, see, speak, and interact in the virtual world and with other virtual robots — these simulated robot solutions are then transferred to real world robots.

--- Luis Bermudez, Overview of Embodied Artificial Intelligence

https://datarelease.blob.core.windows.net/tutorial/VQA2VLN2021/VLN_part1.pdf

Convolutional Neural Networks

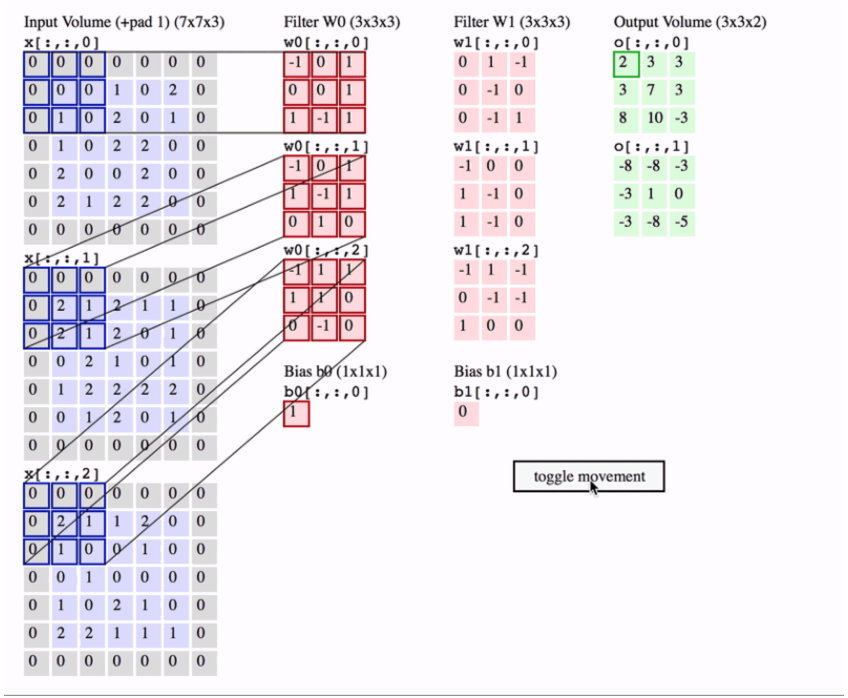


Left: A regular 3-layer Neural Network. Right: A ConvNet arranges its neurons in three dimensions (width, height, depth), as visualized in one of the layers. Every layer of a ConvNet transforms the 3D input volume to a 3D output volume of neuron activations. In this example, the red input layer holds the image, so its width and height would be the dimensions of the image, and the depth would be 3 (Red, Green, Blue channels).

In particular, unlike a regular Neural Network, the layers of a ConvNet have neurons arranged in 3 dimensions: **width, height, depth (equal to 3)**.

https://cs231n.github.io/convolutional-networks/

Convolutional Neural Networks



Input Volume (+pad 1) (7x7x3)

x[:,:,0]

0	0	0	0	0	0	0
0	0	0	1	0	2	0
0	1	0	2	0	1	0
0	1	0	2	2	0	0
0	2	0	0	2	0	0
0	2	1	2	2	0	0
0	0	0	0	0	0	0

x[:,:,1]

0	0	0	0	0	0	0
0	2	1	2	1	1	0
0	2	1	2	0	1	0
0	0	2	1	0	1	0
0	1	2	2	2	2	0
0	0	1	2	0	1	0
0	0	0	0	0	0	0

x[:,:,2]

0	0	0	0	0	0	0
0	2	1	1	2	0	0
0	1	0	0	1	0	0
0	0	1	0	0	0	0
0	1	0	2	1	0	0
0	2	2	1	1	1	0
0	0	0	0	0	0	0

Filter W0 (3x3x3)

w0[:,:,0]

-1	0	1
0	0	1
1	-1	1

w0[:,:,1]

-1	0	1
1	-1	1
0	1	0

w0[:,:,2]

-1	1	1
1	1	0
0	-1	0

Bias b0 (1x1x1)

b0[:,:,0]

1

Filter W1 (3x3x3)

w1[:,:,0]

0	1	-1
0	-1	0
0	-1	1

w1[:,:,1]

-1	0	0
1	-1	0
1	-1	0

w1[:,:,2]

-1	1	-1
0	-1	-1
1	0	0

Bias b1 (1x1x1)

b1[:,:,0]

0

Output Volume (3x3x2)

o[:,:,0]

2	3	3
3	7	3
8	10	-3

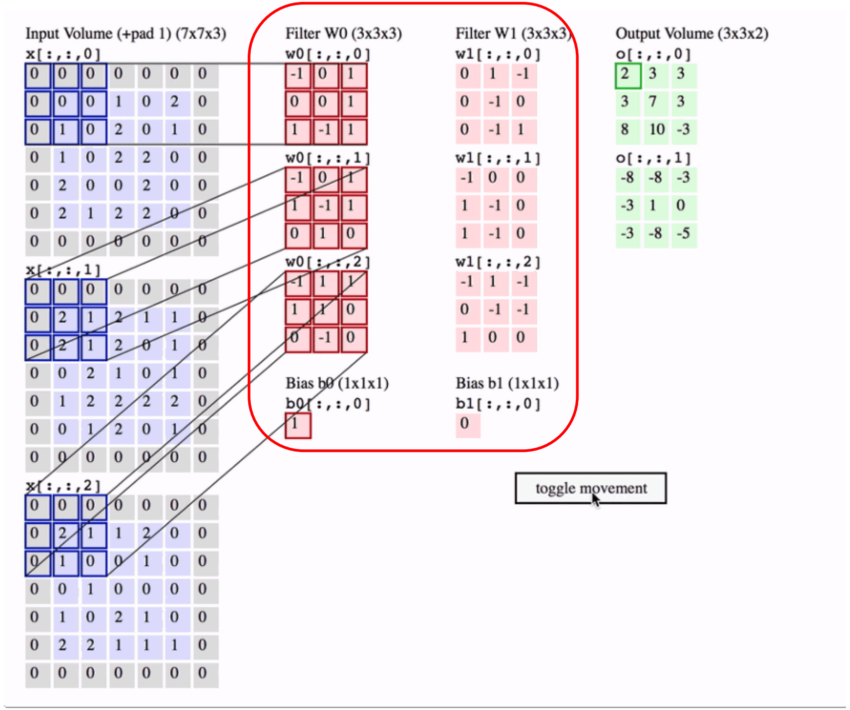
o[:,:,1]

-8	-8	-3
-3	1	0
-3	-8	-5

We use three main types of layers to build ConvNet architectures:
Convolutional Layer, **Pooling Layer**, and **Fully-Connected Layer**

https://cs231n.github.io/convolutional-networks/
https://towardsdatascience.com/convolutional-neural-network-in-natural-language-processing-96d67f91275c

Convolutional Neural Networks



Input Volume (+pad 1) (7x7x3)

x[:,:,0]

0	0	0	0	0	0	0
0	0	0	1	0	2	0
0	1	0	2	0	1	0
0	1	0	2	2	0	0
0	2	0	0	2	0	0
0	2	1	2	2	0	0
0	0	0	0	0	0	0

x[:,:,1]

0	0	0	0	0	0	0
0	2	1	2	1	1	0
0	2	1	2	0	1	0
0	0	2	1	0	1	0
0	1	2	2	2	2	0
0	0	1	2	0	1	0
0	0	0	0	0	0	0

x[:,:,2]

0	0	0	0	0	0	0
0	2	1	1	2	0	0
0	1	0	0	1	0	0
0	0	1	0	0	0	0
0	1	0	2	1	0	0
0	2	2	1	1	1	0
0	0	0	0	0	0	0

Filter W0 (3x3x3)

w0[:,:,0]

-1	0	1
0	0	1
1	-1	1

w0[:,:,1]

-1	0	1
1	-1	1
0	1	0

w0[:,:,2]

-1	1	1
1	1	0
0	-1	0

Bias b0 (1x1x1)

b0[:,:,0]

1

Filter W1 (3x3x3)

w1[:,:,0]

0	1	-1
0	-1	0
0	-1	1

w1[:,:,1]

-1	0	0
1	-1	0
1	-1	0

w1[:,:,2]

-1	1	-1
0	-1	-1
1	0	0

Bias b1 (1x1x1)

b1[:,:,0]

0

Output Volume (3x3x2)

o[:,:,0]

2	3	3
3	7	3
8	10	-3

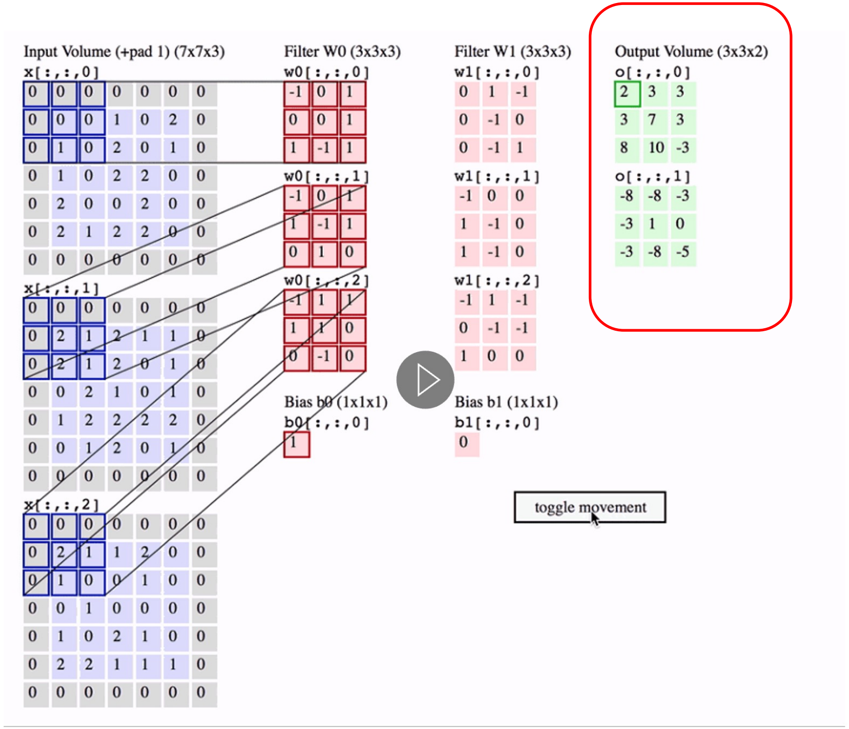
o[:,:,1]

-8	-8	-3
-3	1	0
-3	-8	-5

We have two filters of size 3×3, and they are applied with a **stride** of 2. Therefore, the output volume size has **spatial size** (5 - 3 + 2)/2 + 1 = 3

https://cs231n.github.io/convolutional-networks/
https://towardsdatascience.com/convolutional-neural-network-in-natural-language-processing-96d67f91275c

Convolutional Neural Networks



Input Volume (+pad 1) (7x7x3)

x[:,:,0]

0	0	0	0	0	0	0
0	0	0	1	0	2	0
0	1	0	2	0	1	0
0	1	0	2	2	0	0
0	2	0	0	2	0	0
0	2	1	2	2	0	0
0	0	0	0	0	0	0

x[:,:,1]

0	0	0	0	0	0	0
0	2	1	2	1	1	0
0	2	1	2	0	1	0
0	0	2	1	0	1	0
0	1	2	2	2	2	0
0	0	1	2	0	1	0
0	0	0	0	0	0	0

x[:,:,2]

0	0	0	0	0	0	0
0	2	1	1	2	0	0
0	1	0	0	1	0	0
0	0	1	0	0	0	0
0	1	0	2	1	0	0
0	2	2	1	1	1	0
0	0	0	0	0	0	0

Filter W0 (3x3x3)

w0[:,:,0]

-1	0	1
0	0	1
1	-1	1

w0[:,:,1]

-1	0	1
1	-1	1
0	1	0

w0[:,:,2]

-1	1	1
1	1	0
0	-1	0

Bias b0 (1x1x1)

b0[:,:,0]

1

Filter W1 (3x3x3)

w1[:,:,0]

0	1	-1
0	-1	0
0	-1	1

w1[:,:,1]

-1	0	0
1	-1	0
1	-1	0

w1[:,:,2]

-1	1	-1
0	-1	-1
1	0	0

Bias b1 (1x1x1)

b1[:,:,0]

0

Output Volume (3x3x2)

o[:,:,0]

2	3	3
3	7	3
8	10	-3

o[:,:,1]

-8	-8	-3
-3	1	0
-3	-8	-5

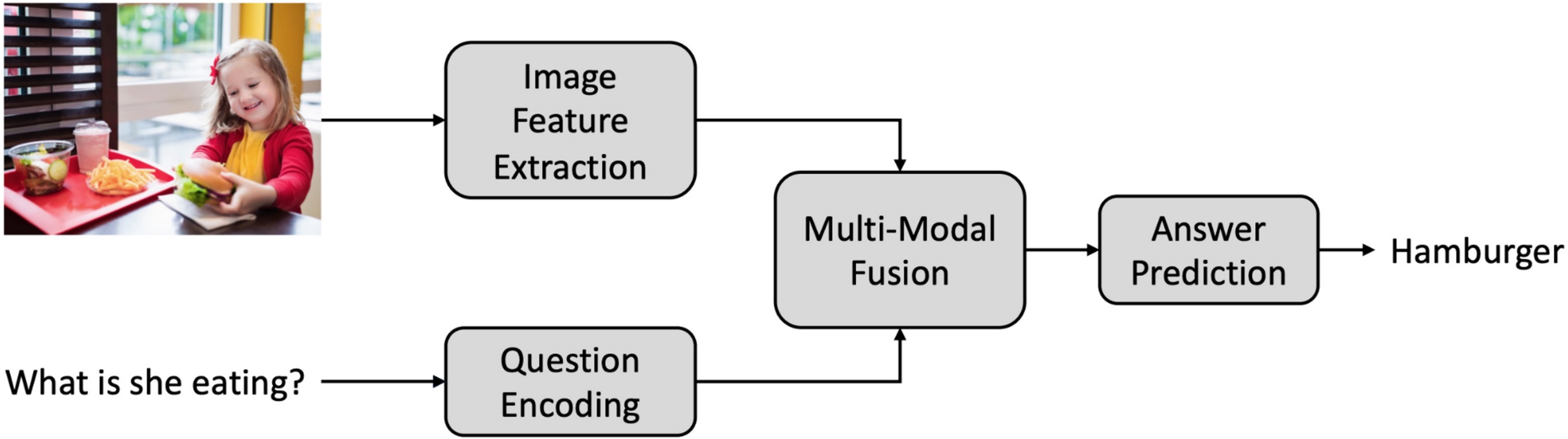
To get output activations (green): summing up, and then offsetting the result by the bias.

https://cs231n.github.io/convolutional-networks/
https://towardsdatascience.com/convolutional-neural-network-in-natural-language-processing-96d67f91275c

Convolutional Neural Networks

https://cs231n.github.io/convolutional-networks/

How a typical VQA system works



What is she eating?

Image Feature Extraction

Question Encoding

Multi-Modal Fusion

Answer Prediction

Hamburger

https://datarelease.blob.core.windows.net/tutorial/VQA2VLN2021/VLP_part1.pdf

MCAN: Deep Modular Co-Attention Network (Yu et al., 2019)

Winner of the VQA Challenge 2019

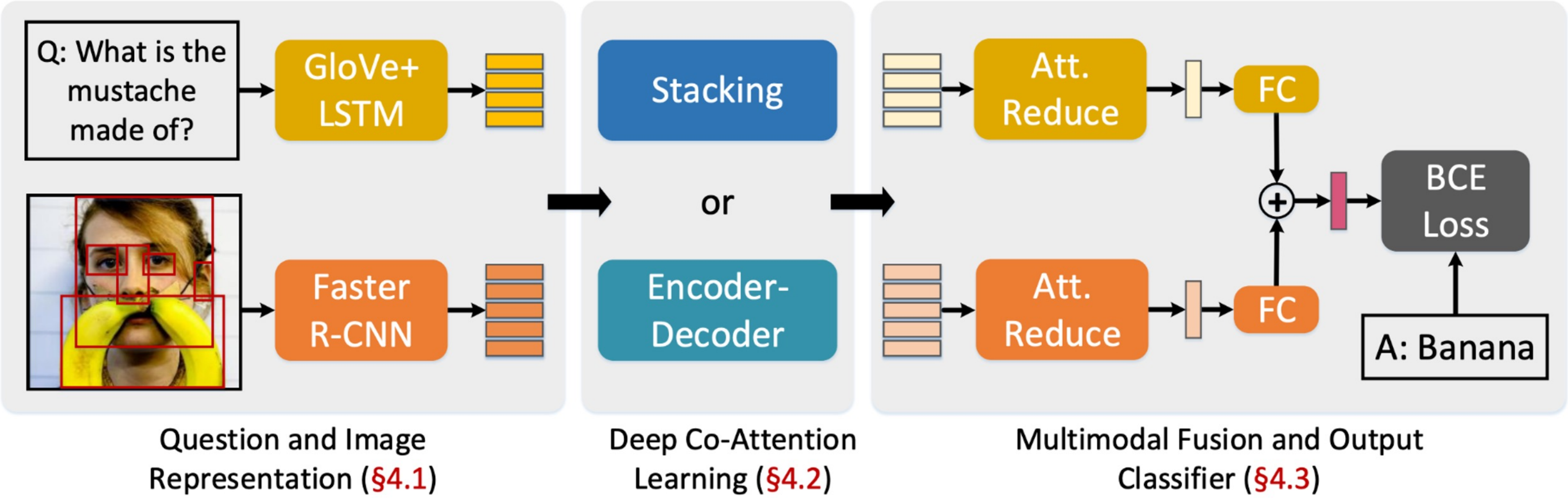
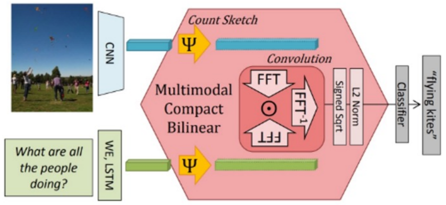

Q: What is the mustache made of?
GloVe+ LSTM
Faster R-CNN
Stacking
or
Encoder-Decoder
Att. Reduce
Att. Reduce
FC
FC
BCE Loss
A: Banana
Question and Image Representation (§4.1)
Deep Co-Attention Learning (§4.2)
Multimodal Fusion and Output Classifier (§4.3)

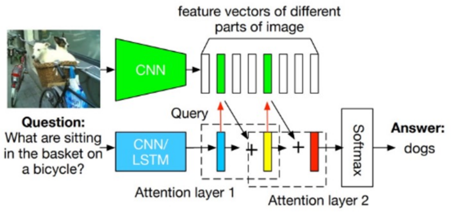
Figure 4: Overall flowchart of the deep Modular Co-Attention Networks (MCAN). In the Deep Co-attention Learning stage, we have two alternative strategies for deep co-attention learning, namely *stacking* and *encoder-decoder*.

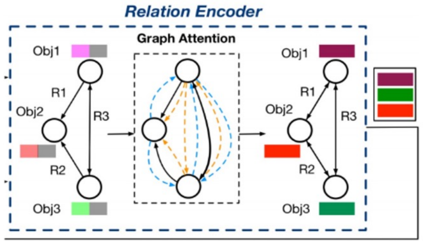
VL Research

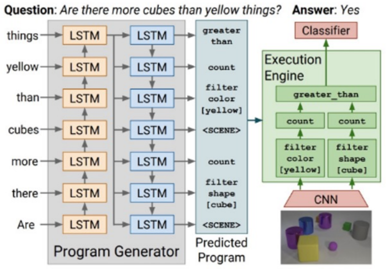
Methods before 2019

- Bilinear pooling
- All kinds of attention
- Incorporation of object relations
- Multi-step reasoning
- Neural modules




feature vectors of different parts of image
CNN
Query
Question:
What are sitting in the basket on a bicycle?
CNN/ LSTM
Attention layer 1
Attention layer 2
Softmax
Answer:
dogs



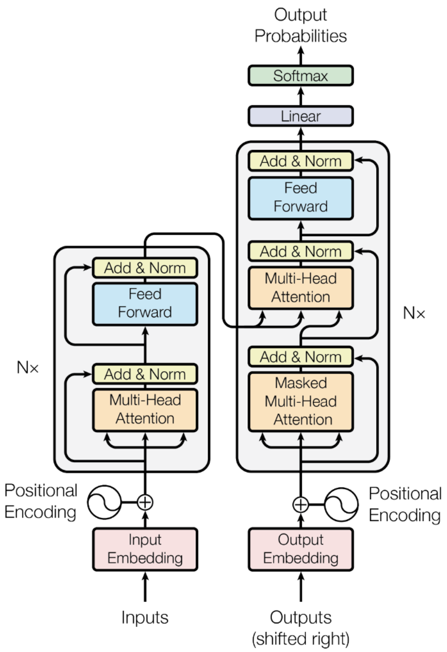


https://datarelease.blob.core.windows.net/tutorial/VQA2VLN2021/VLP_part1.pdf

VL Research

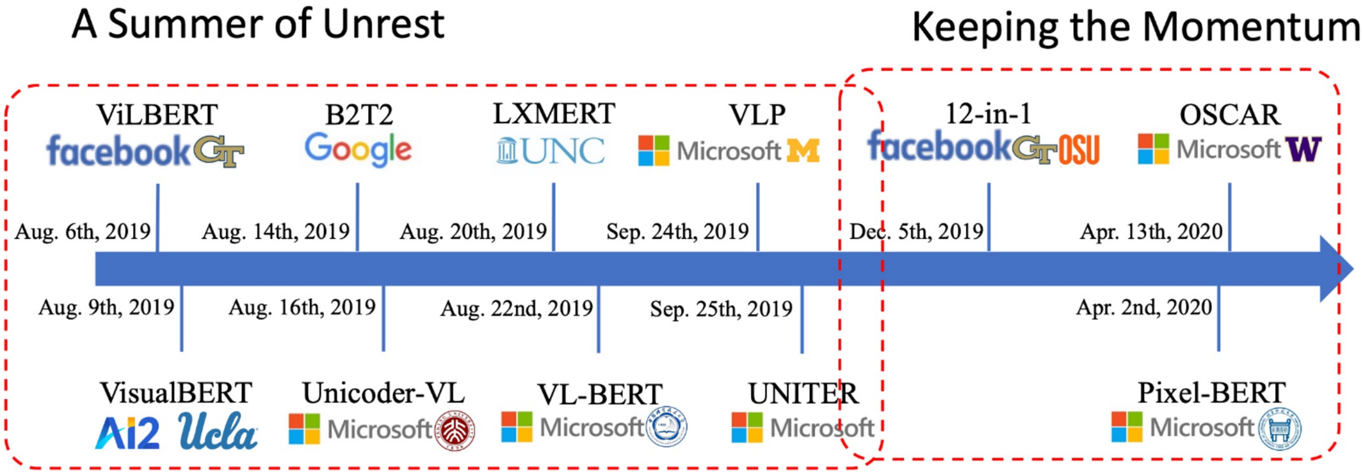
After 2019:

- Large-scale transformer-based self-supervised pre-training
- Transformer: first proposed for NLP, popularized by BERT and GPT-2/3, extended to image generation, vision-language pre-training, and now image classification



https://datarelease.blob.core.windows.net/tutorial/VQA2VLN2021/VLP_part1.pdf

Vision-and-Language Pretraining (VLP) models



Downstream Tasks

- VQA
- VCR
- NLVR2
- Visual Entailment
- Referring Expressions
- Image-Text Retrieval
- Image Captioning

https://datarelease.blob.core.windows.net/tutorial/VQA2VLN2021/VLP_part1.pdf

ViLBERT (Lu et al., 2019)

We generate image region features by extracting bounding boxes and their visual features from a pre-trained object detection network (see Sec. 3.1).

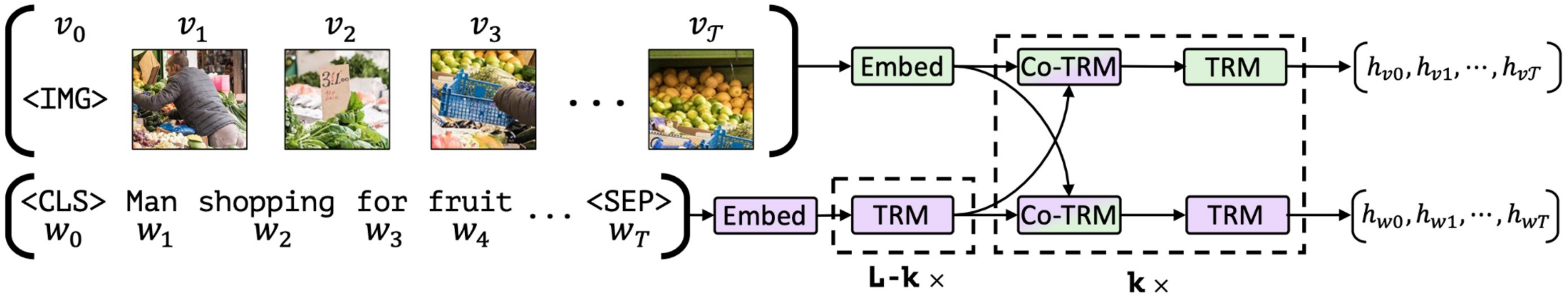
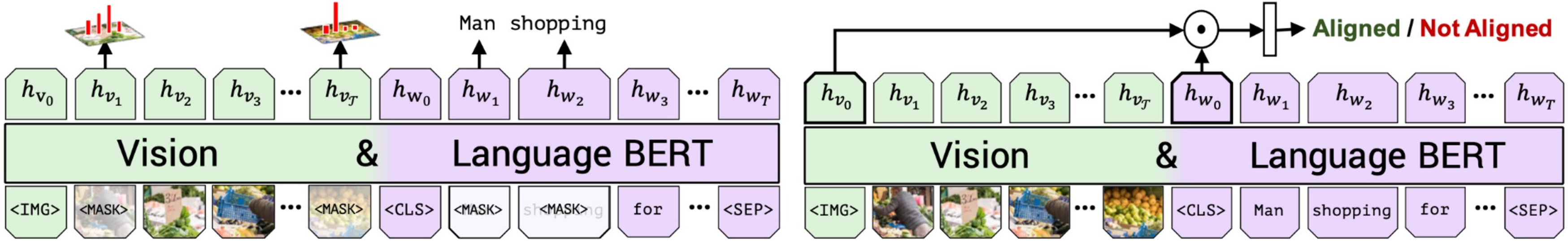


Figure 1: Our ViLBERT model consists of two parallel streams for visual (green) and linguistic (purple) processing that interact through novel co-attentional transformer layers. This structure allows for variable depths for each modality and enables sparse interaction through co-attention. Dashed boxes with multiplier subscripts denote repeated blocks of layers.

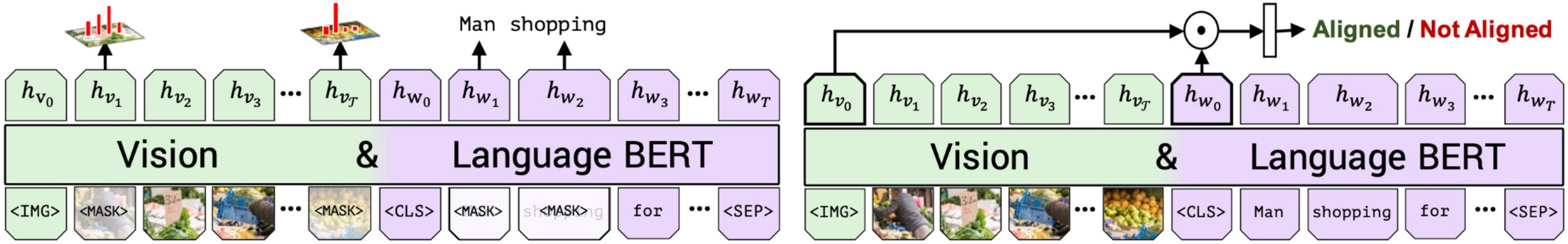
ViLBERT (Lu et al., 2019)

We generate image region features by extracting bounding boxes and their visual features from a pre-trained object detection network (see Sec. 3.1).


Man shopping
Aligned / Not Aligned
h_{v_0} h_{v_1} h_{v_2} h_{v_3} ... $h_{v_\mathcal{T}}$ h_{w_0} h_{w_1} h_{w_2} h_{w_3} ... h_{w_T}
Vision & Language BERT
<IMG> <MASK> ... <MASK> <CLS> <MASK> <MASK> for ... <SEP>
h_{v_0} h_{v_1} h_{v_2} h_{v_3} ... $h_{v_\mathcal{T}}$ h_{w_0} h_{w_1} h_{w_2} h_{w_3} ... h_{w_T}
Vision & Language BERT
<IMG> ... <CLS> Man shopping for ... <SEP>

ViLBERT (Lu et al., 2019)

We generate image region features by extracting bounding boxes and their visual features from a pre-trained object detection network (see Sec. 3.1).



(a) Masked multi-modal learning

(b) Multi-modal alignment prediction

Figure 3: We train ViLBERT on the Conceptual Captions [24] dataset under two training tasks to learn visual grounding. In masked multi-modal learning, the model must reconstruct image region categories or words for masked inputs given the observed inputs. In multi-modal alignment prediction, the model must predict whether or not the caption describes the image content.

VisualBERT (Li et al., 2019)

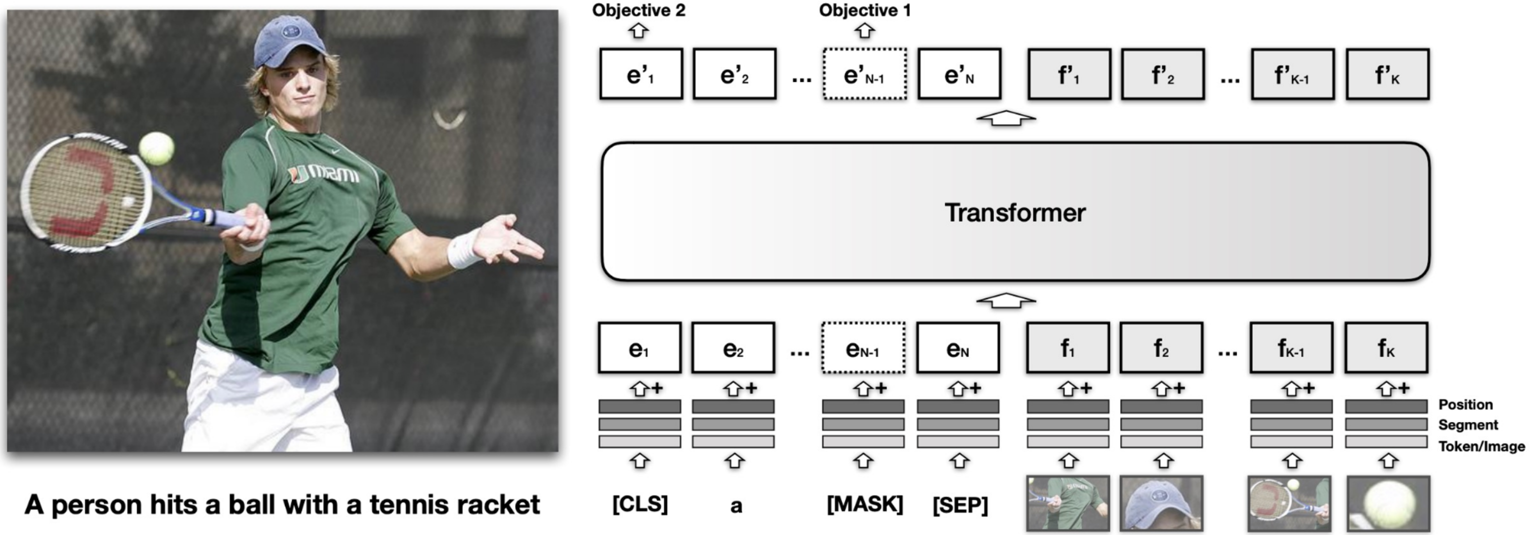
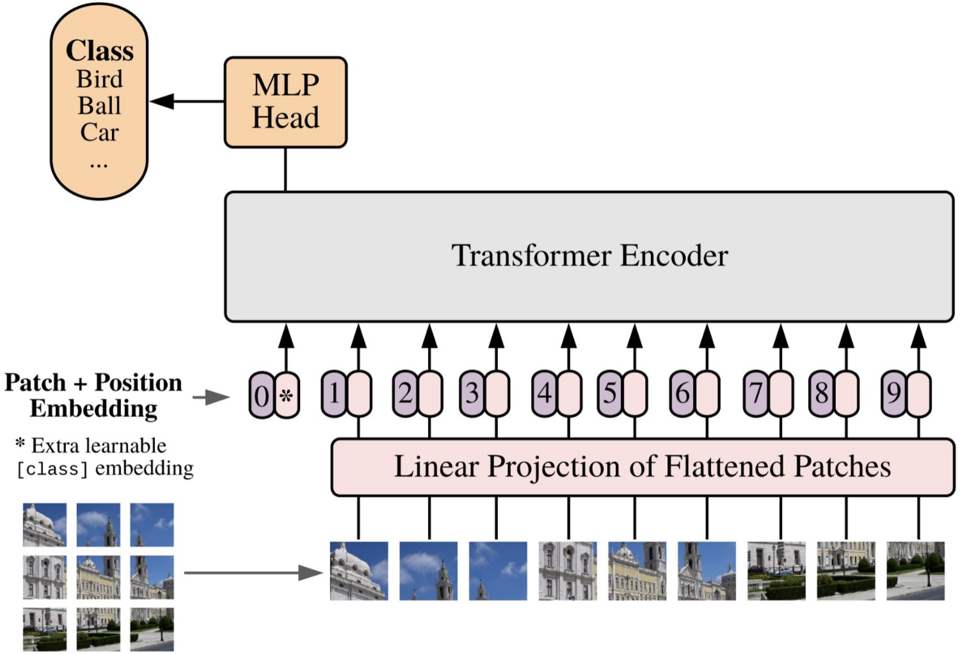

Objective 2
Objective 1
e'_1 e'_2 ... e'_{N-1} e'_N f'_1 f'_2 ... f'_{K-1} f'_K
Transformer
e_1 e_2 ... e_{N-1} e_N f_1 f_2 ... f_{K-1} f_K
Position
Segment
Token/Image
A person hits a ball with a tennis racket
[CLS]
a
[MASK]
[SEP]

Figure 2: The architecture of VisualBERT. Image regions and language are combined with a Transformer to allow the self-attention to discover implicit alignments between language and vision. It is pre-trained with a masked language modeling (Objective 1), and sentence-image prediction task (Objective 2), on caption data and then fine-tuned for different tasks. See §3.3 for more details.

ViT (Vision Transformer) (Dosovitskiy et al., 2020)

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

> "We show that this reliance on CNNs is not necessary and a pure transformer applied directly to sequences of image patches can perform very well on image classification tasks."


Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0
*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

BEiT: BERT Pre-Training of Image Transformers (Bao et al., 2021)

Pretrained Vision Transformer

masked image modeling

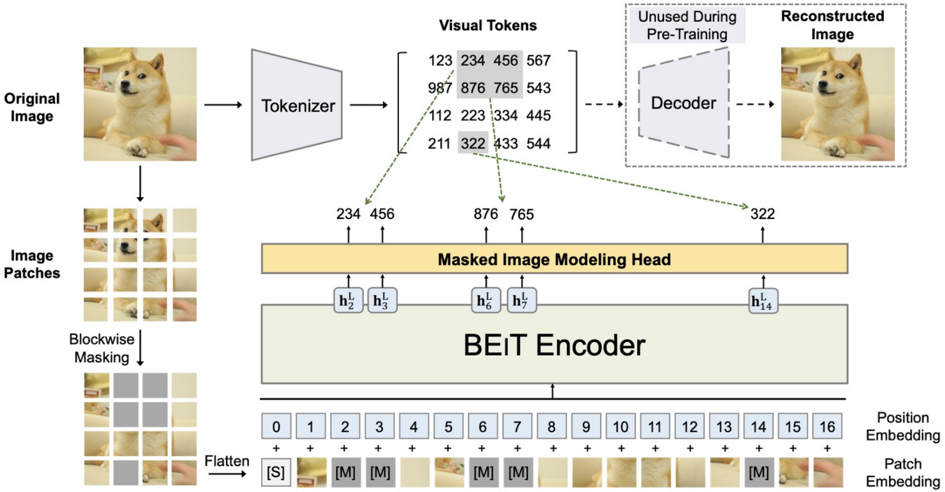


Figure 1: Overview of BEIT pre-training. Before pre-training, we learn an "image tokenizer" via autoencoding-style reconstruction, where an image is tokenized into discrete visual tokens according to the learned vocabulary. During pre-training, each image has two views, i.e., image patches, and visual tokens. We randomly mask some proportion of image patches (gray patches in the figure) and replace them with a special mask embedding [M]. Then the patches are fed to a backbone vision Transformer. The pre-training task aims at predicting the visual tokens of the *original* image based on the encoding vectors of the *corrupted* image.

Masked Autoencoders Are Scalable Vision Learners (He et al., 2021)

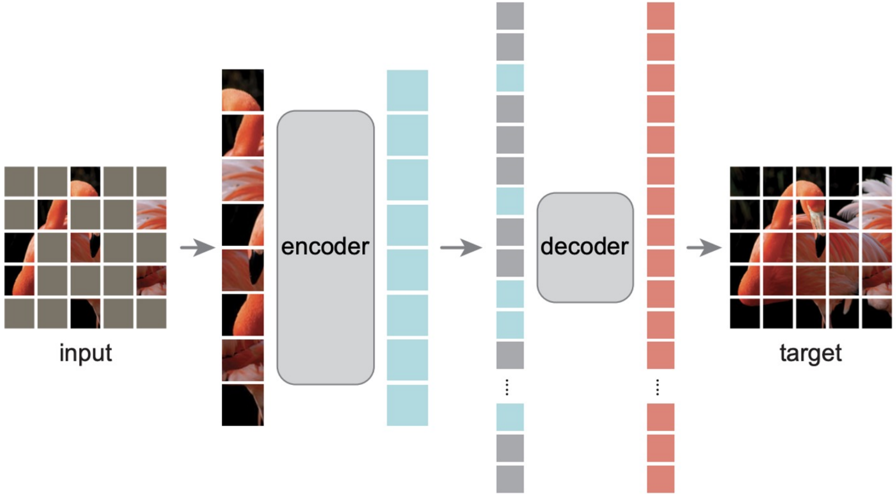
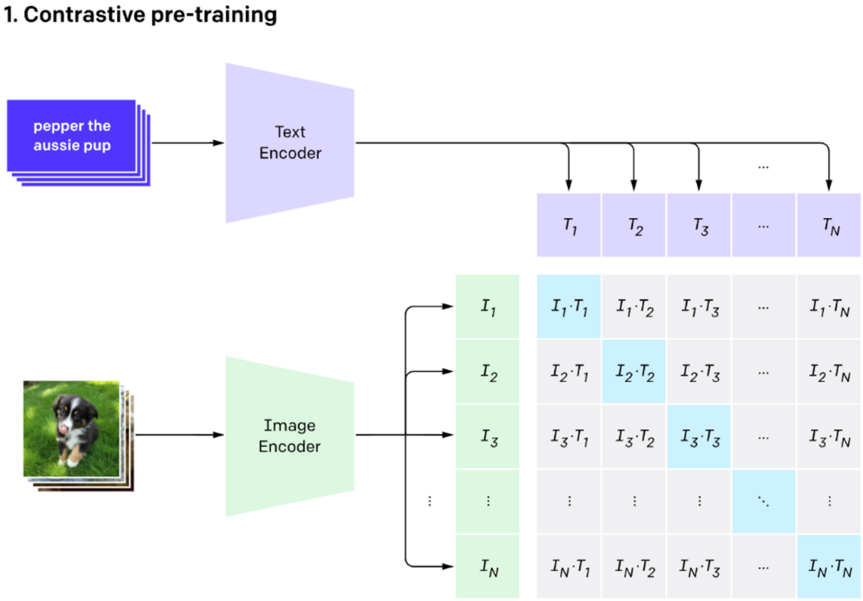


Figure 1. **Our MAE architecture**. During pre-training, a large random subset of image patches (*e.g.*, 75%) is masked out. The encoder is applied to the small subset of *visible patches*. Mask tokens are introduced *after* the encoder, and the full set of encoded patches and mask tokens is processed by a small decoder that reconstructs the original image in pixels. After pre-training, the decoder is discarded and the encoder is applied to uncorrupted images (full sets of patches) for recognition tasks.

CLIP: Contrastive language-image pretraining

CLIP pre-trains an image encoder and a text encoder to predict which images were paired with which texts in our dataset. We then use this behavior to turn CLIP into a zero-shot classifier. We convert all of a dataset's classes into captions such as "a photo of a *dog*" and predict the class of the caption CLIP estimates best pairs with a given image.



1. Contrastive pre-training

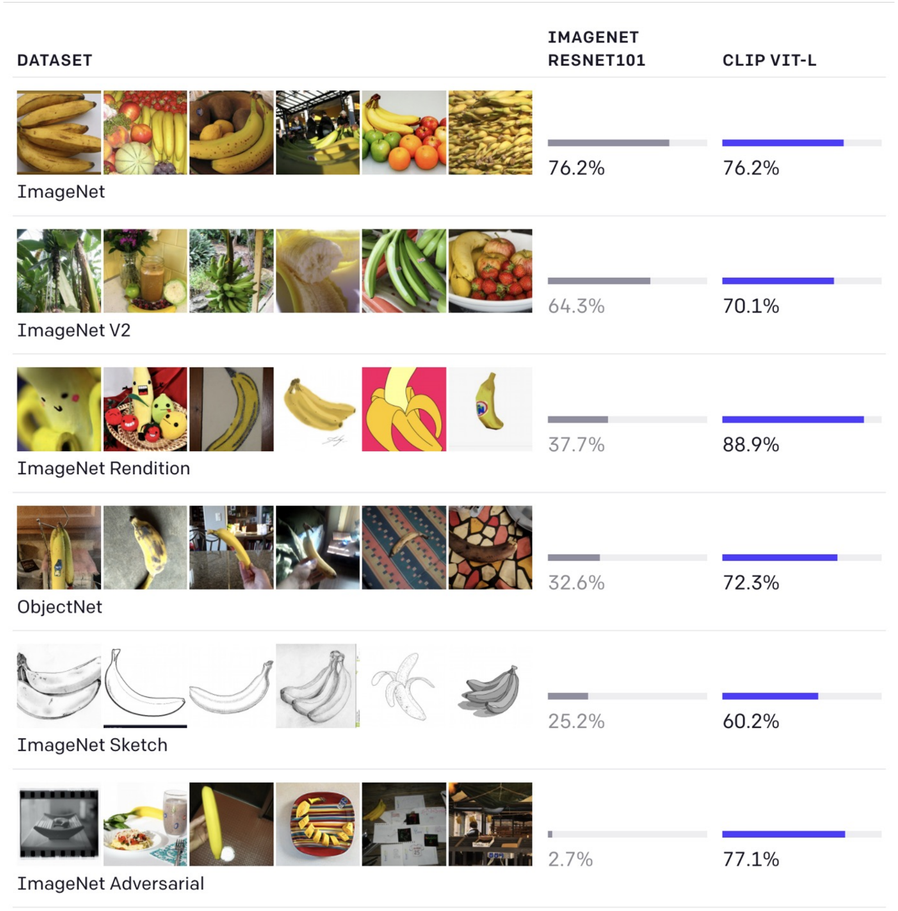
	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$	...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$	...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$	...	$I_3 \cdot T_N$
⋮	⋮	⋮	⋮	⋱	⋮
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$	...	$I_N \cdot T_N$

https://openai.com/blog/clip/

CLIP: Contrastive language-image pretraining

Although both models have the same accuracy on the ImageNet test set, **CLIP's performance is much more representative of how it will fare on datasets that measure accuracy in different, non-ImageNet settings**.

For instance, ObjectNet checks a model's ability to recognize objects in many different poses and with many different backgrounds inside homes while ImageNet Rendition and ImageNet Sketch check a model's ability to recognize more abstract depictions of objects.



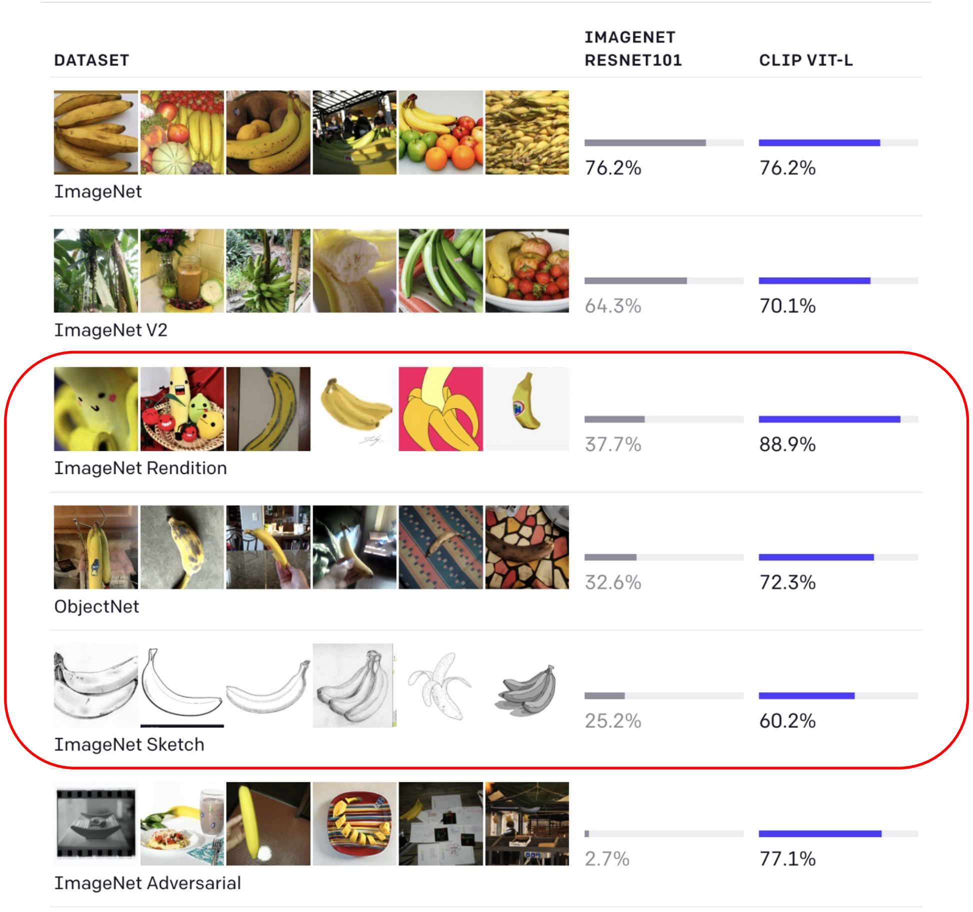
DATASET	IMAGENET RESNET101	CLIP VIT-L
ImageNet	76.2%	76.2%
ImageNet V2	64.3%	70.1%
ImageNet Rendition	37.7%	88.9%
ObjectNet	32.6%	72.3%
ImageNet Sketch	25.2%	60.2%
ImageNet Adversarial	2.7%	77.1%

https://openai.com/blog/clip/

CLIP: Contrastive language-image pretraining

Although both models have the same accuracy on the ImageNet test set, **CLIP's performance is much more representative of how it will fare on datasets that measure accuracy in different, non-ImageNet settings**.

For instance, **ObjectNet** checks a model's ability to recognize objects in many different poses and with many different backgrounds inside homes while **ImageNet Rendition and ImageNet Sketch** check a model's ability to recognize more abstract depictions of objects.

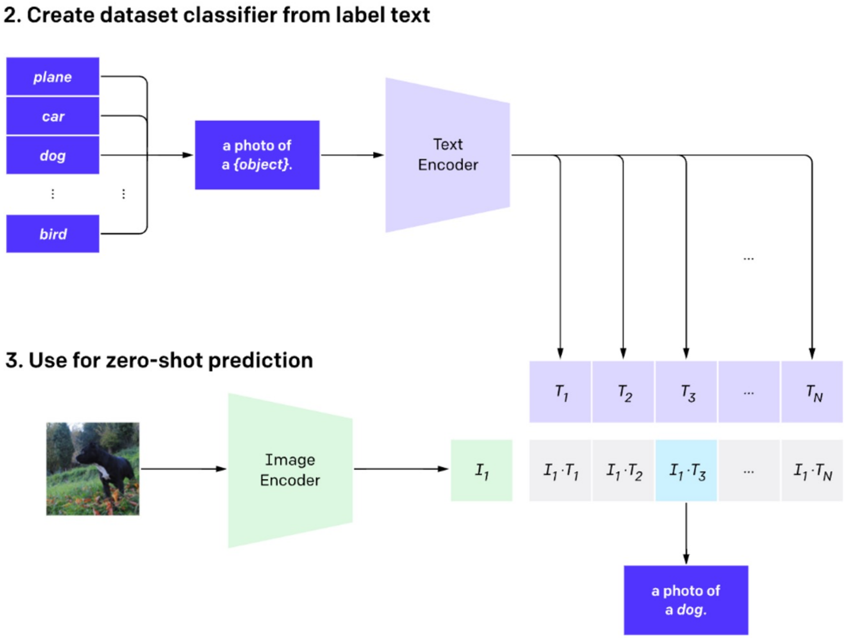


DATASET	IMAGENET RESNET101	CLIP VIT-L
ImageNet	76.2%	76.2%
ImageNet V2	64.3%	70.1%
ImageNet Rendition	37.7%	88.9%
ObjectNet	32.6%	72.3%
ImageNet Sketch	25.2%	60.2%
ImageNet Adversarial	2.7%	77.1%

https://openai.com/blog/clip/

CLIP for zero-shot learning

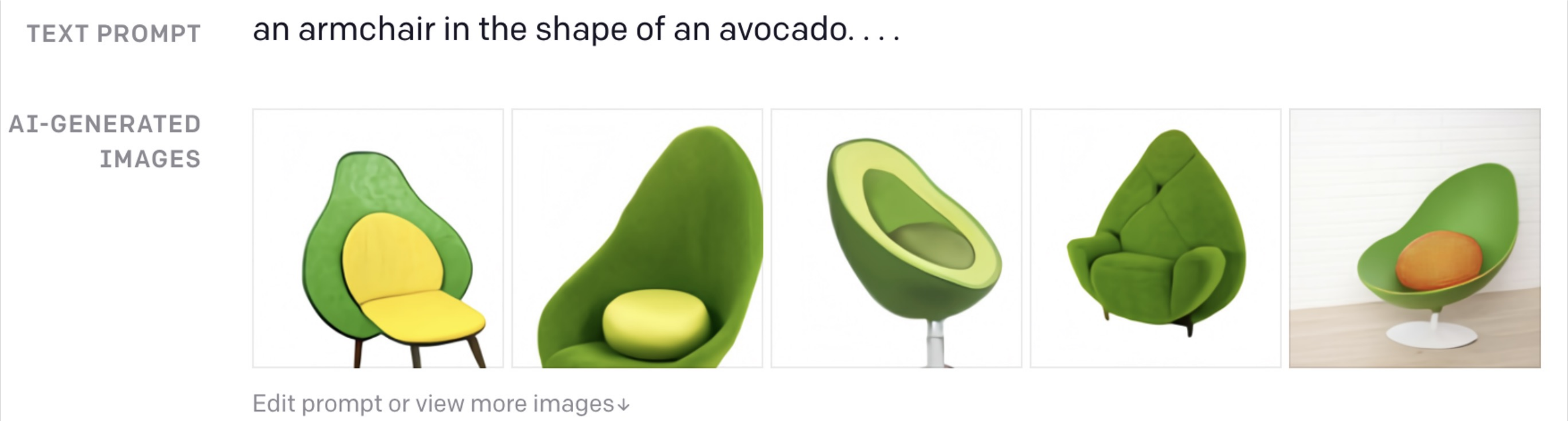
CLIP pre-trains an image encoder and a text encoder to predict which images were paired with which texts in our dataset. We then use this behavior to turn CLIP into a zero-shot classifier. We convert all of a dataset's classes into captions such as "a photo of a *dog*" and predict the class of the caption CLIP estimates best pairs with a given image.



https://openai.com/blog/clip/

DALL·E: Creating Images from Text

DALL·E is a 12-billion parameter version of GPT-3 trained to generate images from text descriptions, using a dataset of text–image pairs.



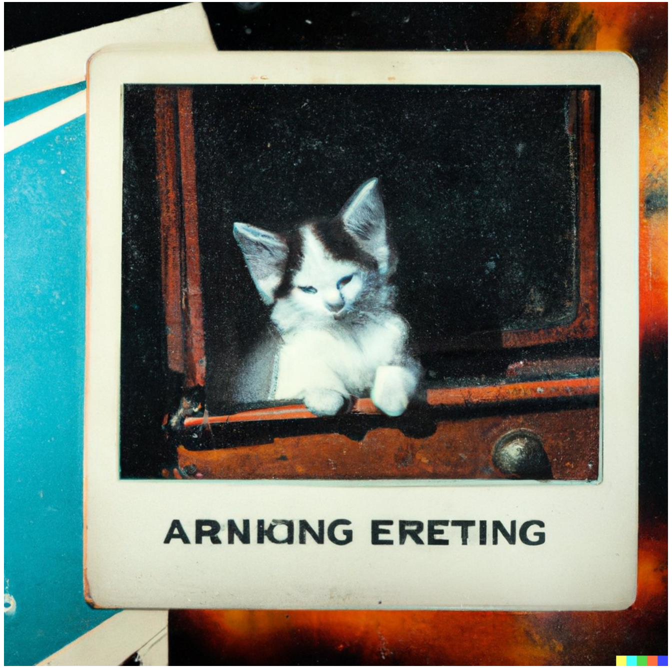
TEXT PROMPT an armchair in the shape of an avocado. . . .

https://openai.com/blog/dall-e/

(Some) Limitations of DALL·E

https://twitter.com/benjamin_hilton/status/1520032772072607747

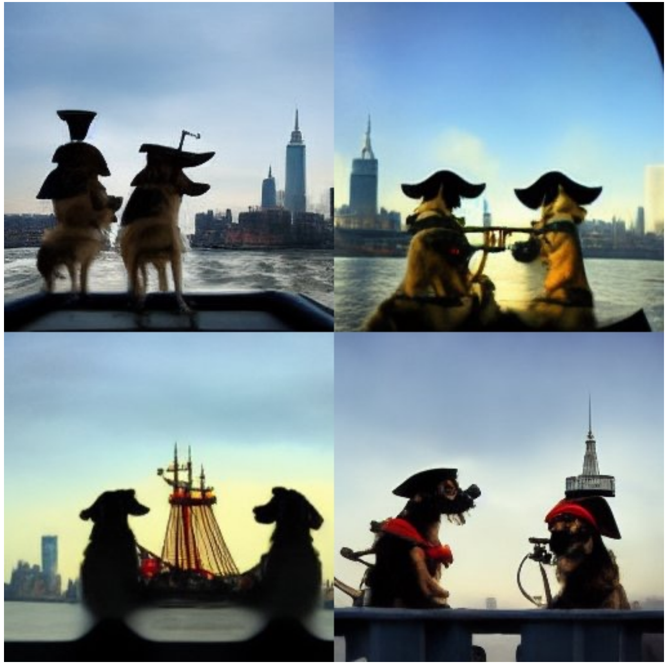
Sometimes it makes up letter-like things that aren't real letters.



(Some) Limitations of DALL·E

https://twitter.com/benjamin_hilton/status/1520032772072607747

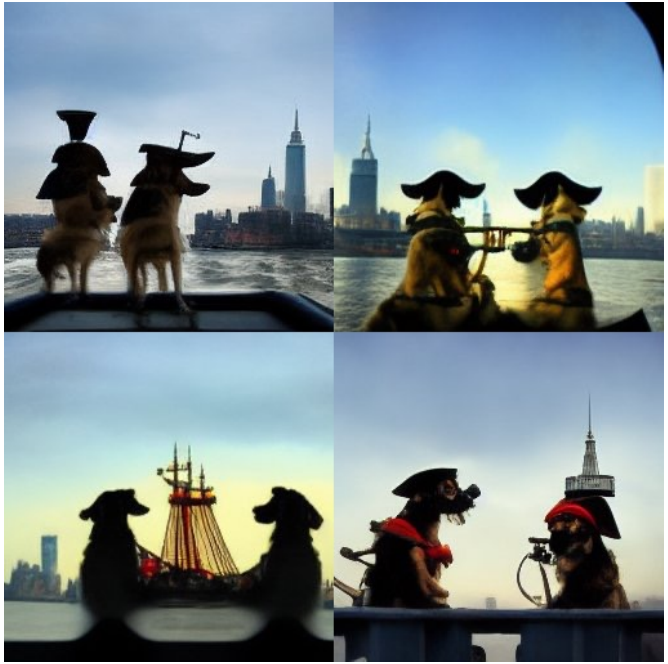
Two dogs dressed like roman soldiers on a pirate ship looking at New York City through a spyglass



(Some) Limitations of DALL·E

https://twitter.com/benjamin_hilton/status/1520032772072607747

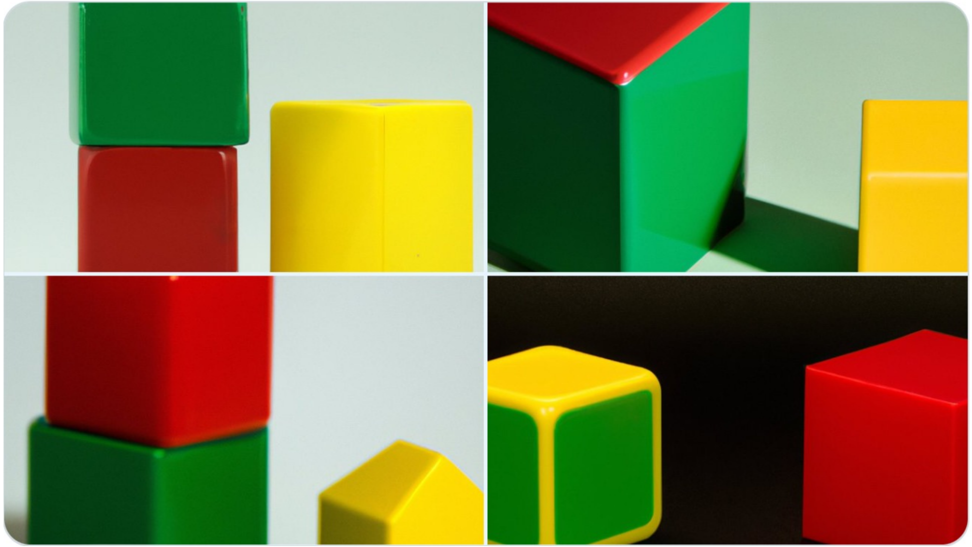
Two dogs dressed like roman soldiers on a pirate ship looking at New York City through a spyglass → DALL-E can't deal with lots of extras or very long descriptions.



(Some) Limitations of DALL·E

https://twitter.com/benjamin_hilton/status/1520032772072607747

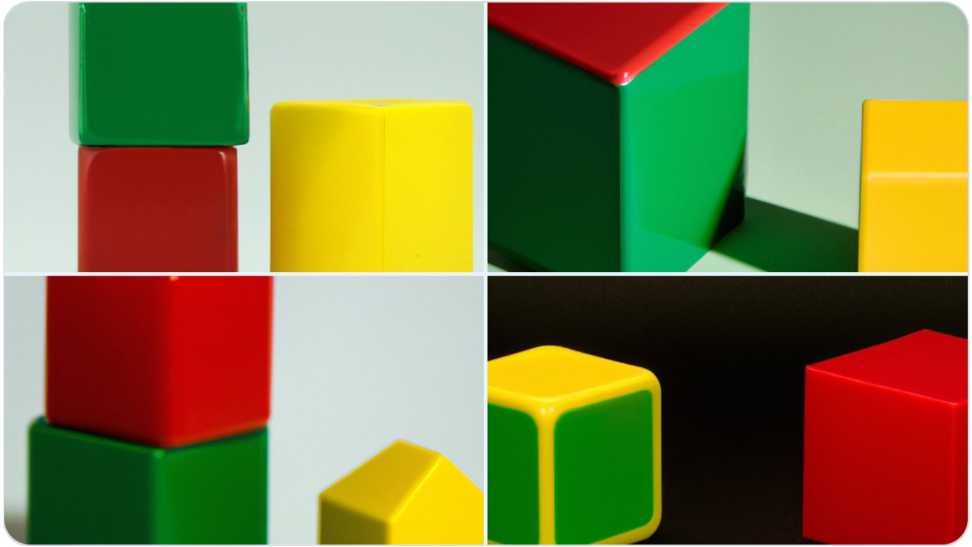
These are for the prompt:"A red cube, on top of a yellow cube, to the left of a green cube”?



(Some) Limitations of DALL·E

https://twitter.com/benjamin_hilton/status/1520032772072607747

These are for the prompt:"A red cube, on top of a yellow cube, to the left of a green cube”? ---→DALL-E isn't great at composition



Extras

VideoBERT (Sun et al., 2019)

VideoBERT: A Joint Model for Video and Language Representation Learning

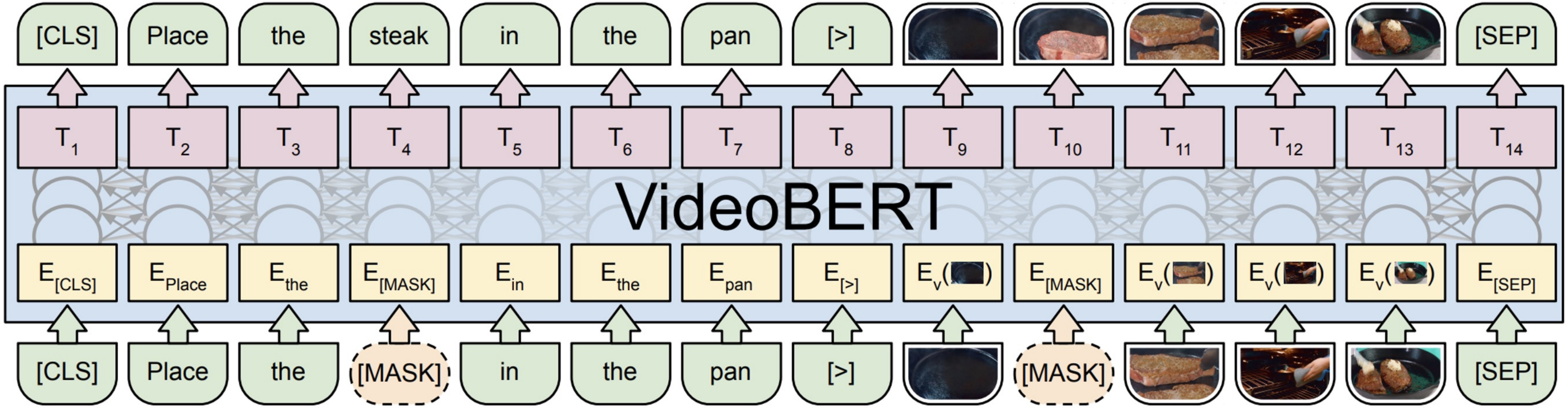
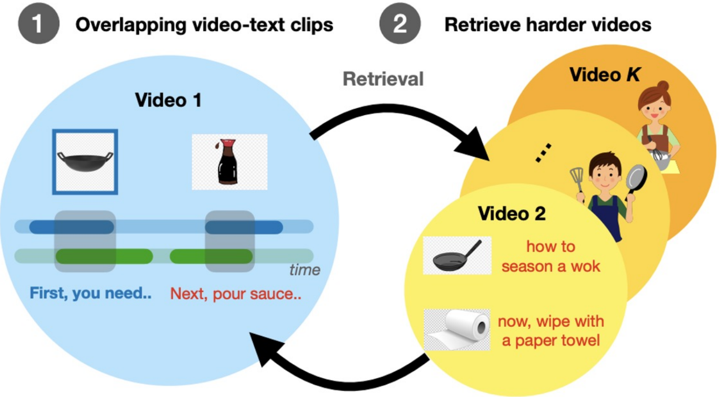


Figure 3: Illustration of VideoBERT in the context of a video and text masked token prediction, or *cloze*, task. This task also allows for training with text-only and video-only data, and VideoBERT can furthermore be trained using a linguistic-visual alignment classification objective (not shown here, see text for details).

VideoCLIP

VideoCLIP: Contrastive Pre-training for Zero-shot Video-Text Understanding



VideoCLIP: Contrastive learning with hard-retrieved negatives and overlapping positives for video-text pre-training.

Figure 1: VideoCLIP aims for zero-shot video understanding via learning fine-grained association between video and text in a transformer using a contrastive objective with two key novelties: (1) for *positive* pairs, we use video and text clips that are *loosely* temporarily overlapping instead of enforcing strict start/end timestamp overlap; (2) for *negative* pairs, we employ a retrieval based sampling technique that uses video clusters to form batches with mutually harder videos.