Intelligent Agents
Vision and Language

Prof. Dr. Ralf Möller
Universität zu Lübeck
Institut für Informationssysteme

Acknowledgements

vision & language

CS 685, Spring 2022
Advanced Natural Language Processing
http://people.cs.umass.edu/~miyyer/cs685/

Mohit Iyyer
College of Information and Computer Sciences
University of Massachusetts Amherst

some slides adapted from Vicente Ordonez, Fei-Fei Li, and Jacob Andreas

http://people.cs.umass.edu/~miyyer/cs685/

Image captioning

A red truck is parked
on a street lined with trees

http://people.cs.umass.edu/~miyyer/cs685/

Visual question answering

- Is this truck considered "vintage"?
- Does the road look new?
- What kind of tree is behind the truck?

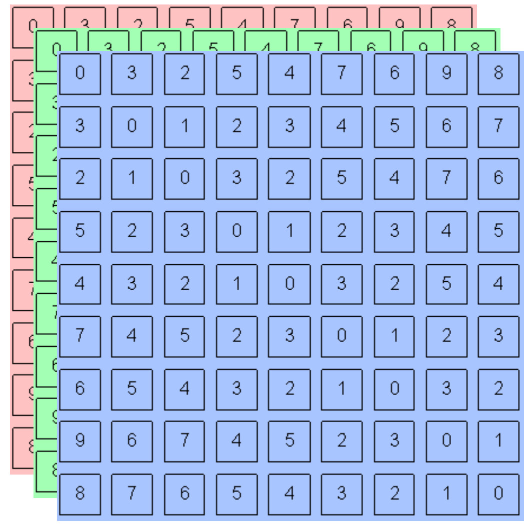
We’ve seen how to compute representations of words and sentences. What about images?

Grayscale images are matrices

0	3	2	5	4	7	6	9	8
3	0	1	2	3	4	5	6	7
2	1	0	3	2	5	4	7	6
5	2	3	0	1	2	3	4	5
4	3	2	1	0	3	2	5	4
7	4	5	2	3	0	1	2	3
6	5	4	3	2	1	0	3	2
9	6	7	4	5	2	3	0	1
8	7	6	5	4	3	2	1	0

What range of values can each pixel take?

Color images are tensors

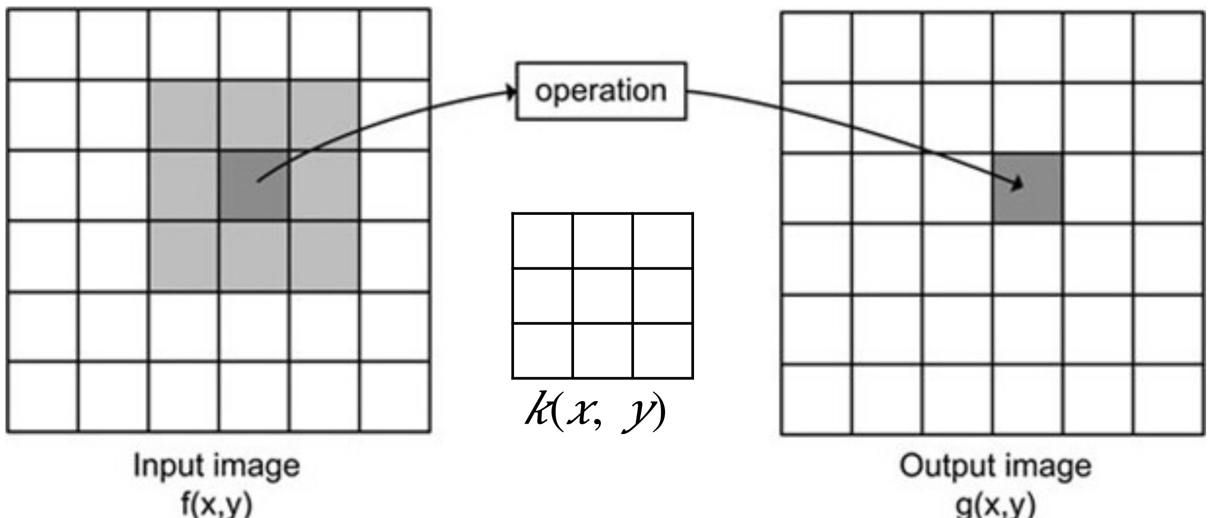


channel x height x width

Channels are usually RGB: Red, Green, and Blue

Other color spaces: HSV, HSL, LUV, XYZ, Lab, CMYK, etc

Convolution operator



$k(x, y)$

$$g(x, y) = \sum_{v} \sum_{u} k(u, v) f(x-u, y-v)$$

Image Credits: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video- and-image-processing-part-1/

(Filter, Kernel)

Input image * Weights → Output image

Input image:

4	5	7	6	6
3	2	8	0	7
6	7	7	1	5
3	0	1	1	1
4	3	2	1	7

*

Weights:

0	0	0
1	0	1
0	0	0

→

Output image:

	11	2	15	
	13	8	12	
	?			

Demo:
http://setosa.io/ev/image-kernels/

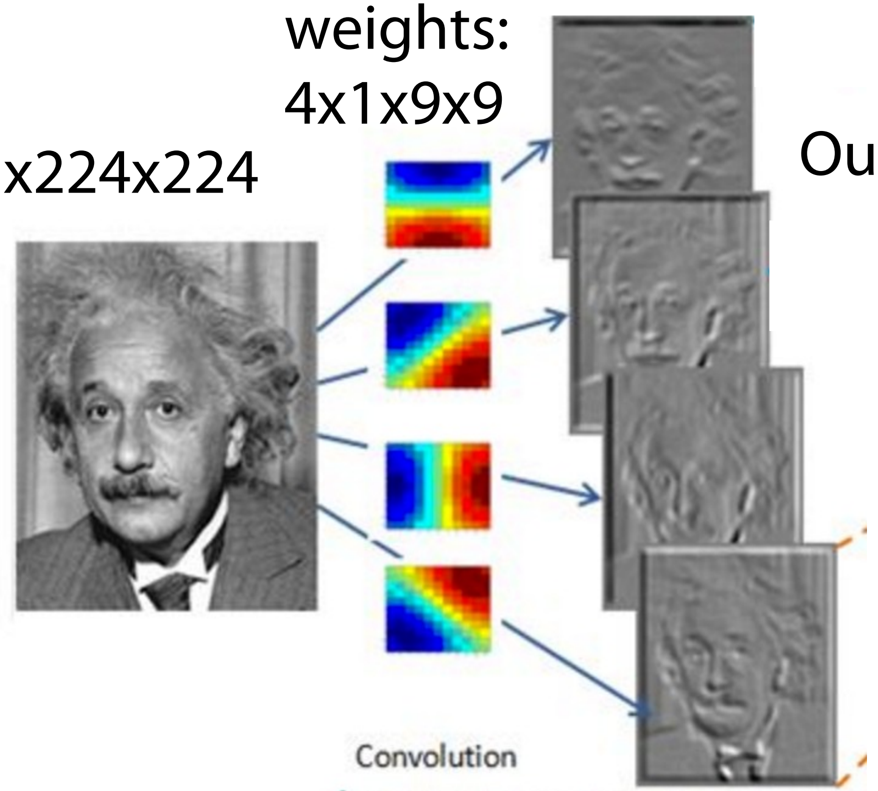
Convolutional Layer (with 4 filters)

weights:
4x1x9x9

Input: 1x224x224

Output: 4x224x224

if zero padding,
and stride = 1



http://people.cs.umass.edu/~miyyer/cs685/

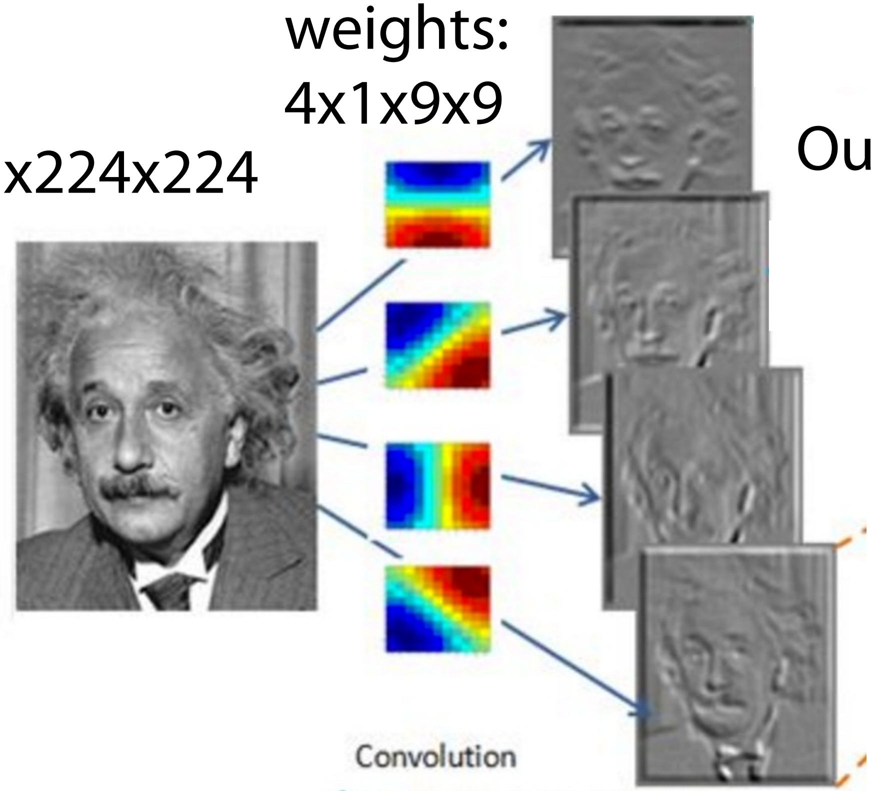
Convolutional Layer (with 4 filters)

weights:
4x1x9x9

Input: 1x224x224

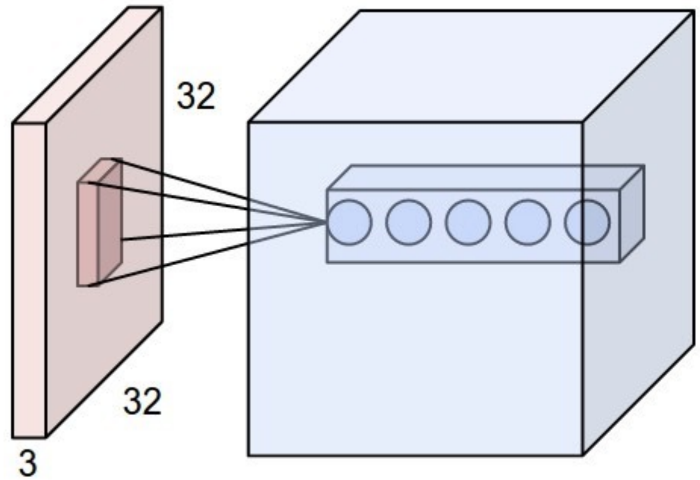
Output: 4x112x112

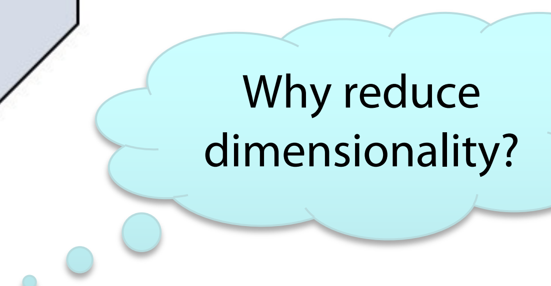
if zero padding,
but stride = 2



Pooling layers to reduce dimensionality

Convolutional Layers: slide a set of small filters over the image





Pooling Layers: reduce dimensionality of representation

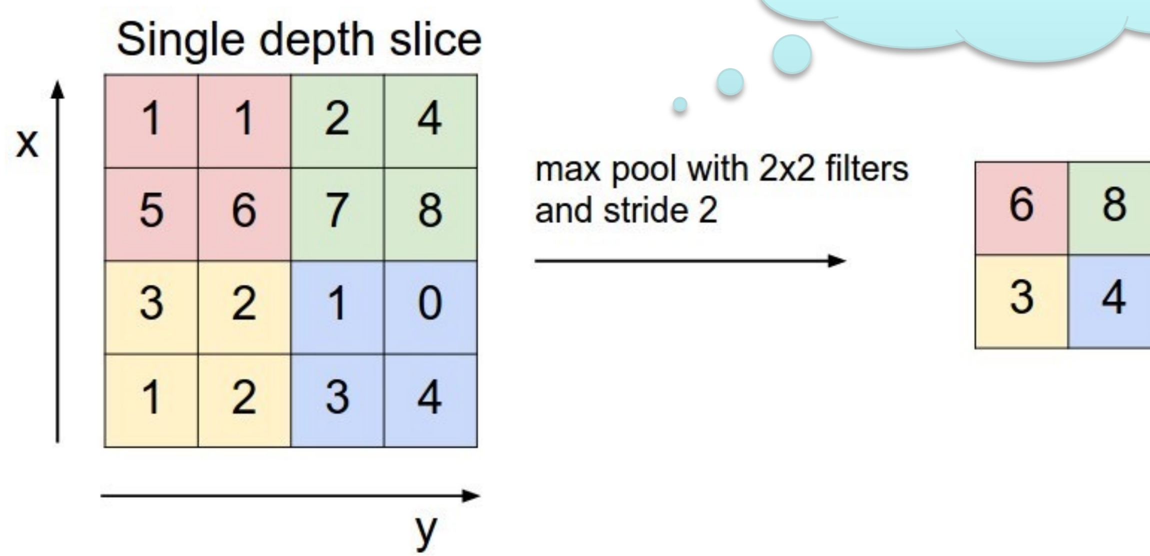


image: https://cs231n.github.io/convolutional-networks/

Alexnet

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

The paper that started the deep learning revolution!

http://people.cs.umass.edu/~miyyer/cs685/

Image classification

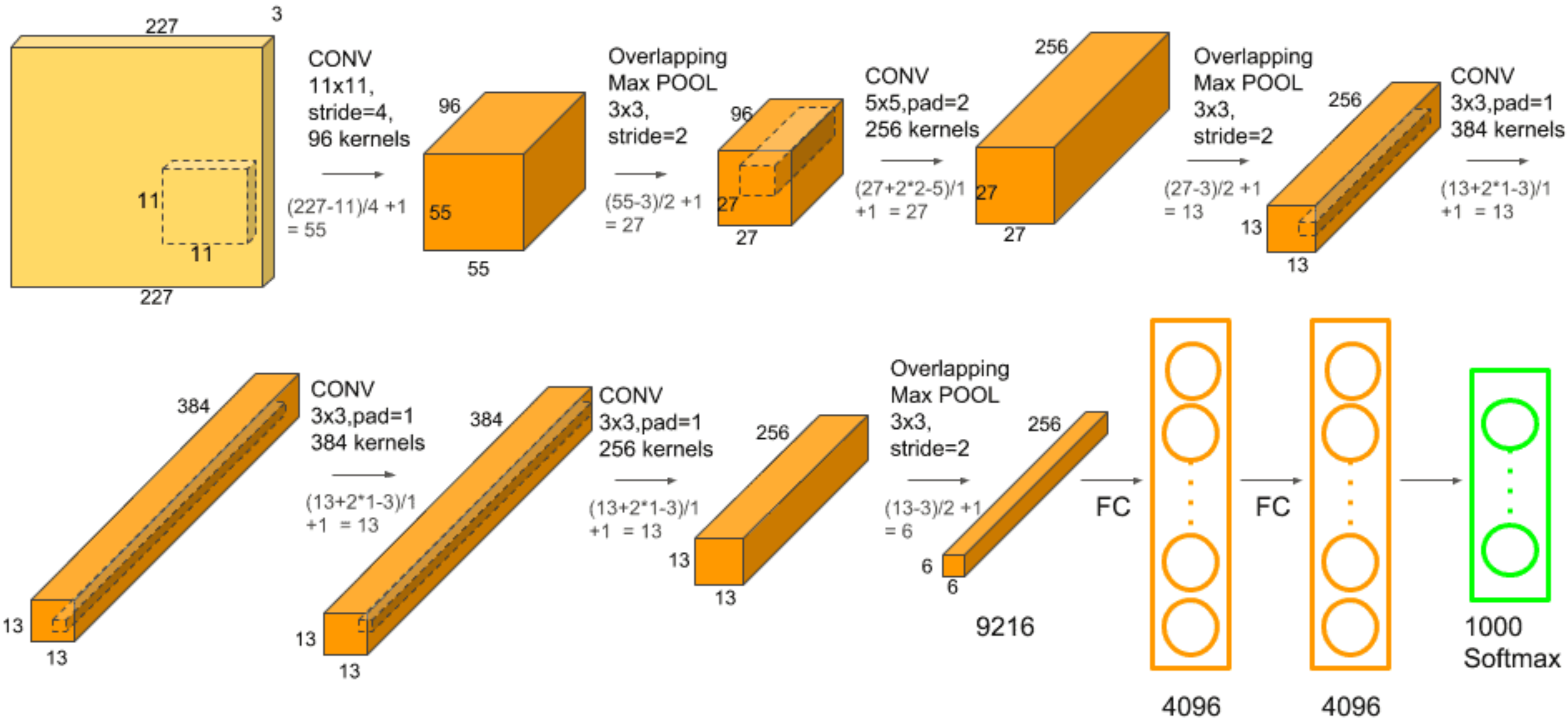
Classify an image into 1000 possible classes:

e.g. Abyssinian cat, Bulldog, French Terrier, Cormorant, Chickadee, Red fox, banjo, barbell, hourglass, knot, maze, viaduct, etc.

cat, tabby cat (0.71)
Egyptian cat (0.22)
red fox (0.11)
…..

Train on ImageNet challenge dataset, ~1.2 million images

Alexnet

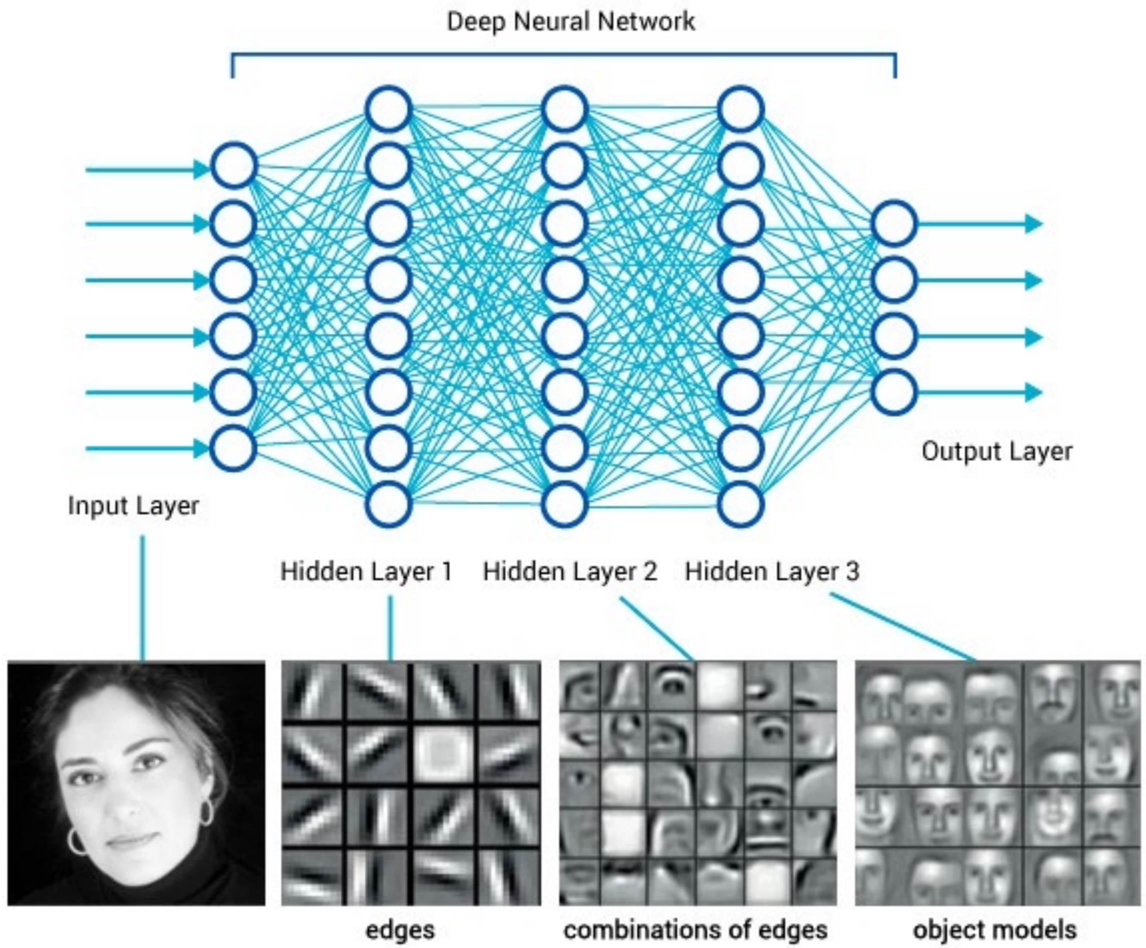


- Initially vectors of 227*227*3 = 154 587 features).
- Represented as a vector of 4096 features

- The two fully connected and softmax layers are similar to a multi layer perception and could actually be replaced by other kinds of classifiers such as Random Forests or SVMs. However they are really important for the training phase of the neural net.

https://learnopencv.com/understanding-alexnet/

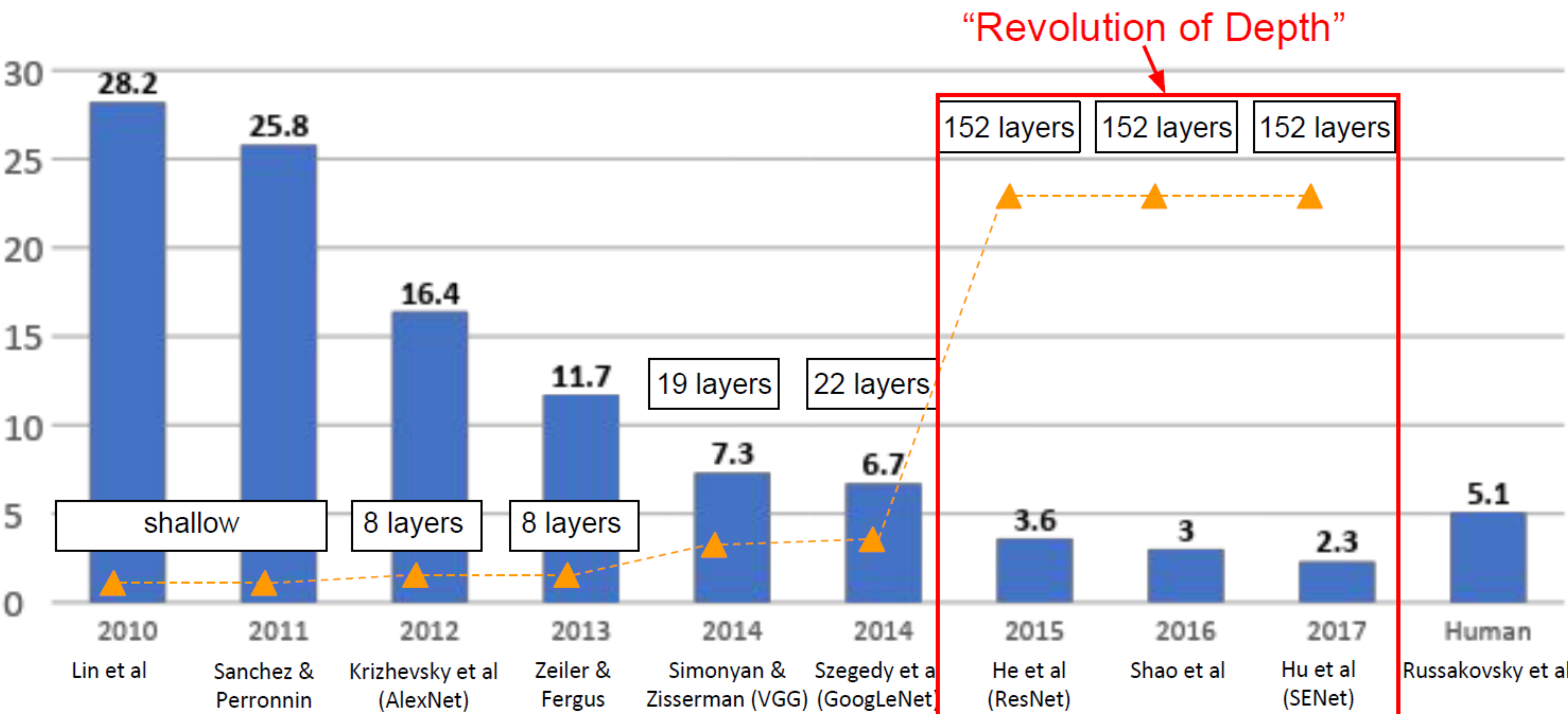
What is happening?



https://www.saagie.com/fr/blog/object-detection-part1

http://people.cs.umass.edu/~miyyer/cs685/

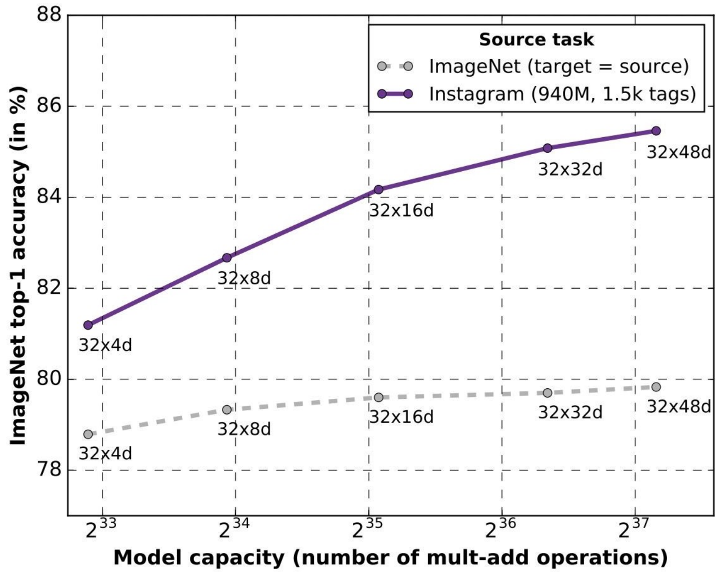
Revolution of depth



He et al.(2015), Deep Residual Learning for Image Recognition

ImageNet pretraining -> Instagram pretraining

Bigger models are saturated on ImageNet, but with more data bigger models do better



Biggest network was pretrained on 3.5B Instagram images

Trained on 336 GPUs for 22 days

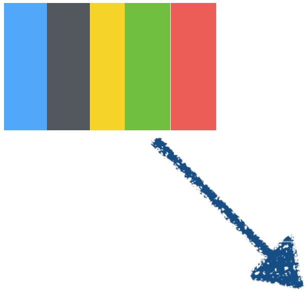
Mahajan et al, "Exploring the Limits of Weakly Supervised Pretraining", arXiv 2018

At the end of the day, …

… we generate a fixed size vector from an image and run a classifier over it

CNN (

) =



softmax: predict ‘truck’

Key insight

This vector is useful for many more tasks than just image classification! We can use it for *transfer learning*

Simple visual QA

- i := *CNN*(image) → use an existing network trained for image classification and freeze weights
- q := *BERT*(question) → learn weights
- Answer = softmax(linear([i;q]))

Why isn't this a real cool way of doing visual QA?

http://people.cs.umass.edu/~miyyer/cs685/

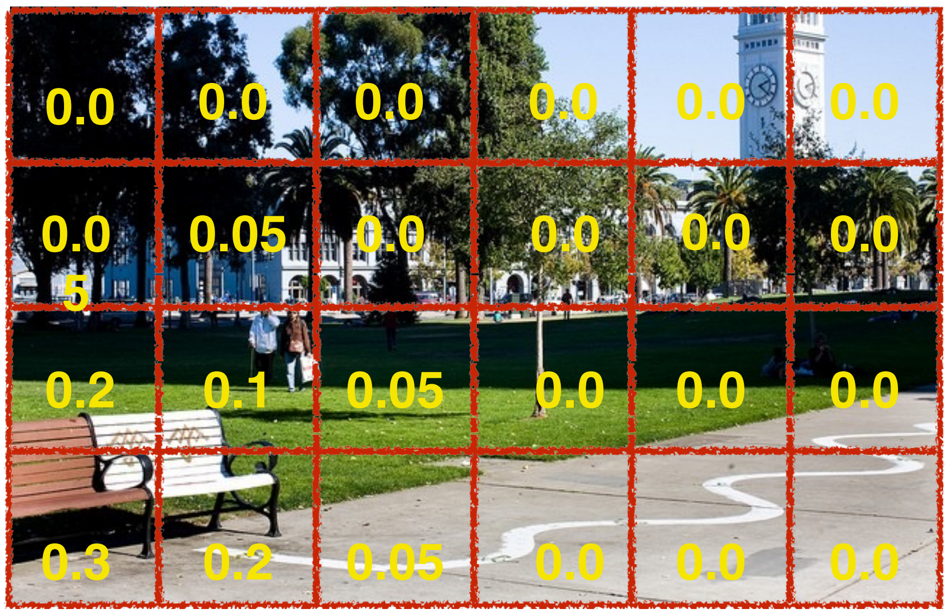
Visual attention

Use the question representation *q* to determine where in the image to look

How many benches are shown?

Attention over final convolutional layer in network: 196 boxes, captures color and positional information

softmax:
predict answer

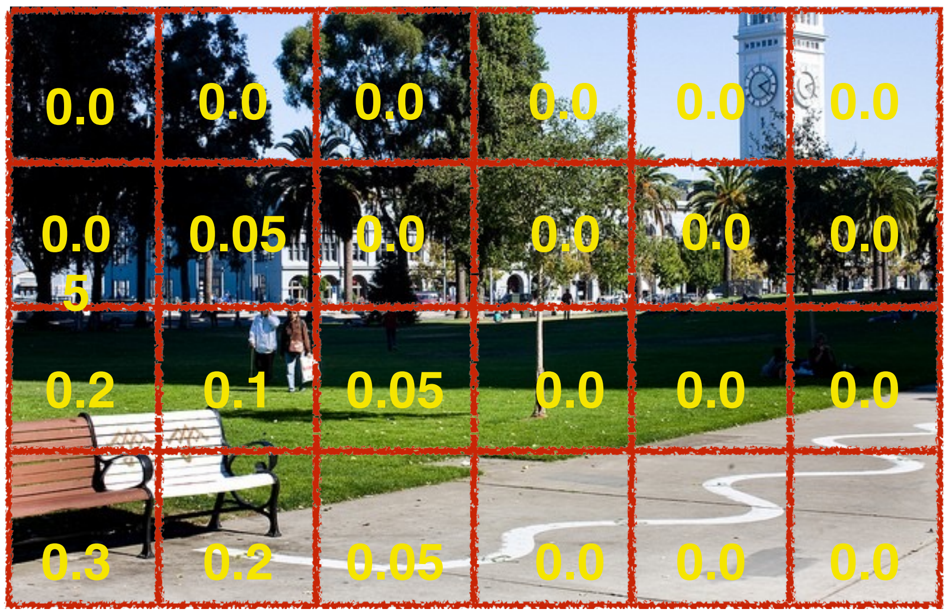


How many benches are shown?

http://people.cs.umass.edu/~miyyer/cs685/

Attention over final convolutional layer in network: 196 boxes, captures color and positional information

softmax:
predict answer



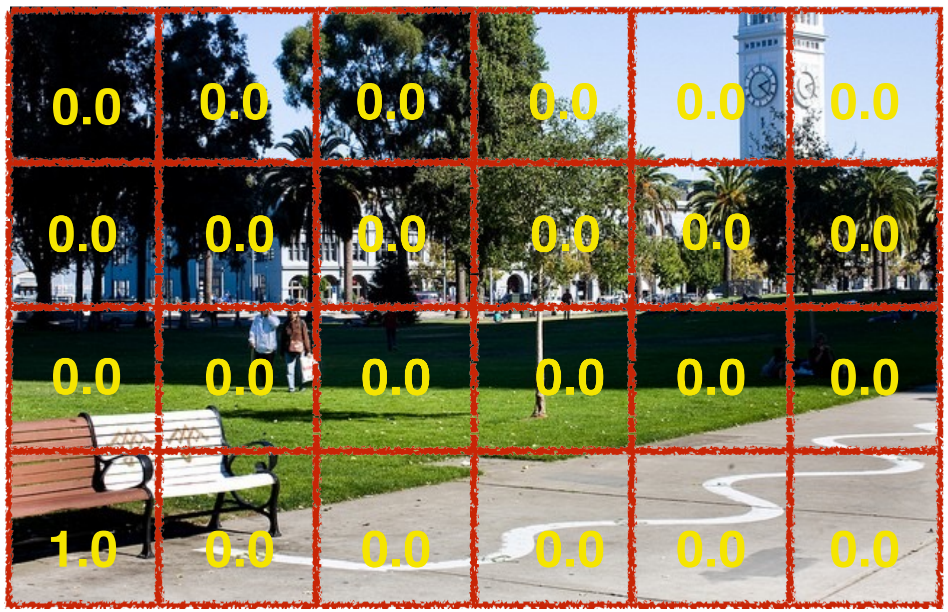
How can we compute these attention scores?

How many benches are shown?

Hard Attention

softmax:
predict answer

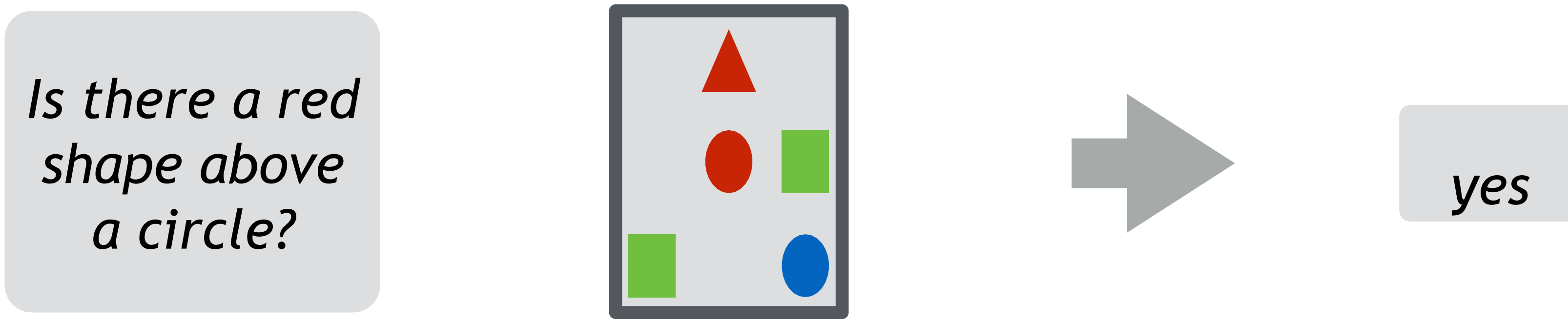
Attention over final convolutional layer in network: 196 boxes, captures color and positional information



We can use *reinforcement learning* to focus on just one box

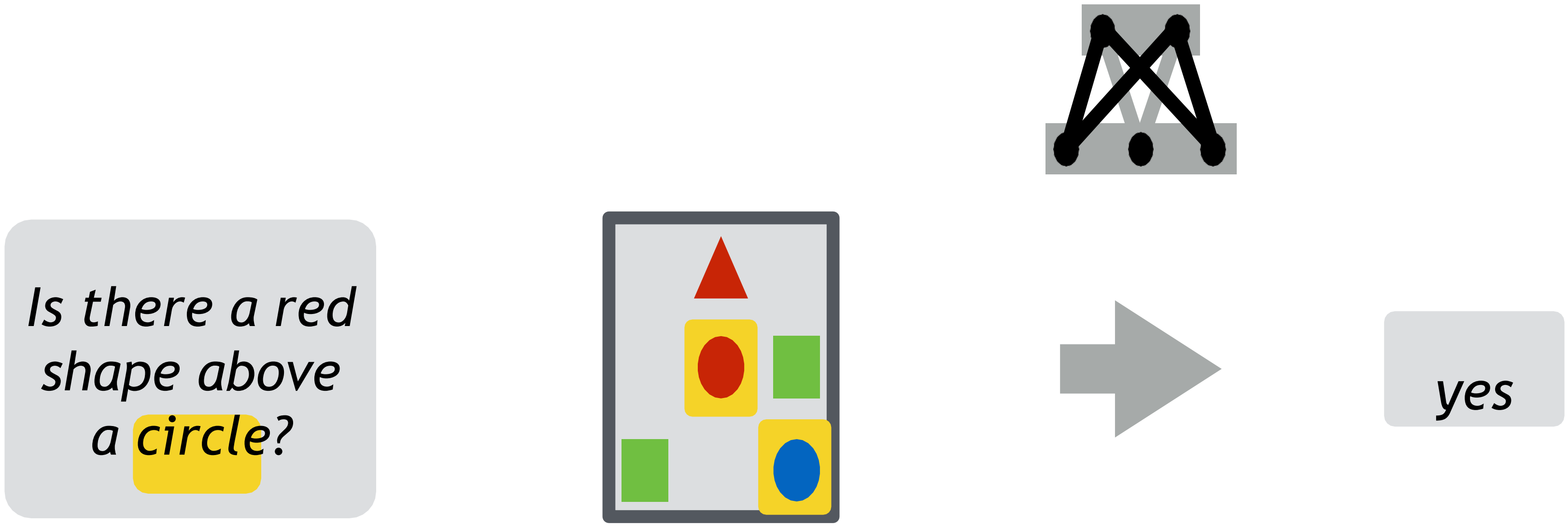
How many benches are shown?

Grounded question answering



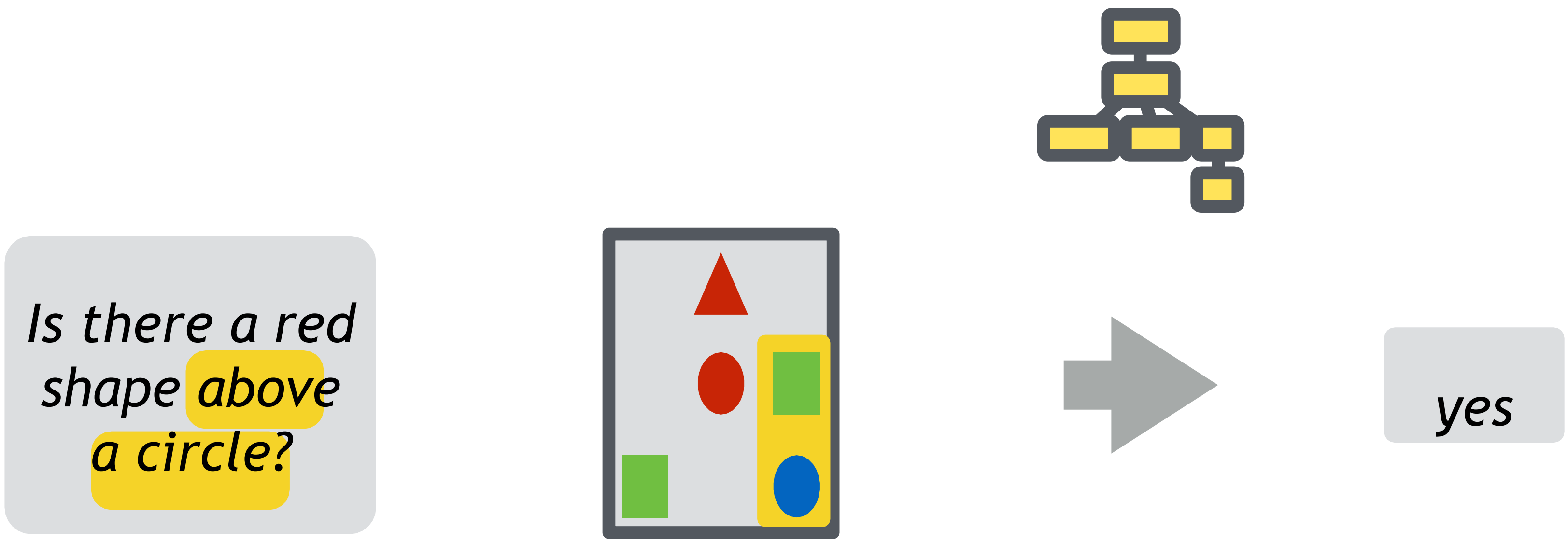
Slide credit: JacobAndreas

Neural nets learn lexical groundings



[Iyyer et al. 2014, Bordes et al. 2014, Yang et al. 2015, Malinowski et al., 2015]

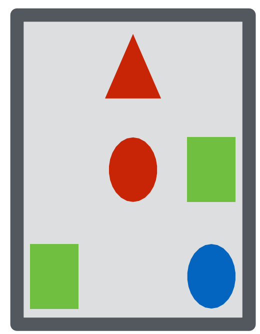
Semantic parsers learn composition

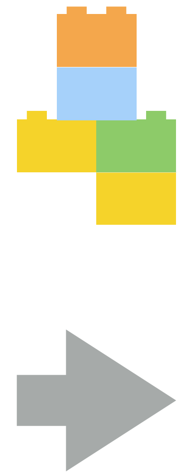


[Wong & Mooney 2007, Kwiatkowski et al. 2010, Liang et al. 2011, A et al. 2013]

Neural module networks learn both!

Is there a red shape above a circle?



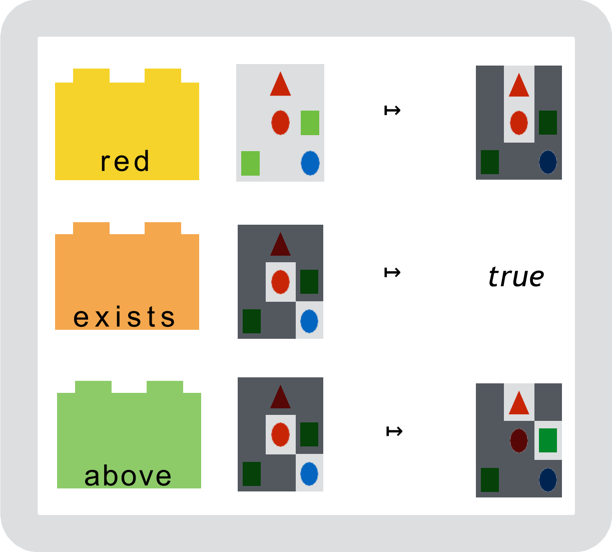


yes

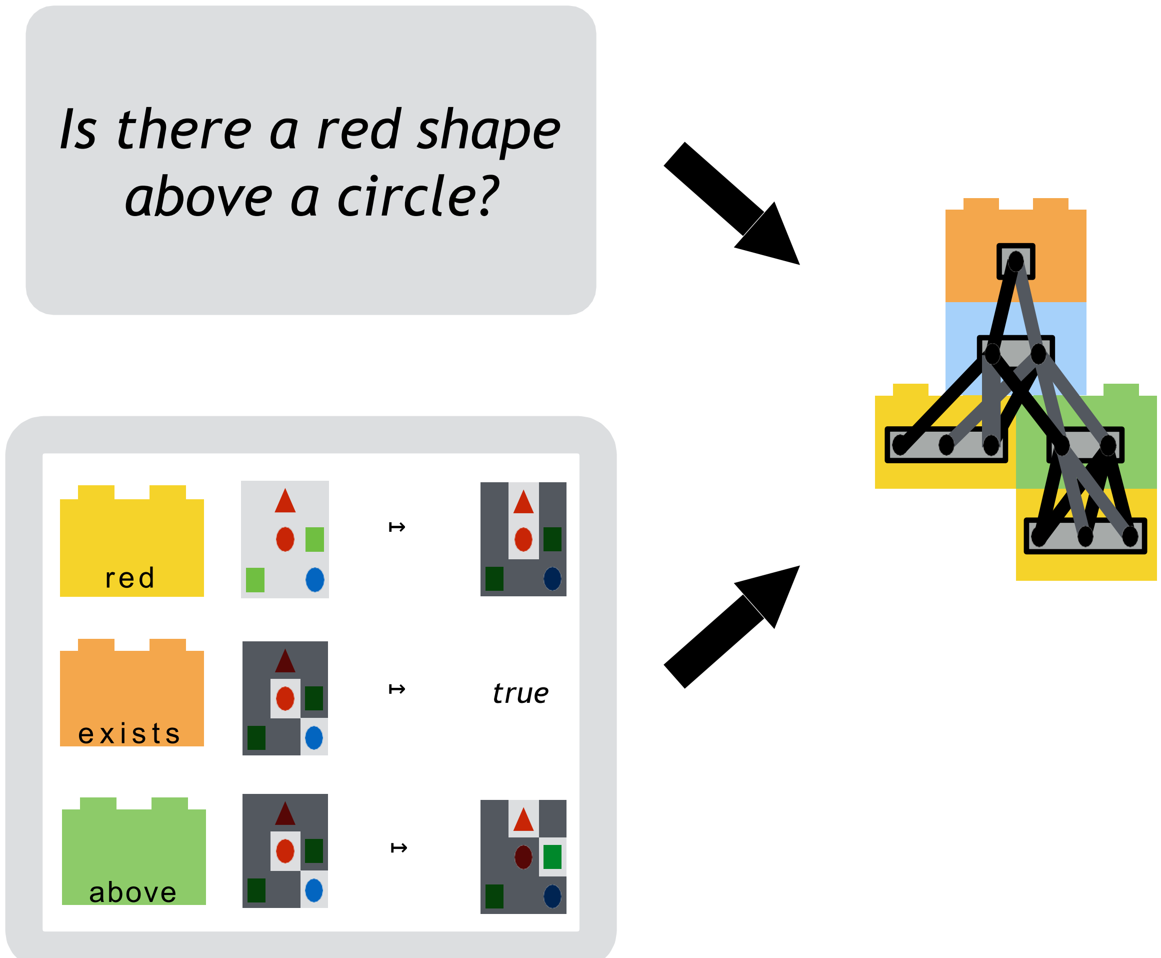
Slide credit: JacobAndreas

Neural module networks

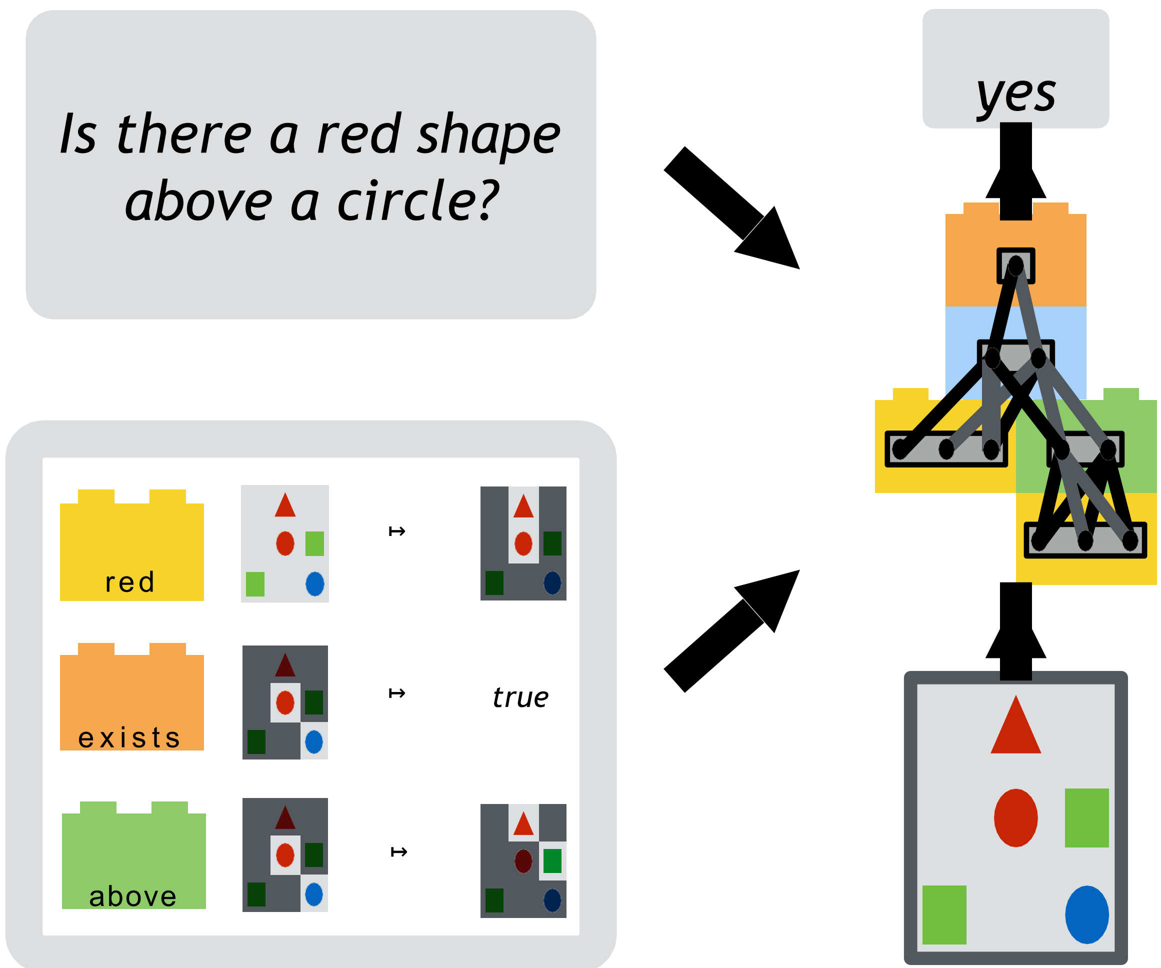
Is there a red shape above a circle?



Neural module networks

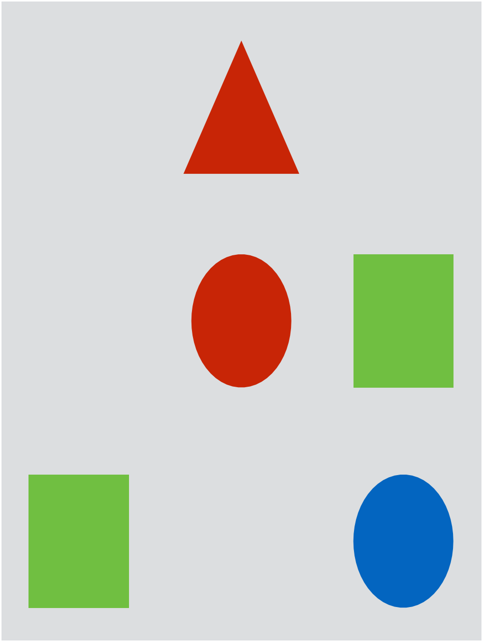


Neural module networks

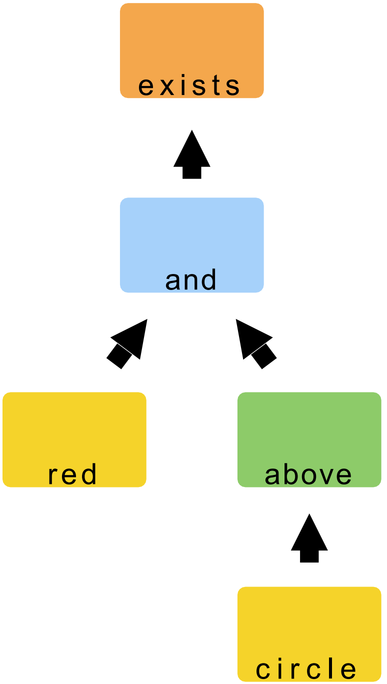


Slide credit: JacobAndreas

Sentence meanings are computations



Is there a red shape above a circle?



NLVR^2: natural language for visual reasoning! (Suhr et al., 2018)

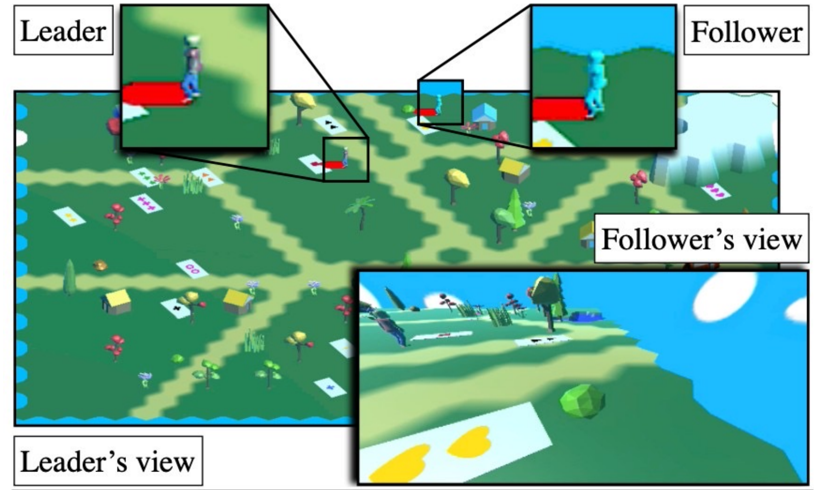
TRUE OR FALSE: the left image contains twice the number of dogs as the right image, and at least two dogs in total are standing.

CerealBar: Situated, Collaborative Natural Language Understanding

CerealBar is a two-person collaborative game. We built CerealBar to study natural language understanding in collaborative interactions.

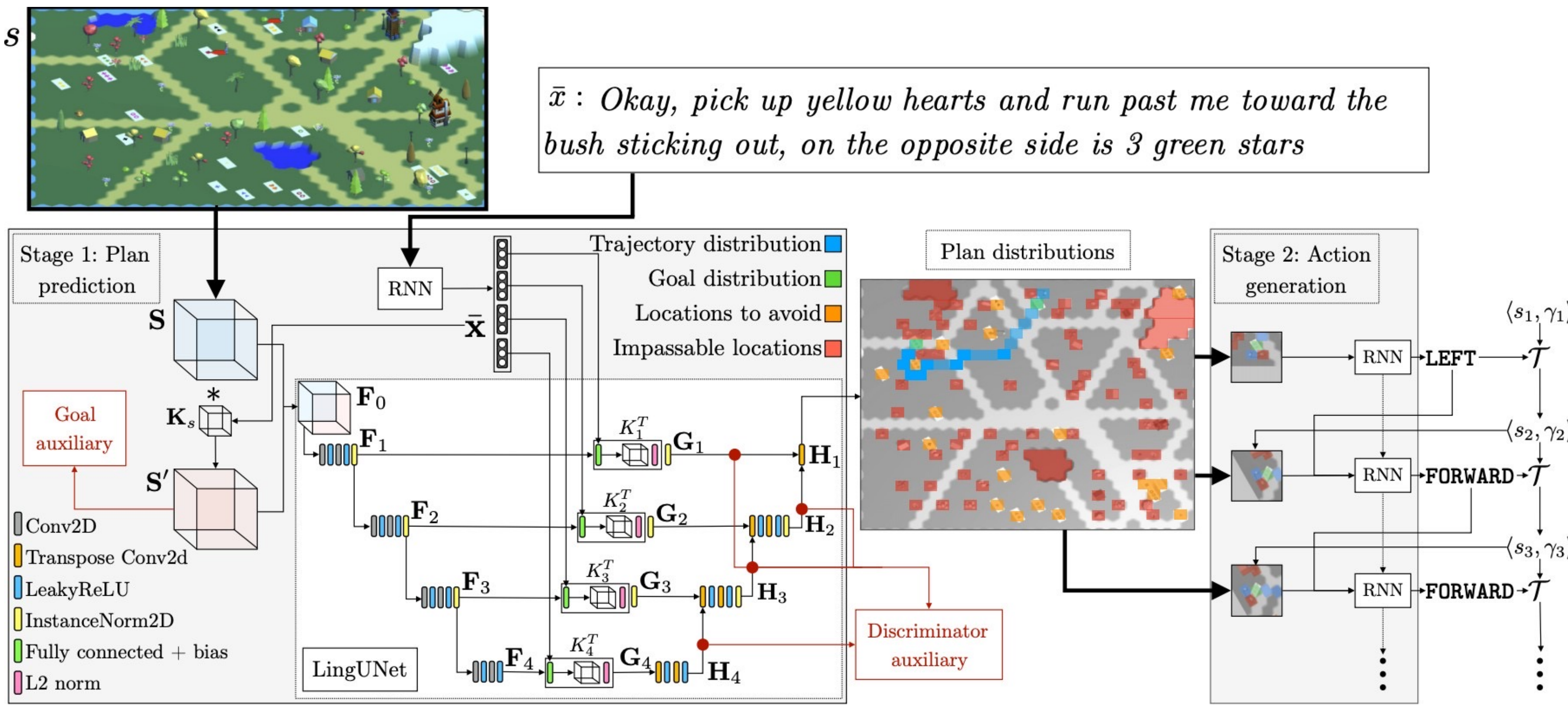
- Two players -- a **leader** and a **follower** -- take turns moving around the game board to collect sets of cards and earn points.
- In addition to moving, the **leader** uses their access to the full environment to plan which set of cards should be collected next, and writes instructions to the follower.
- The **follower** only has access to a first-person view, so their job is to follow the leader's instructions to the best of their ability. However, the follower can move farther than the leader in each turn.

We crowdsourced interactions between human players in the CerealBar game. We also designed and trained a **neural network agent** to play as the follower in CerealBar. Our approach makes contributions in modeling, learning, and evaluation. The CerealBar game, data, and modeling approach is described in Suhr et al. 2019 (EMNLP 2019).



...
$\bar{x}_3$: *turn left and head toward the yellow hearts, but don't pick them up yet. I'll get the next card first.*
$\bar{x}_4$: ***Okay, pick up yellow hearts and run past me toward the bush sticking out, on the opposite side is 3 green stars***
`[Set made. New score: 4]`
...

Figure 1: A snapshot from an interaction in CEREALBAR. The current instruction is in bold. The large image shows the entire environment. This overhead view is available only to the leader. The follower sees a first-person view only (bottom right). The zoom boxes (top) show the leader and follower.



Suhr et al., 2019 (“CEREALBAR”)

Image Captioning

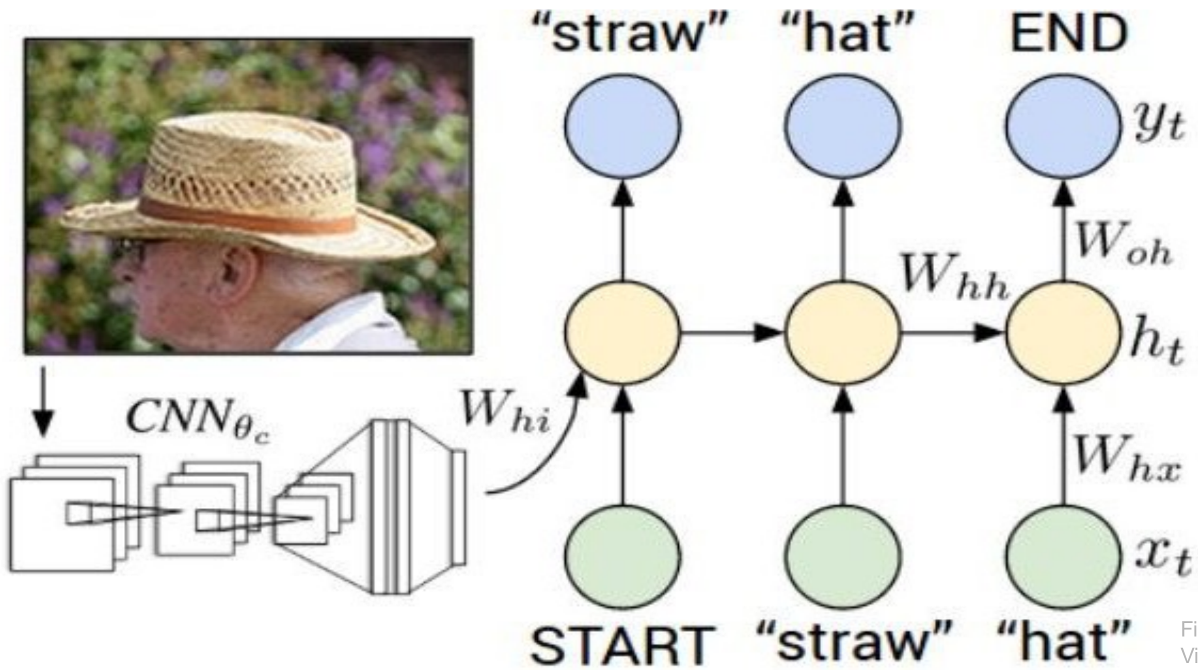


Figure fr
Visual-S
Image D
copyrigh
Reprodu

Around 2014

- Explain Images with Multimodal Recurrent Neural Networks, Mao et al.
- Deep Visual-Semantic Alignments for Generating Image Descriptions, Karpathy and Fei-Fei
- Show and Tell: A Neural Image Caption Generator, Vinyals et al.
- Long-term Recurrent Convolutional Networks for Visual Recognition and Description, Donahue et al.
- Learning a Recurrent Visual Representation for Image Caption Generation, Chen and Zitnick

test image

test image

This is our ImageNet CNN, now used as a feature extractor image

conv-64
conv-64
maxpool

conv-128
conv-128
maxpool

conv-256
conv-256
maxpool

conv-512
conv-512
maxpool

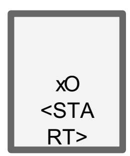
conv-512
conv-512
maxpool

FC-4096
FC-4096
~~FC-1000~~
~~softmax~~

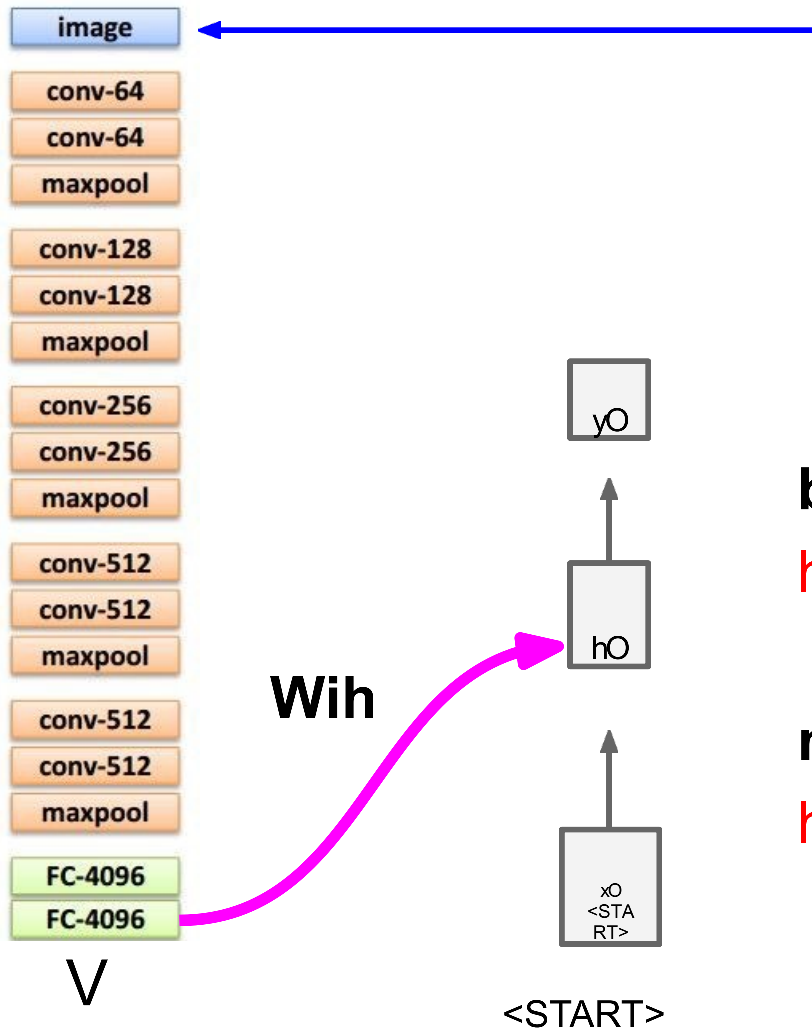
test image

This is our ImageNet CNN, now used as a feature extractor

test image



<START>



test image

before:

$$h = \tanh(W_{xh} * x + W_{hh} * h)$$

now:

$$h = \tanh(W_{xh} * x + W_{hh} * h + W_{ih} * v)$$

let's use the image features to create a conditional LM

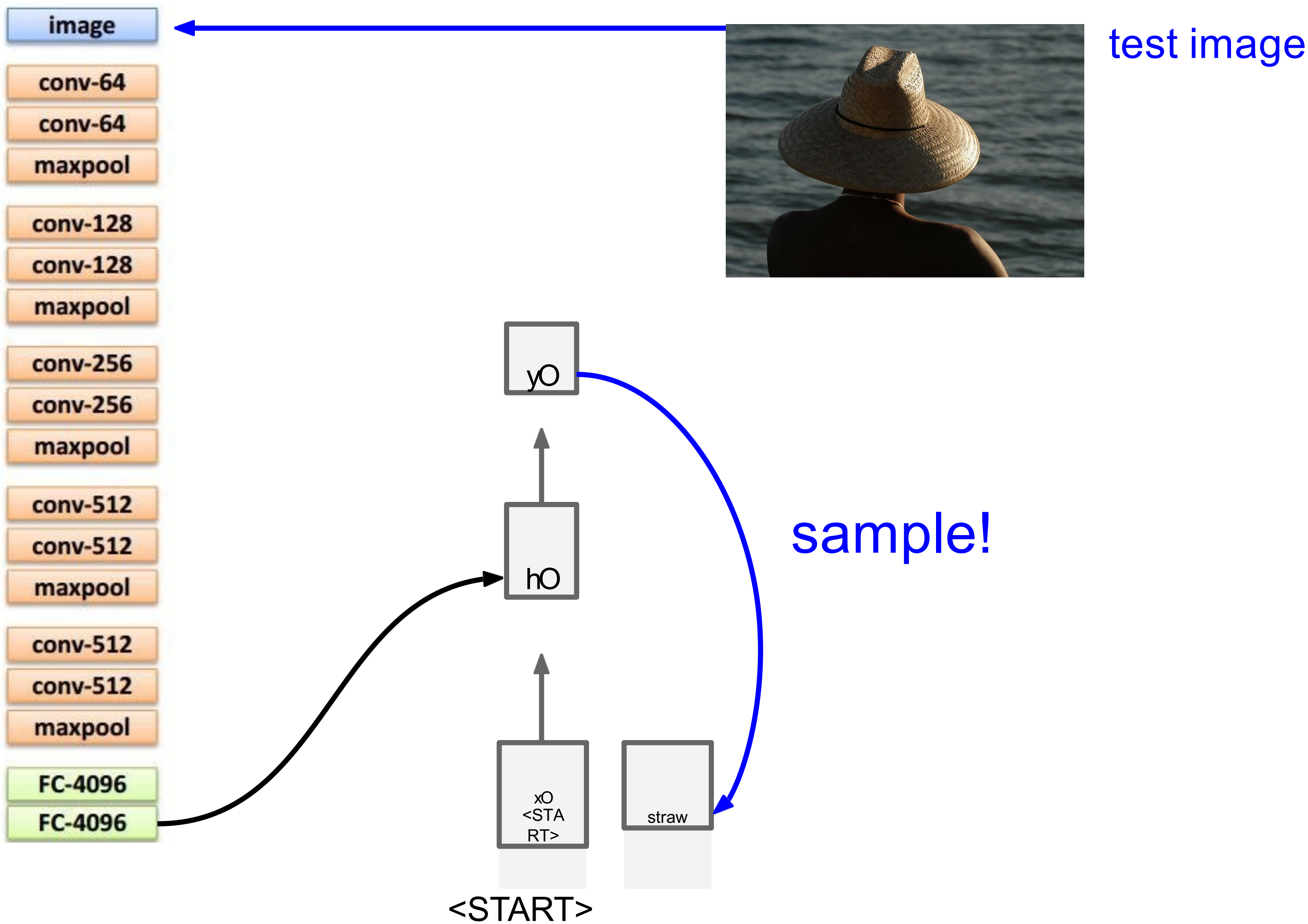
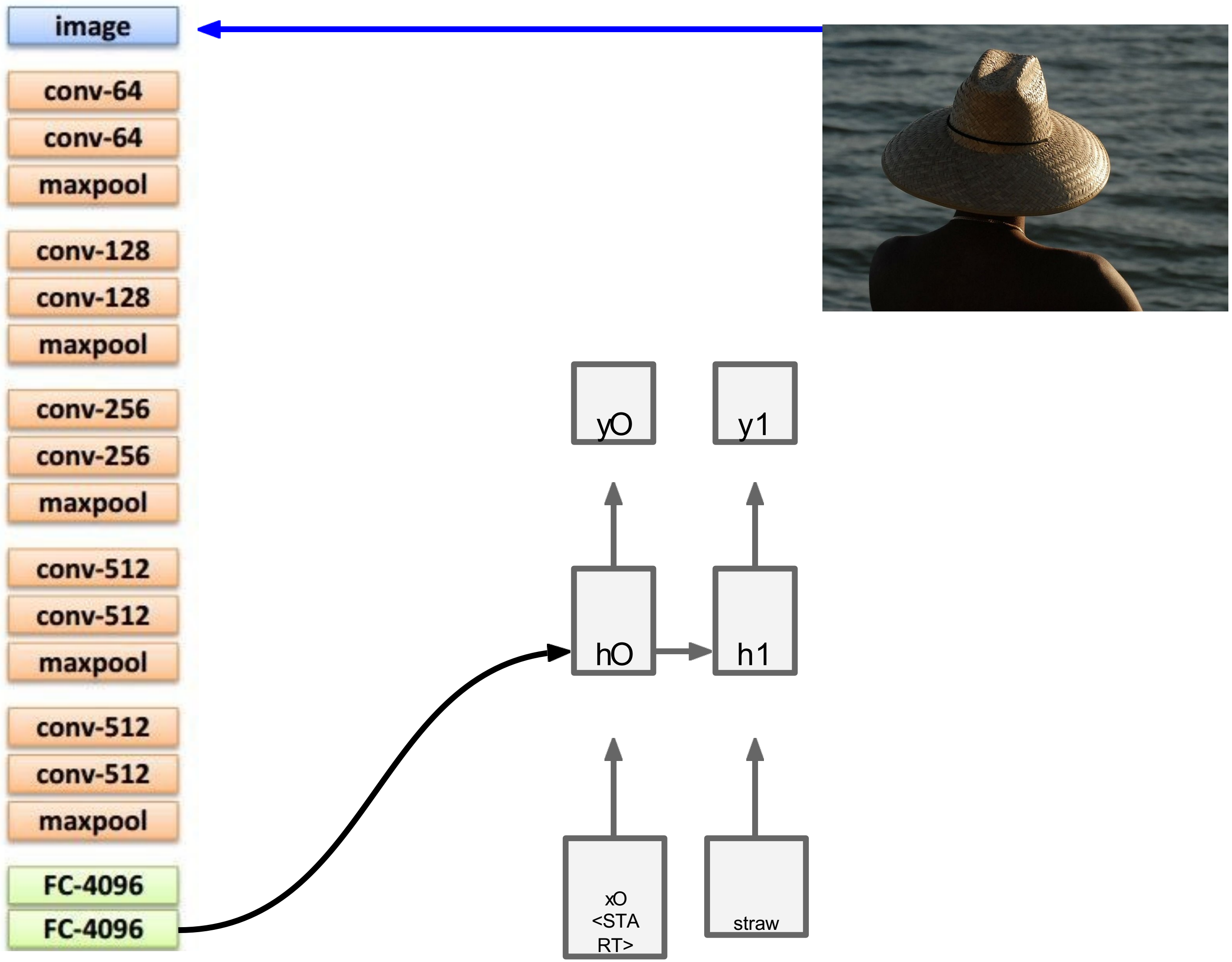


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
yO
hO
sample!
xO
<STA
RT>
straw
<START>



test image

<START>

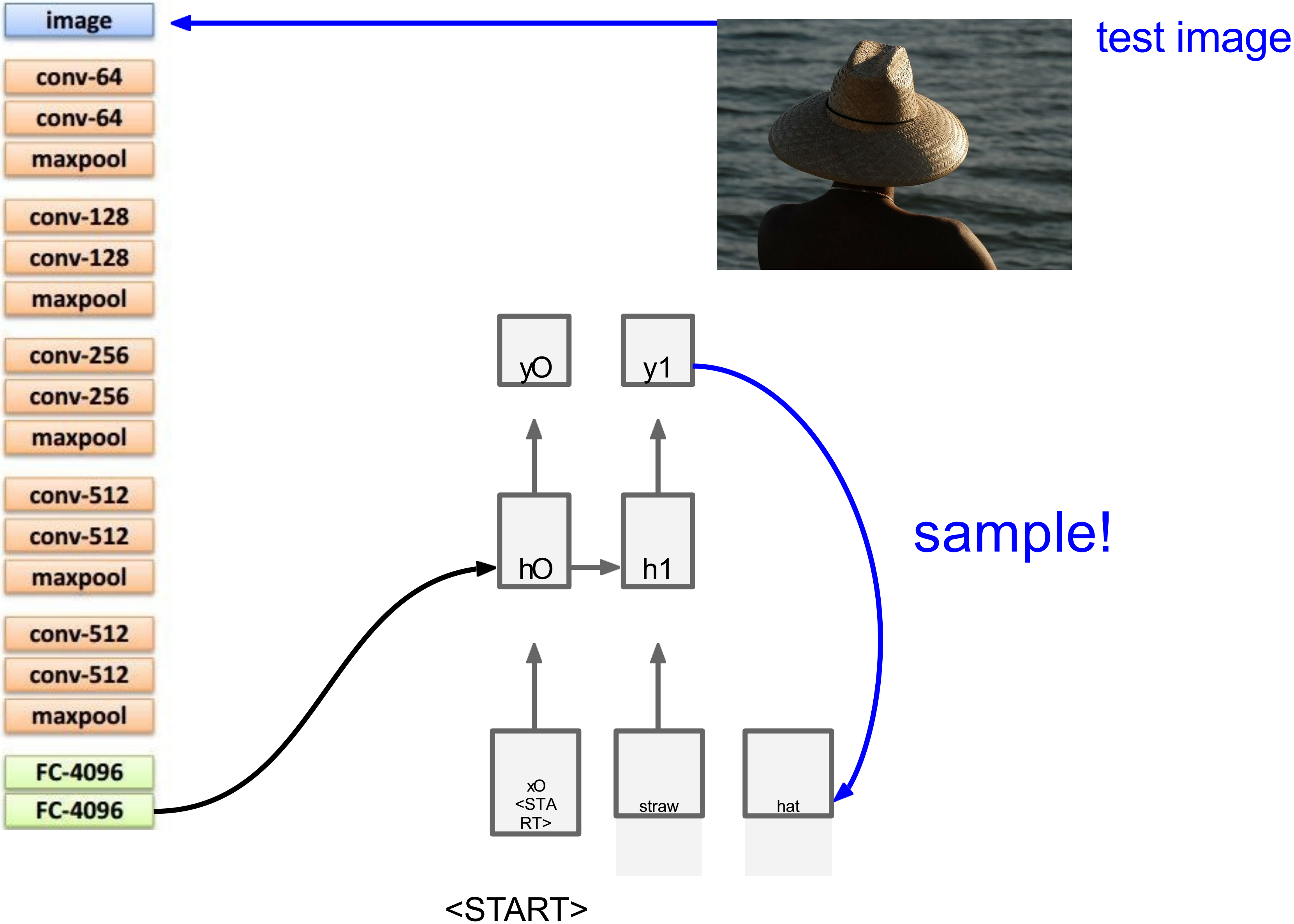


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
y1
h0
h1
sample!
x0 <START>
straw
hat
<START>

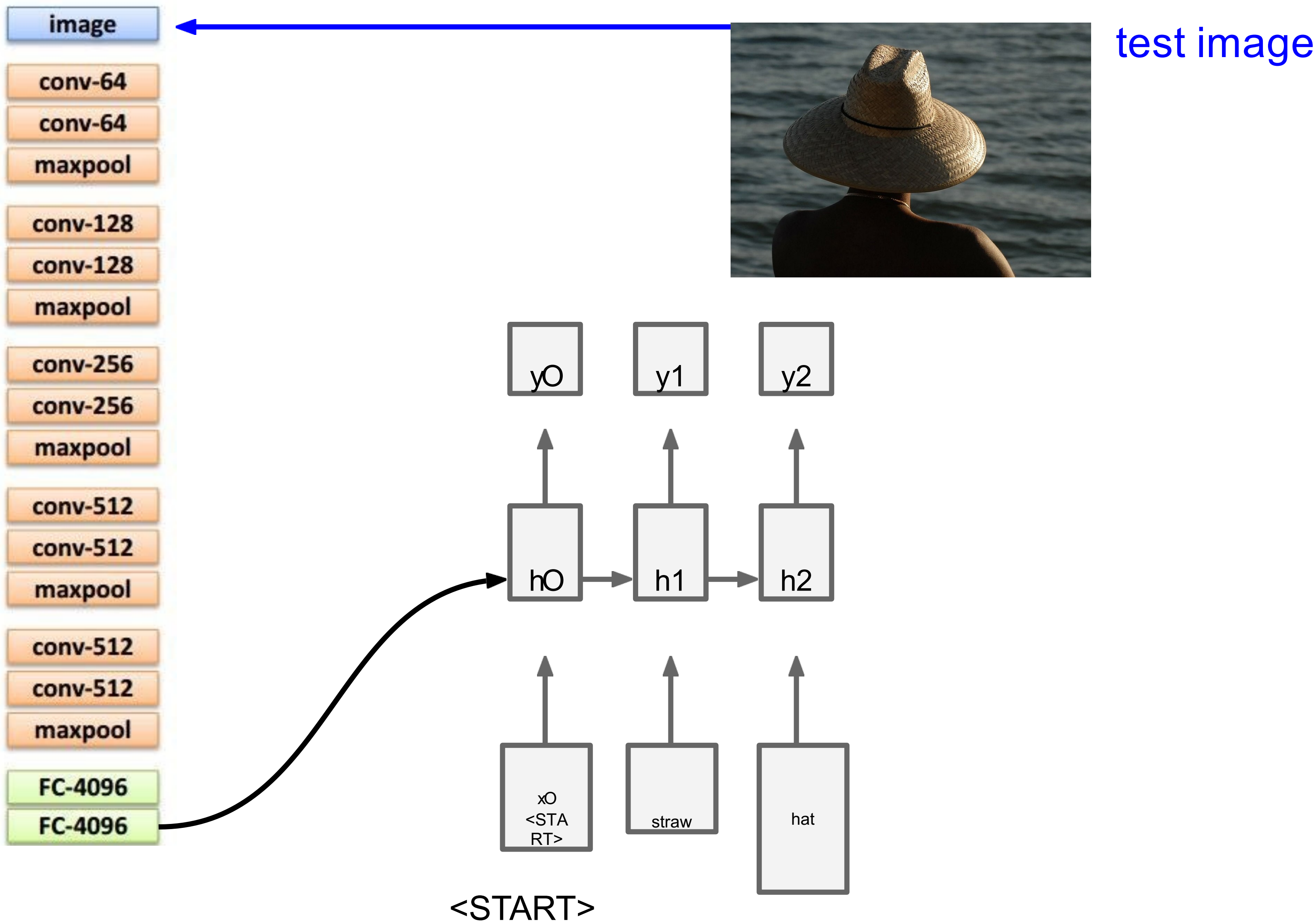


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
yO
y1
y2
hO
h1
h2
xO
<STA
RT>
straw
hat
<START>

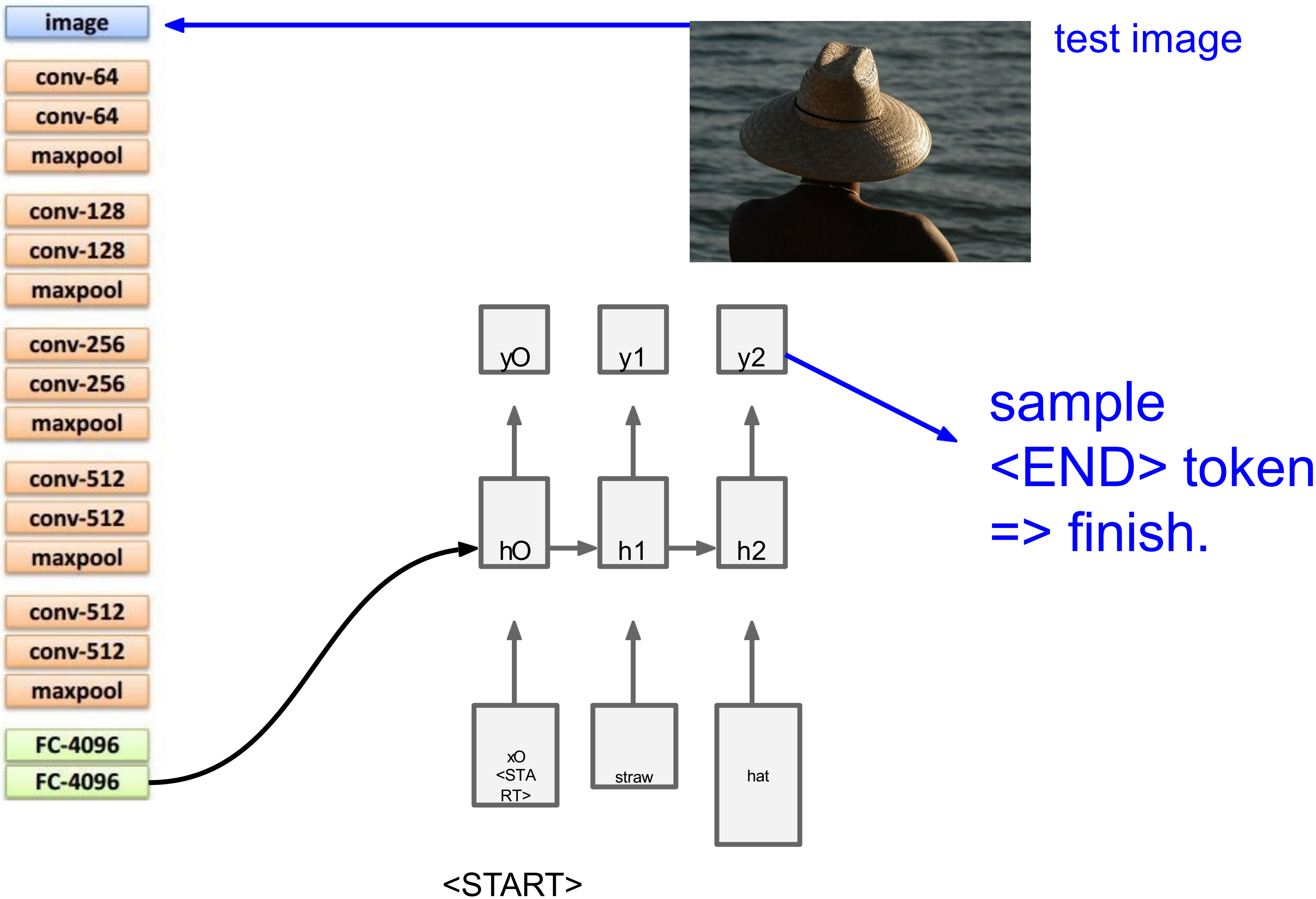
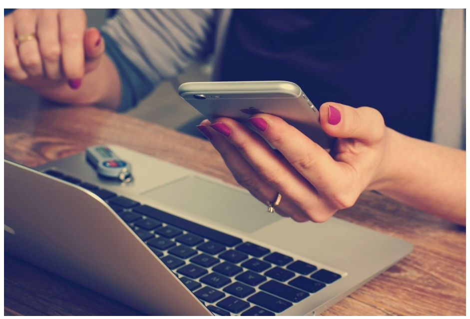


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
y0
y1
y2
h0
h1
h2
x0 <START>
straw
hat
sample <END> token => finish.
<START>

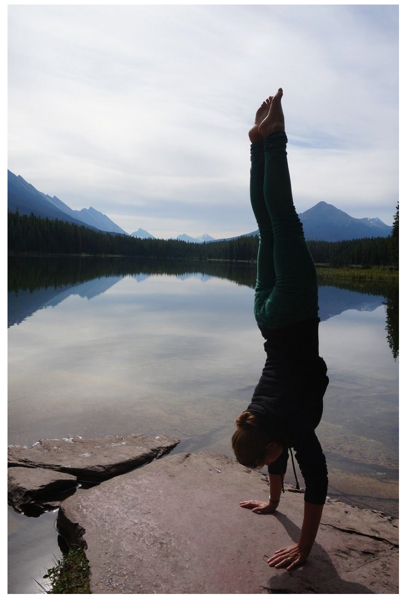
Image Captioning: Failure Cases

Captions generated using neuraltalk2
All images are CC0 Public domain: fu
coat, handstand, spider web, basebal

A woman is holding a cat in her hand



A person holding a computer mouse on a desk



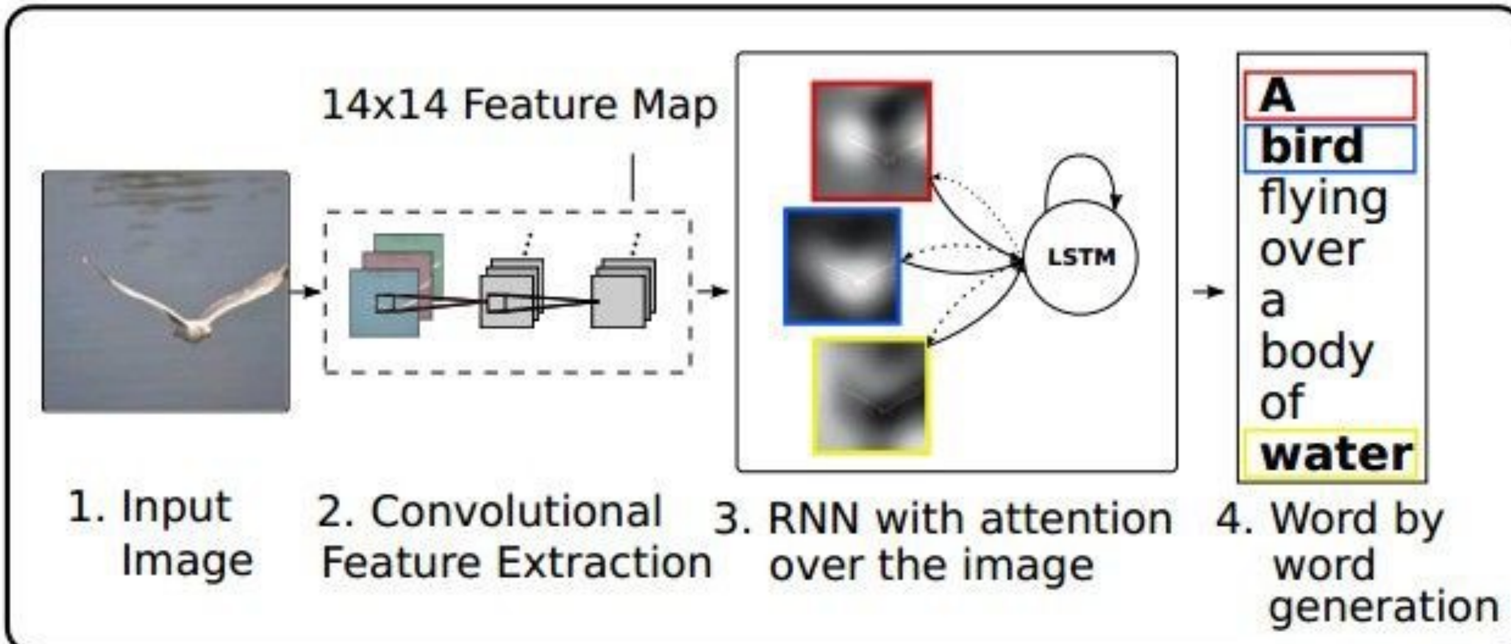
A woman standing on a beach holding a surtboard

A bird is perched on a tree branch

A man in a baseball unitorm throwing a ball

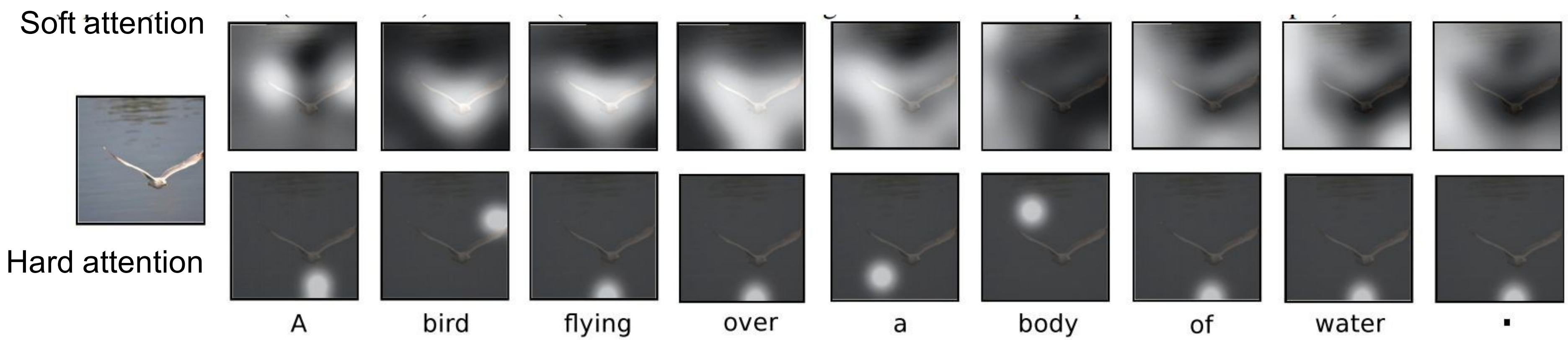
Image Captioning with Attention

RNN focuses its attention at a different spatial location when generating each word



Xu et al, "Show, Attend, and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Image Captioning with Attention



Xu et al, "Show, Attend, and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Image Captioning with Attention

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Xu et al, "Show, Attend, and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Image Captioning using Transformers

Hybrid Solution

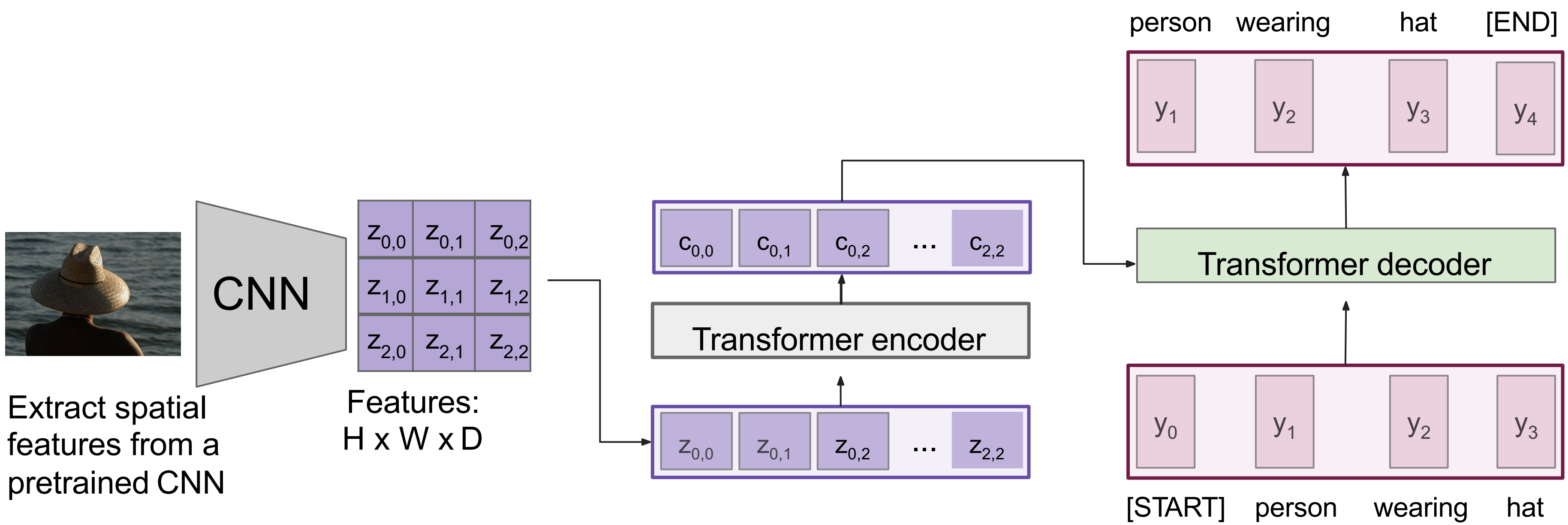
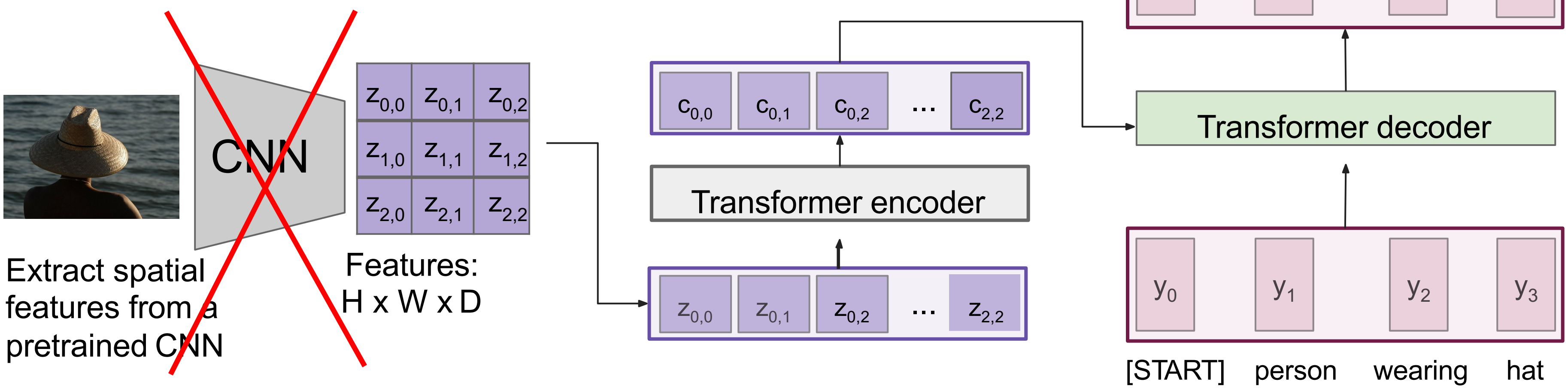


Image Captioning using transformers

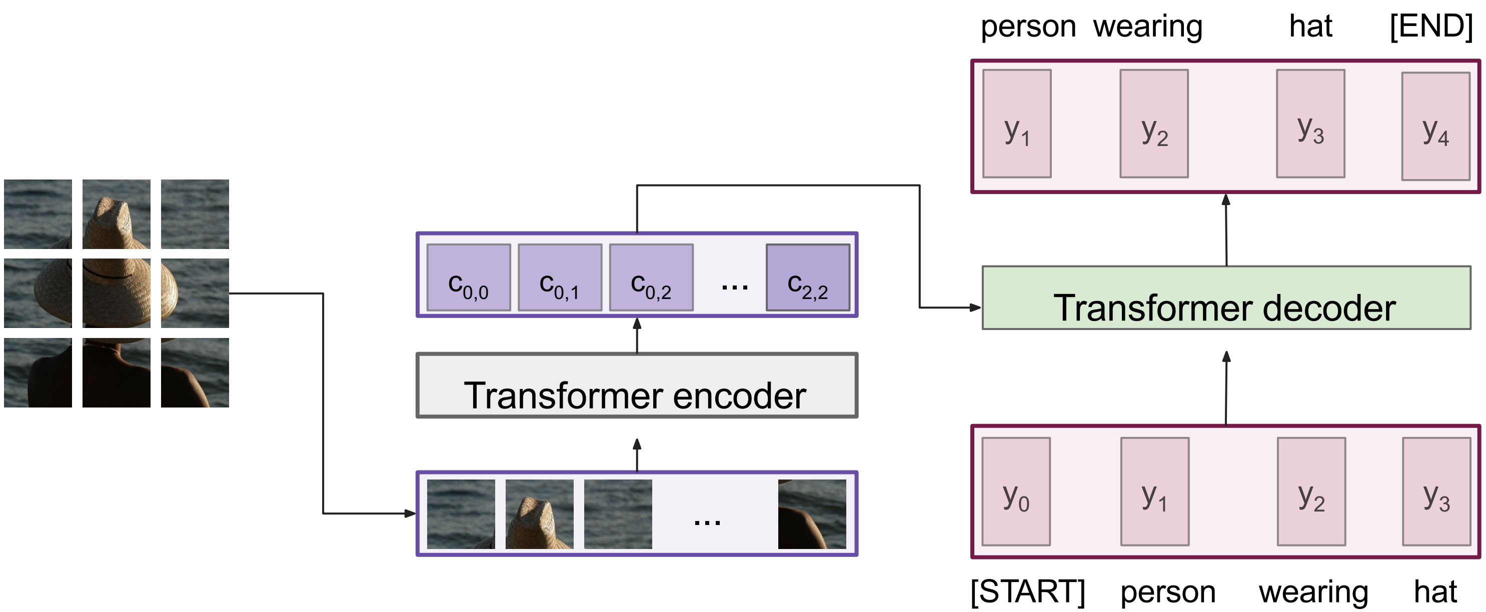
- **Perhaps we don't need convolutions at all?**



http://people.cs.umass.edu/~miyyer/cs685/

Image Captioning using ONLY transformers

- **Transformers from pixels to language**



Dosovitskiy et al, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ArXiv 2020
Colab link to an implementation of vision transformers

Vision Transformers (ViT) vs. ResNets (BiT)

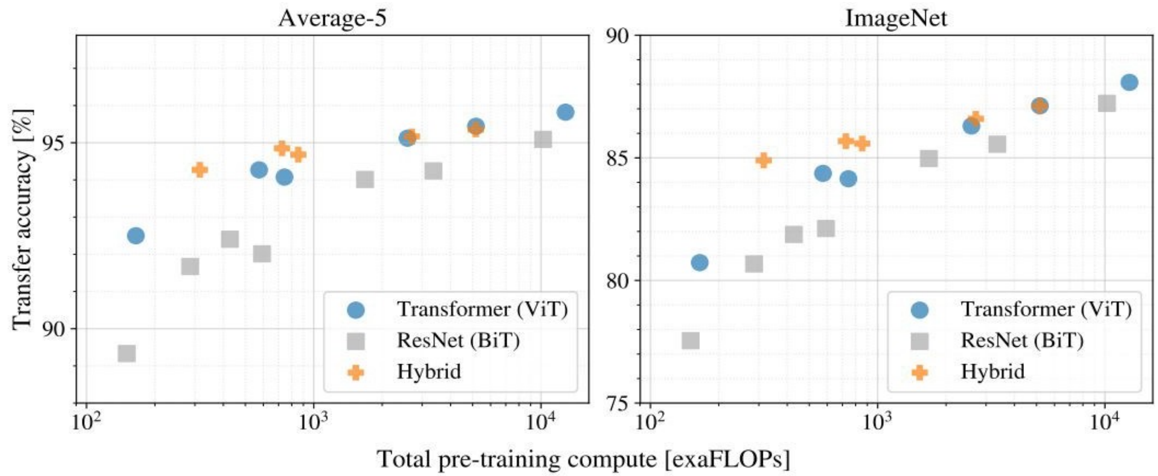


Figure 5: Performance versus cost for different architectures: Vision Transformers, ResNets, and hybrids. Vision Transformers generally outperform ResNets with the same computational budget. Hybrids improve upon pure Transformers for smaller model sizes, but the gap vanishes for larger models.

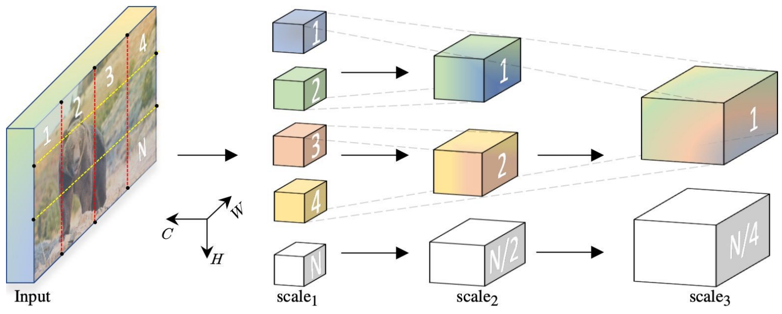
The BiT model was proposed in [Big Transfer (BiT): General Visual Representation Learning] by Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, Neil Houlsby. BiT is a simple recipe for scaling up pre-training of [ResNet]-like architectures (specifically, ResNetv2). The method results in significant improvements for transfer learning.

Dosovitskiy et al, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", ArXiv 2020
Colab link to an implementation of vision transformers

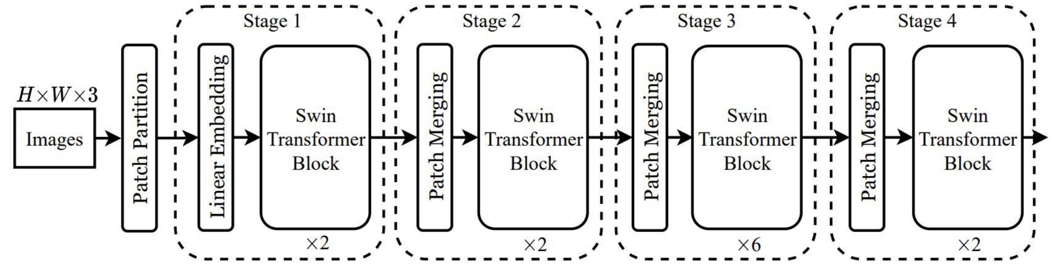
Intelligent Agents
Vision and Language

Prof. Dr. Ralf Möller
Universität zu Lübeck
Institut für Informationssysteme

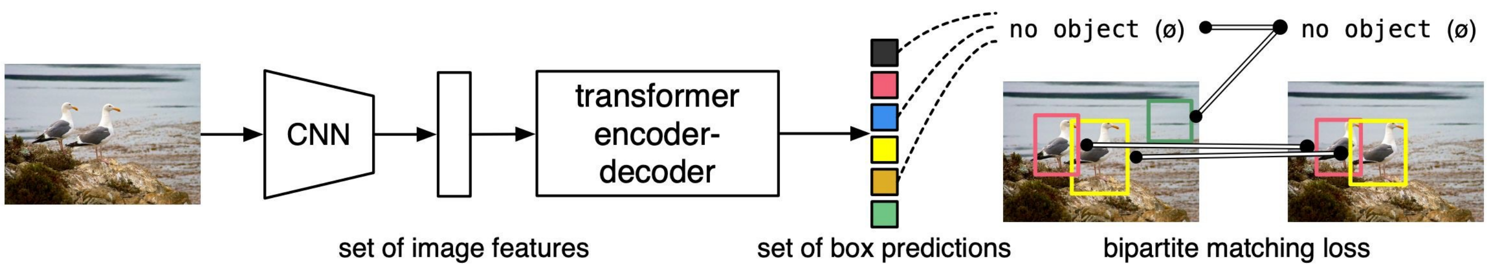
Vision Transformers



Fan et al, “Multiscale Vision Transformers”, ICCV 2021



Liu et al, “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows”, CVPR 2021



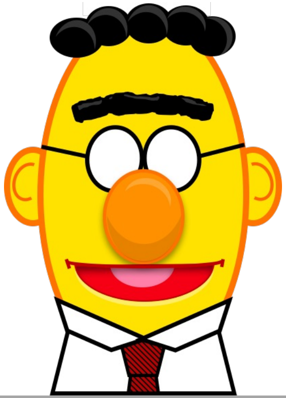
Carion et al, “End-to-End Object Detection with Transformers”, ECCV 2020

ViLBERT (Vision and Language BERT)

ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. Proceedings of the 33rd International Conference on Neural Information Processing Systems. Curran Associates Inc., Red Hook, NY, USA, Article 2, 13–23. **2019**

Presented by - **Sidharth Singla**, 20774908



Model

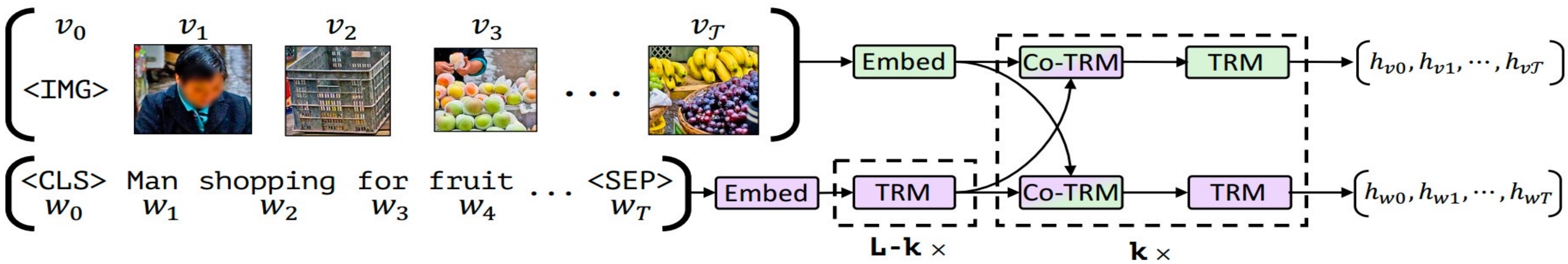
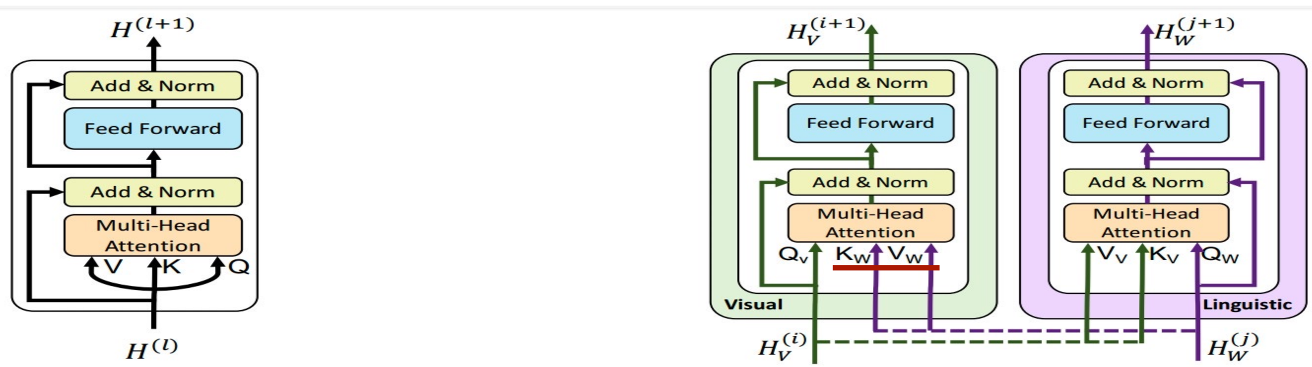


Figure 1: Our ViLBERT model consists of two parallel streams for visual (green) and linguistic (purple) processing that interact through novel co-attentional transformer layers. This structure allows for variable depths for each modality and enables sparse interaction through co-attention. Dashed boxes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

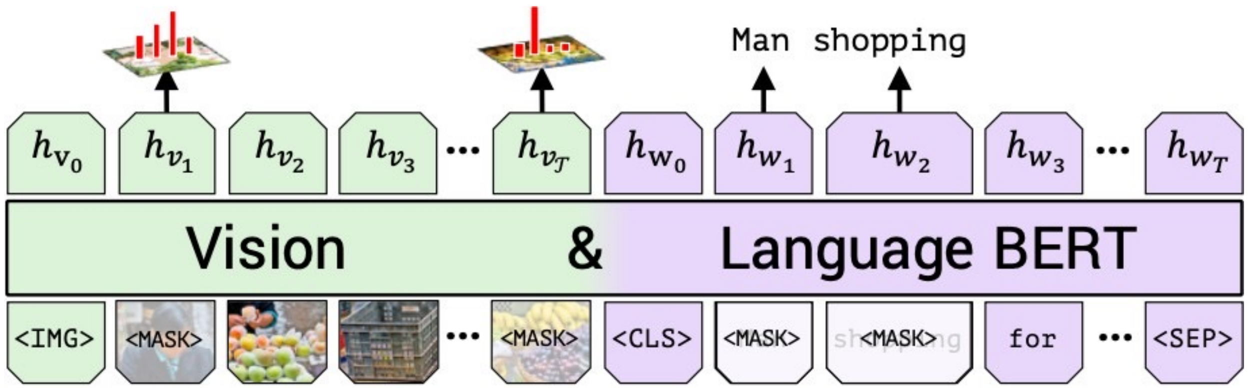


(a) Standard encoder transformer block

(b) Our co-attention transformer layer

Figure 2: We introduce a novel co-attention mechanism based on the transformer architecture. By exchanging key-value pairs in multi-headed attention, this structure enables vision-attended language features to be incorporated into visual representations (and vice versa).

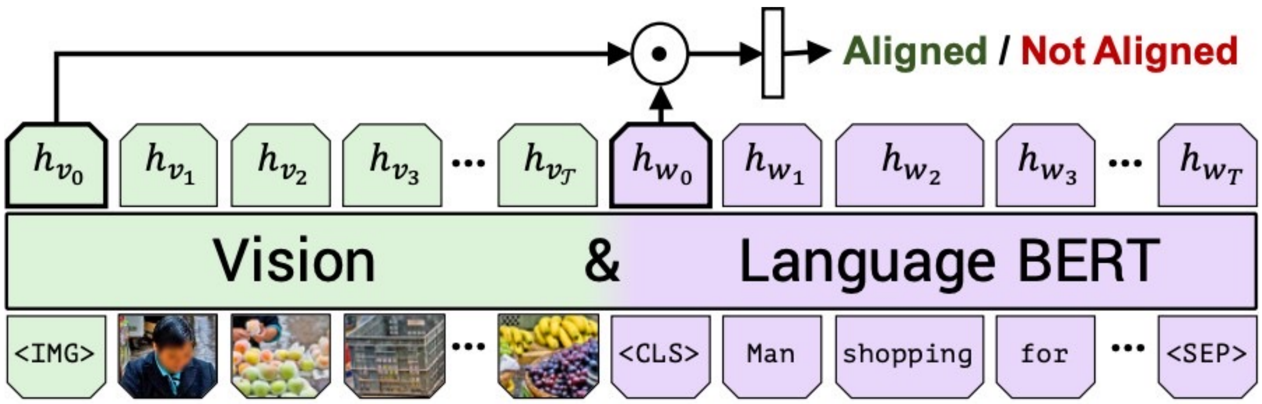
Pretraining: Masked Multi-Modal Learning Task



(a) Masked multi-modal learning

- Approximately 15% of both words and image region are masked and reconstructed given the remaining inputs
- Image features zeroed out 90% and unaltered 10%. Masked text inputs are handled as in BERT
- Model predicts a distribution over semantic classes rather than directly regressing the masked feature values for the corresponding image region
- Supervision by output distribution for the region from the pretrained detection model used. Minimize KL divergence

Pretraining: Multi-modal alignment task



(b) Multi-modal alignment prediction

- Prediction whether the text describes the image(image aligned with the text).
- Element-wise product between h_{IMG} and h_{CLS} and a linear layer is learnt to make the binary prediction
- Trained on Conceptual Captions Dataset
    - Collection of 3.3 million image-caption pairs automatically scraped from alt-text enabled web images

https://ai.google.com/research/ConceptualCaptions/

Transfer tasks

- Pretrained ViLBERT model transferred to a set of four established vision-and-language tasks and one diagnostic task.
- Fine-tuning strategy to modify the pretrained base model and perform the new task by training the entire model end-to-end.

Visual Question Answering (VQA)

Training and Evaluation on VQA 2.0 dataset

- Fine-tuning:
  Two layer MLP is learnt on top of the elementwise product of the image and text representations hIMG and hCLS.

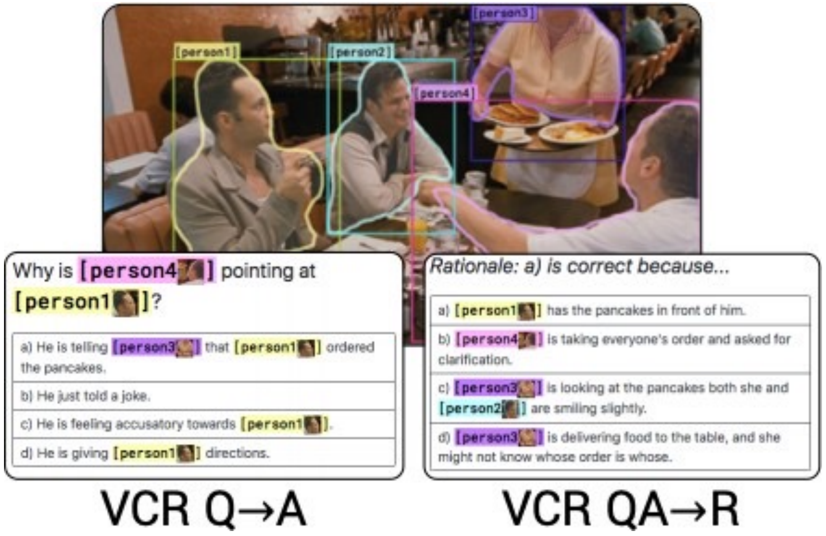
- Multi-label classification task:
  Binary cross-entropy loss.
  Batch size 256. Maximum 20 epochs.
  Initial learning rate 4e-5.

VQA

In information theory, the **cross-entropy** between two probability distributions p and q over the same underlying set of events measures the average number of bits needed to identify an event drawn from the set if a coding scheme used for the set is optimized for an estimated probability distribution q, rather than the true distribution p.

Visual Commonsense Reasoning (VCR)

- Given an image, Visual Question Answering (Q->A) and Answer justification (QA->R).
- Trained on Visual Commonsense Reasoning (VCR) dataset having object tags integrated into the language providing direct grounding supervision and explicitly excludes referring expressions.
- Fine-tuning: Question and each possible response is concatenated and four different text inputs are passed along with the image. A linear layer is learnt on top of the post-element-wise product representation.
- Softmax prediction. Loss - Cross-entropy loss. 20 epochs. Batch size 64. Initial learning rate 2e-5.



https://paperswithcode.com/dataset/vcr

Grounding Referring Expressions

- Localize an image region given a natural language reference.
- Training and Evaluation is done on RefCOCO+ dataset.
- Bounding box proposals provided by *MAttNet[5]*, which use a Mask R-CNN are directly used.
- Fine-tuning: Final representation h_{vi} is passed into a learned linear layer to predict a matching score. IoU is computed with the ground truth box thresholding at 0.5.
- Loss - Binary cross-entropy loss. Maximum 20 epochs. Batch size 256. Initial learning rate 4e-5.

Caption-Based Image Retrieval

- **Caption-Based Image Retrieval**
  - Identifying an image from a pool given a caption describing its content.
  - Training and Evaluation is done on the Flickr30k dataset.Trained in a 4-way multiple-choice setting by randomly sampling three distractors for each image-caption pair - substituting a random caption, a random image, or a hard negative from among the 100 nearest neighbors of the target image.
  - Alignment score(same as in alignment prediction pretraining) is computed for each. Softmax applied. Loss - Cross-entropy loss. 20 epochs. Batch size64. Initial learning rate 2e-5.

- **'Zero-shot' Caption-Based Image Retrieval**
  - Pre-trained multi-modal alignment prediction model on Conceptual Captions dataset is used directly. No fine-tuning.
  - Demonstrates that the pretraining has developed the ability to ground text.Tested on the caption- based image retrieval task test-set.

Contrastive pretraining

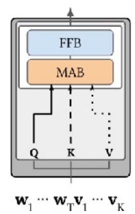
- During unsupervised contrastive pre-training,
- **the unlabeled images are clustered in the latent space,**
- **forming fairly good decision boundaries between different classes**.
- Based on this clustering, the subsequent supervised fine-tuning
- will achieve better performance than random initialization.

Nowadays: Many different V&L BERTs

Single- &Dual-Stream Architectures

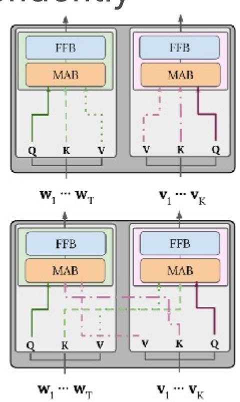
Single-Stream

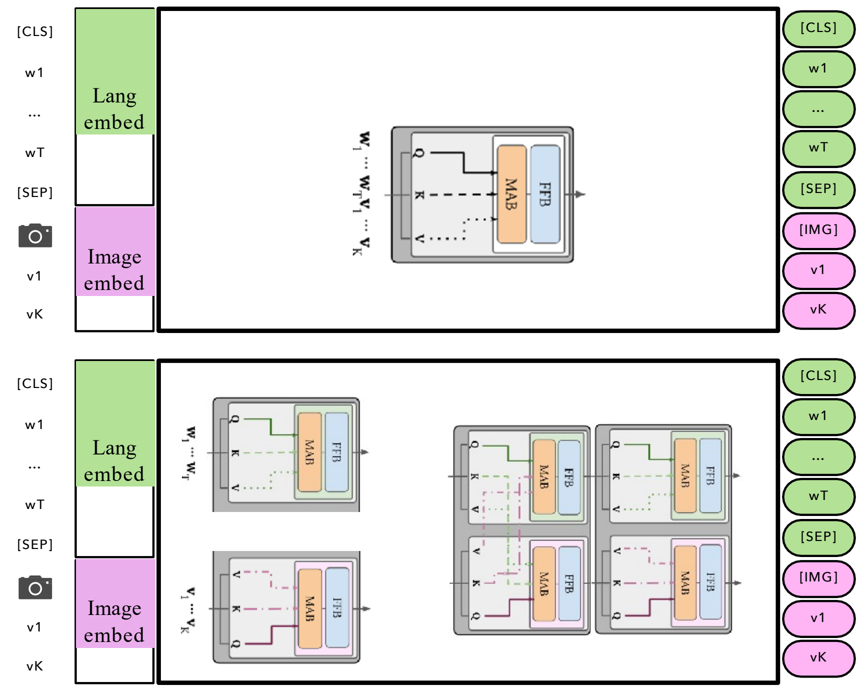
- Concat image–text inputs



Dual-Stream

1. Image and text independently
2. Cross-modal layers
   - Intra-modal
   - Inter-modal





General approach

AI becomes successful:
Not just knowledge representation languages,
but systems that can be used out of the box and
that can be fine-tuned

- Unsupervised pretraining
  - Zero-shot training / generalization
  - Few-shot training / examples
  - Effective for very large vision&language models
- Fine-tuning for specific tasks
  - Reinforcement

Acknowledgements

Learning Transferable Visual Models From Natural Language Supervision

Alec Radford, JongWook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, Ilya Sutskever

OpenAI

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, Ilya Sutskever: Learning Transferable Visual Models From Natural Language Supervision. ICML **2021**: 8748-8763

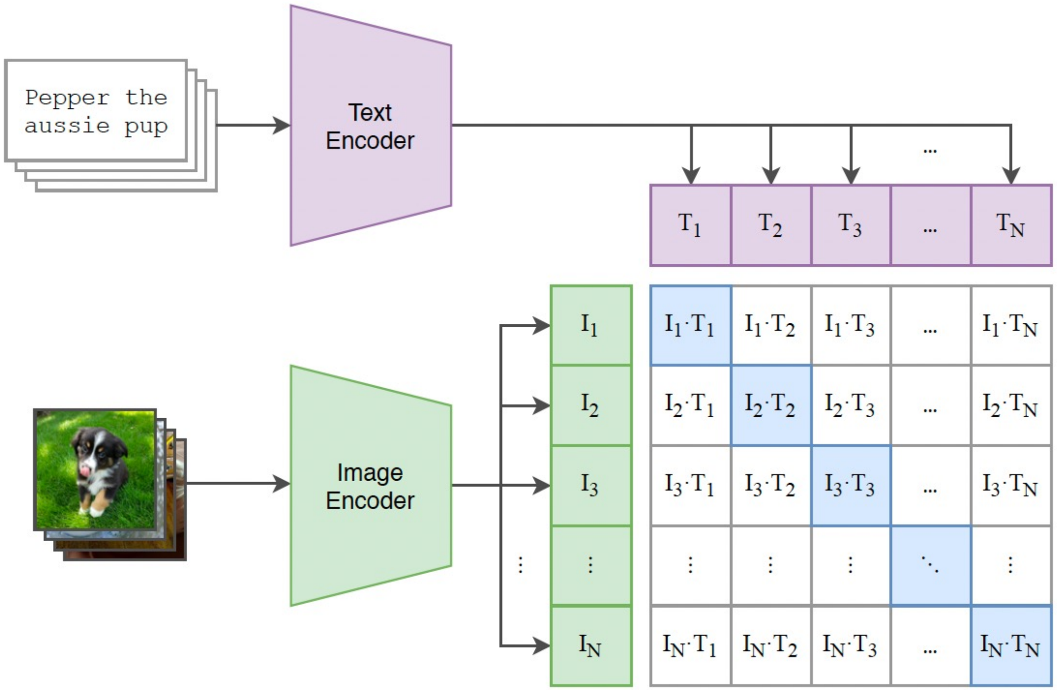
Contrastive language-image pretraining

- ViLBERT and similar methods (e.g., LXMERT) rely on small labeled datasets like MS COCO and Visual Genome (~100K images each)

- OpenAI collected 400 million (image, text) pairs from the web

- Then, they train an image encoder and a text encoder with a simple contrastive loss: given a collection of images and text, predict which (image, text) pairs actually occurred in the dataset

Method: Contrastive Pre-training

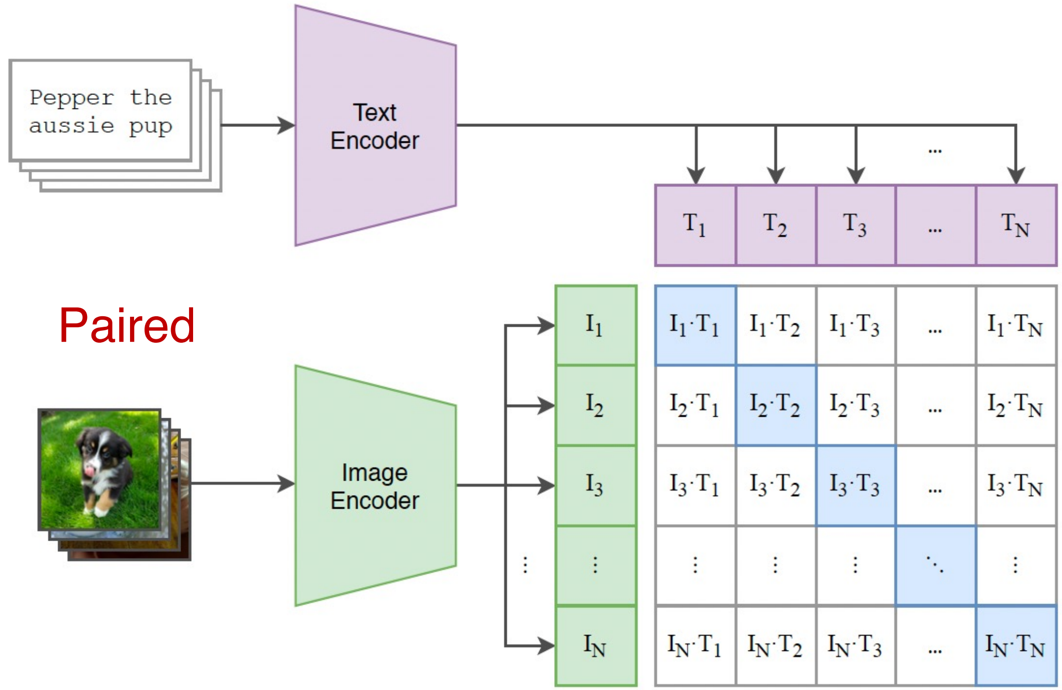


Method: Contrastive Pre-training

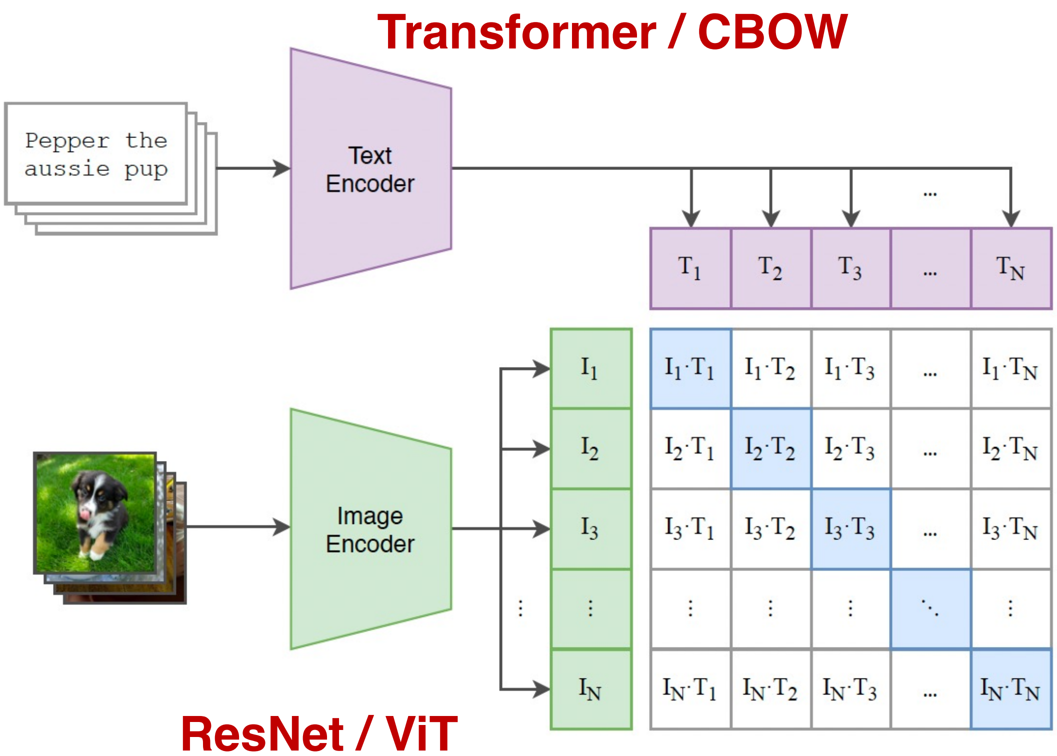
Batch of texts

Paired

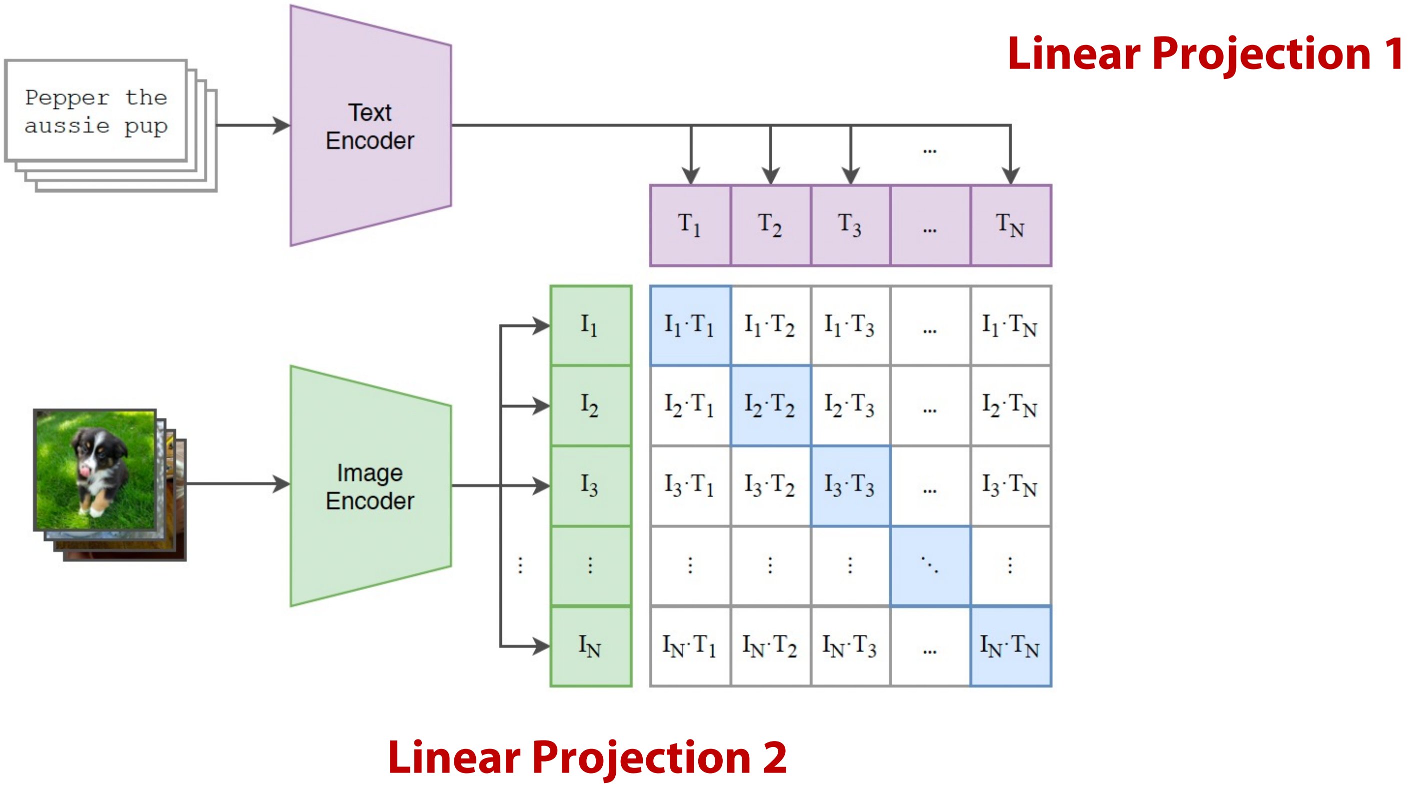
Batch of images
(Resized & Cropped)



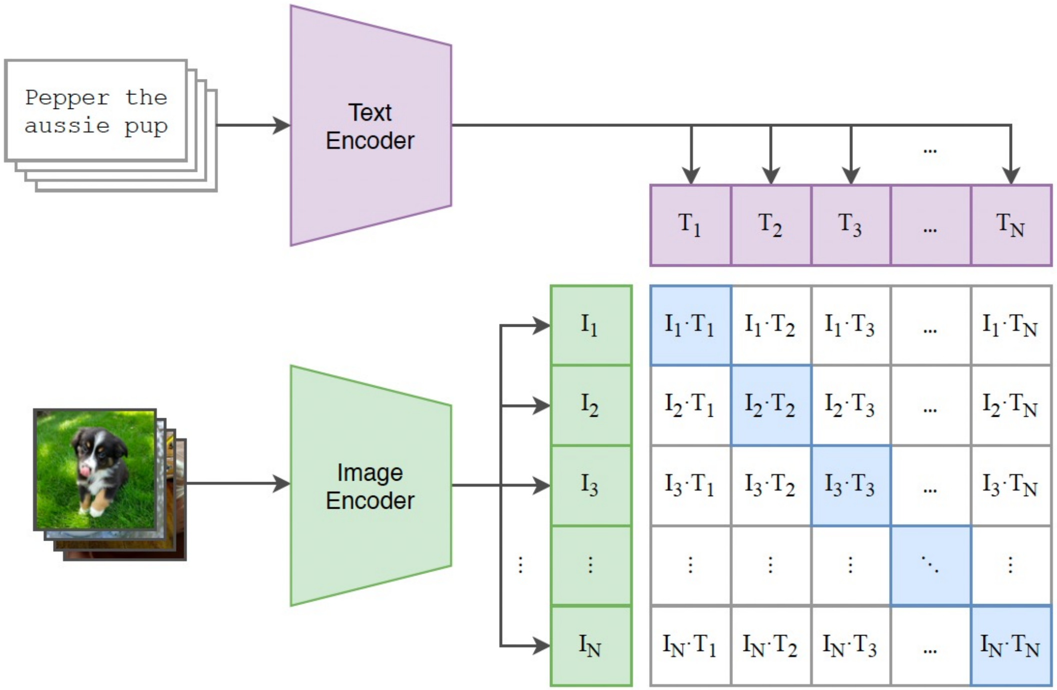
Method: Contrastive Pre-training



Method: Contrastive Pre-training

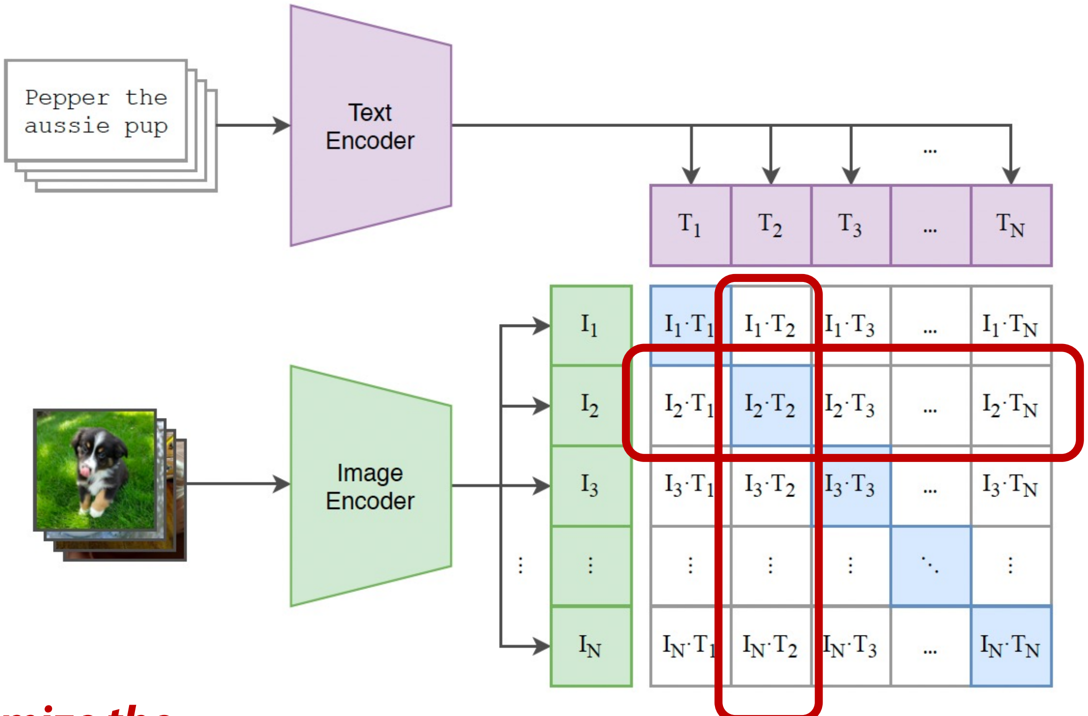


Method: Contrastive Pre-training



Cosine Similarity Matrix

Method: Contrastive Pre-training

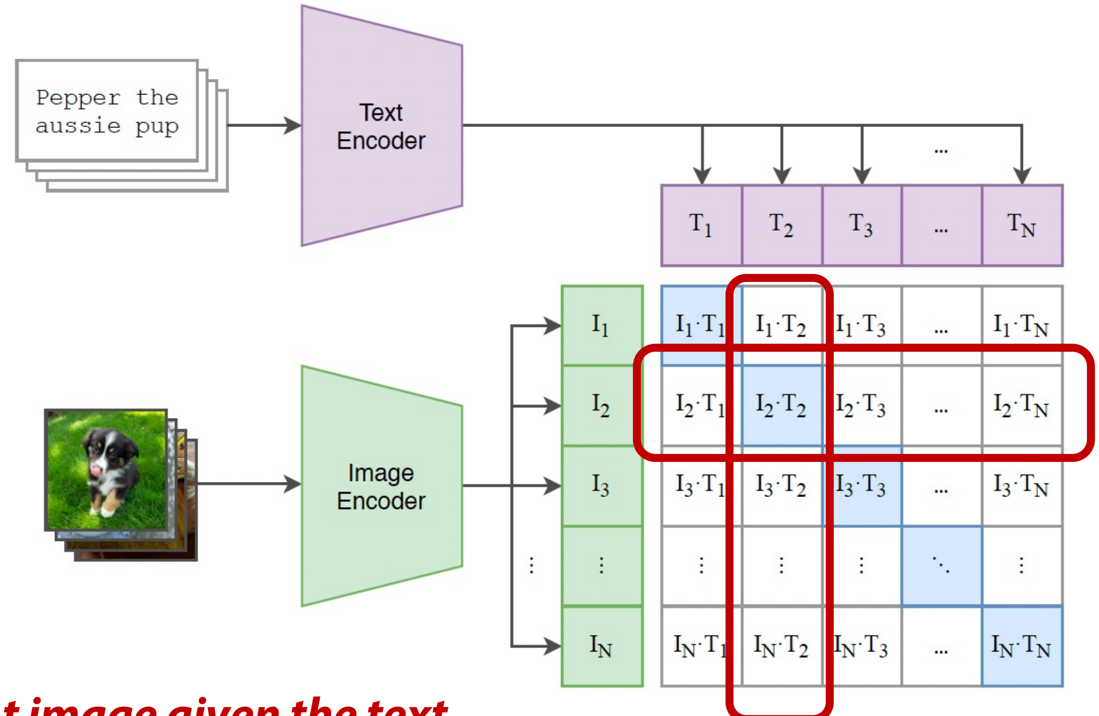


Cross-entropy on dimension 1

Maximize the diagonal and minimize the others

Cross-entropy on dimension 2

Method: Contrastive Pre-training



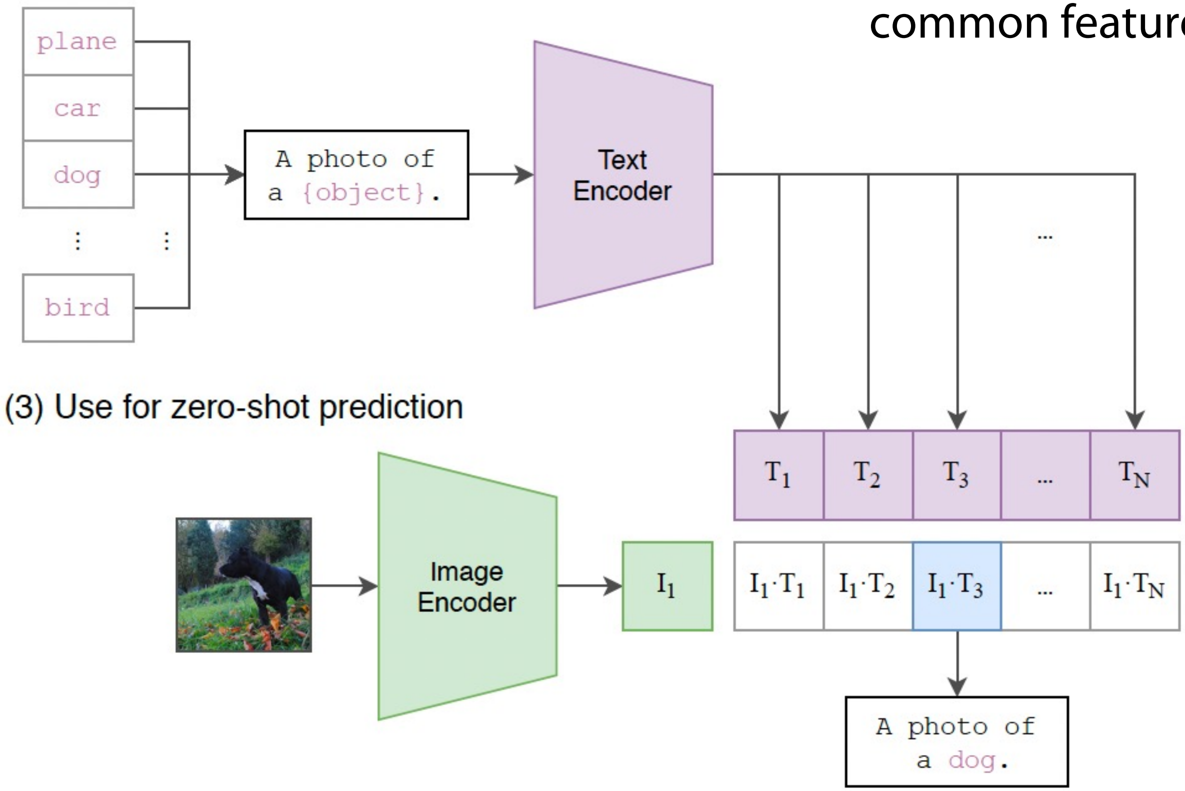
Cross-entropy on dimension 1

Cross-entropy on dimension 2

Pick the right image given the text
&
Pick the right text given the image

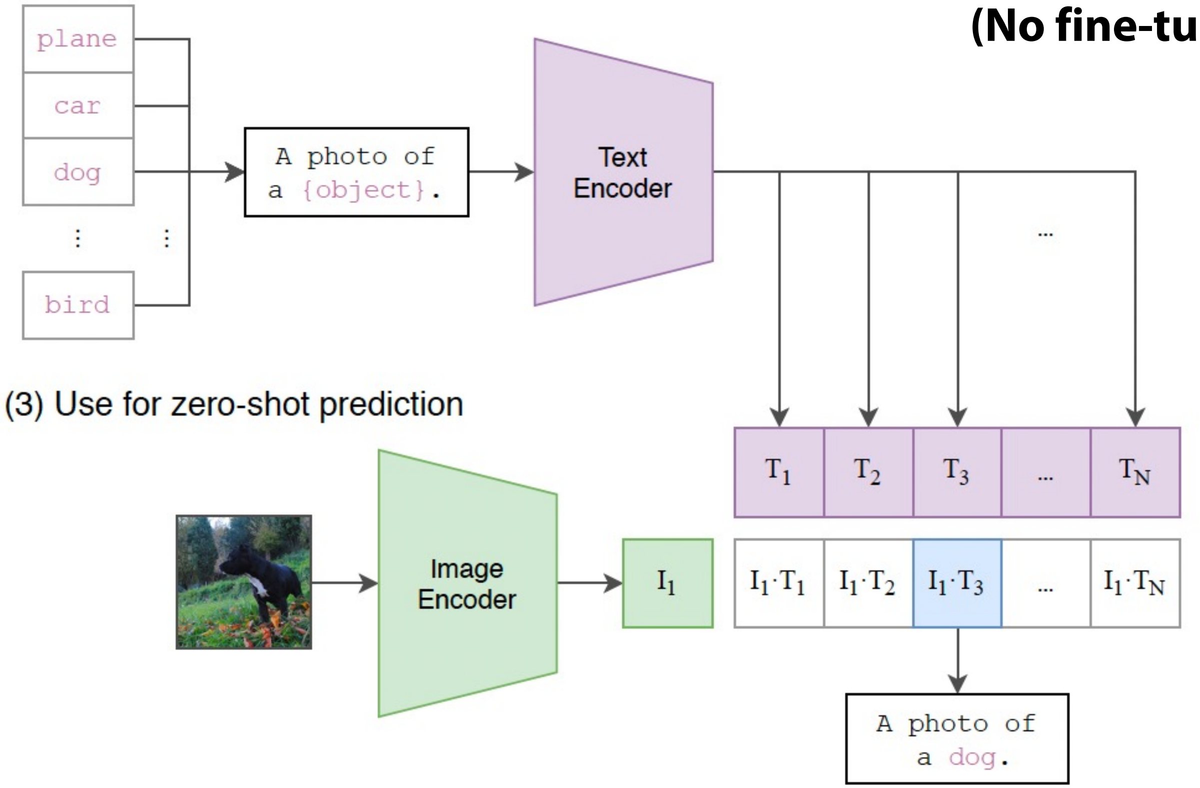
Method: Zero-Shot Testing

Core:
Images and text have been mapped into a common feature space



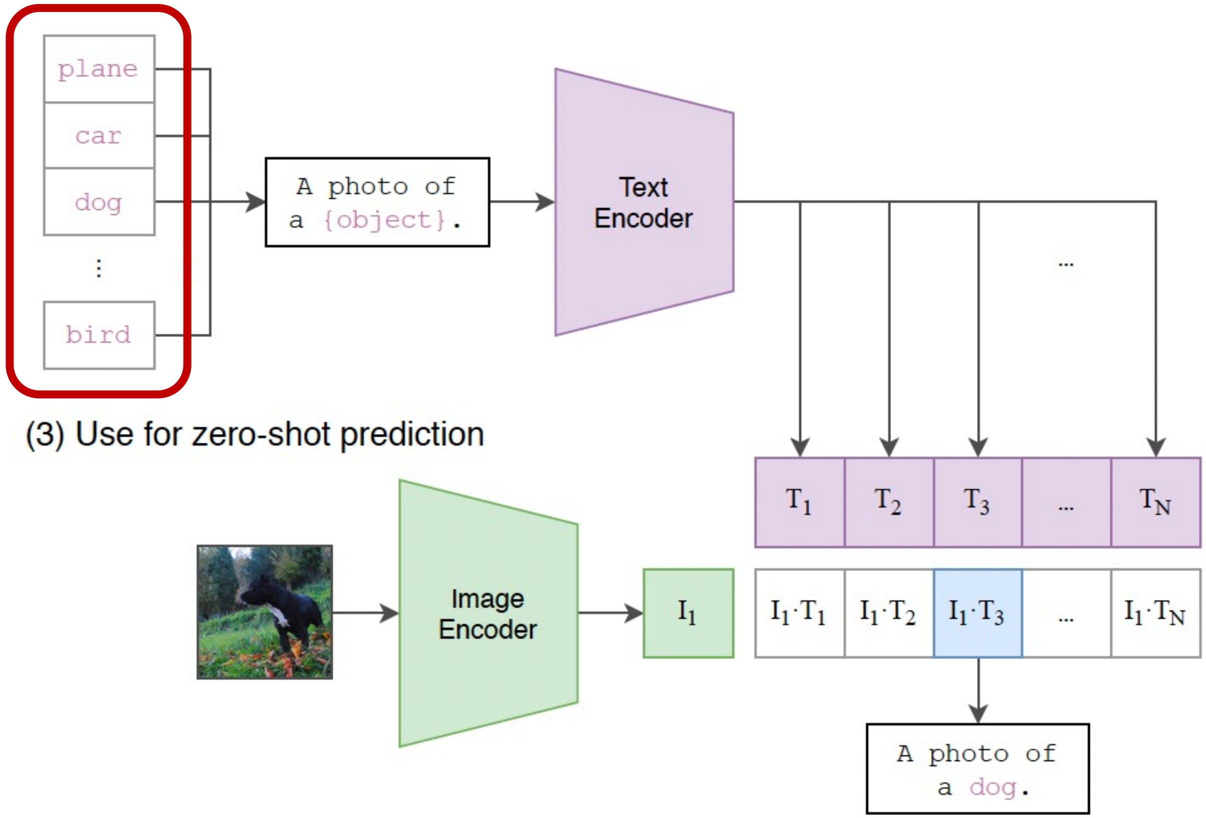
Method: Zero-Shot Testing

The classes are not pre-defined but chosen on demand (No fine-tuning)

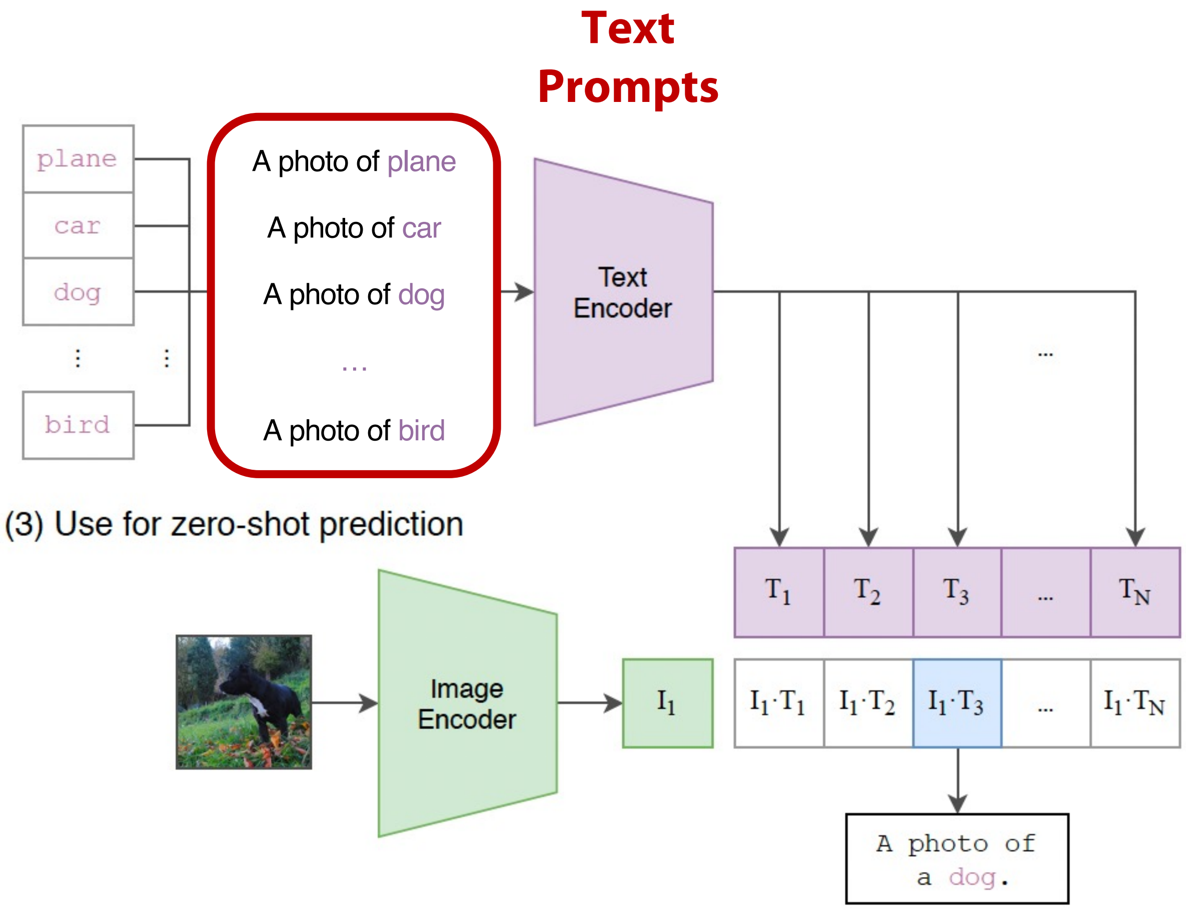


Method: Zero-Shot Testing

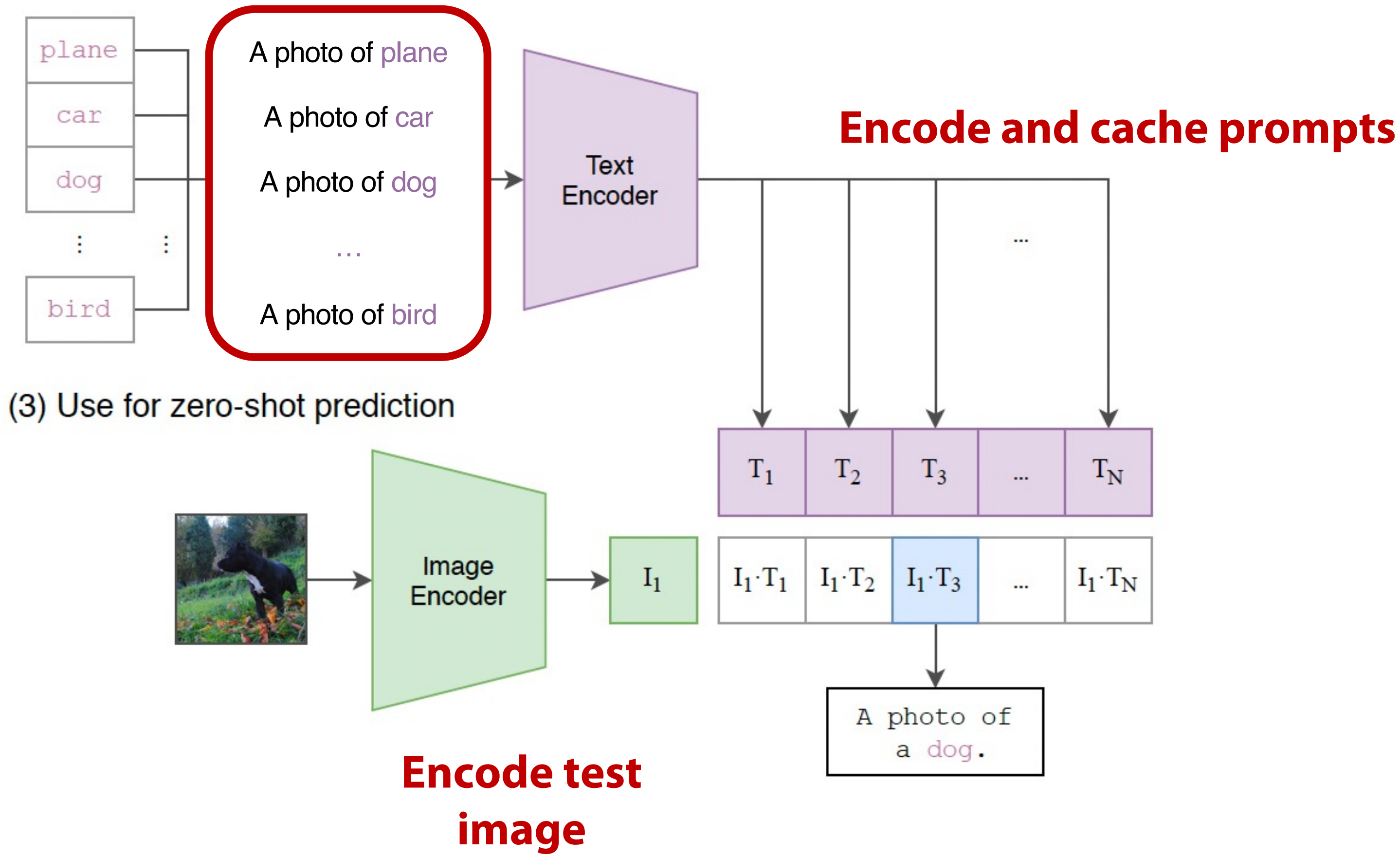
Classes on demand



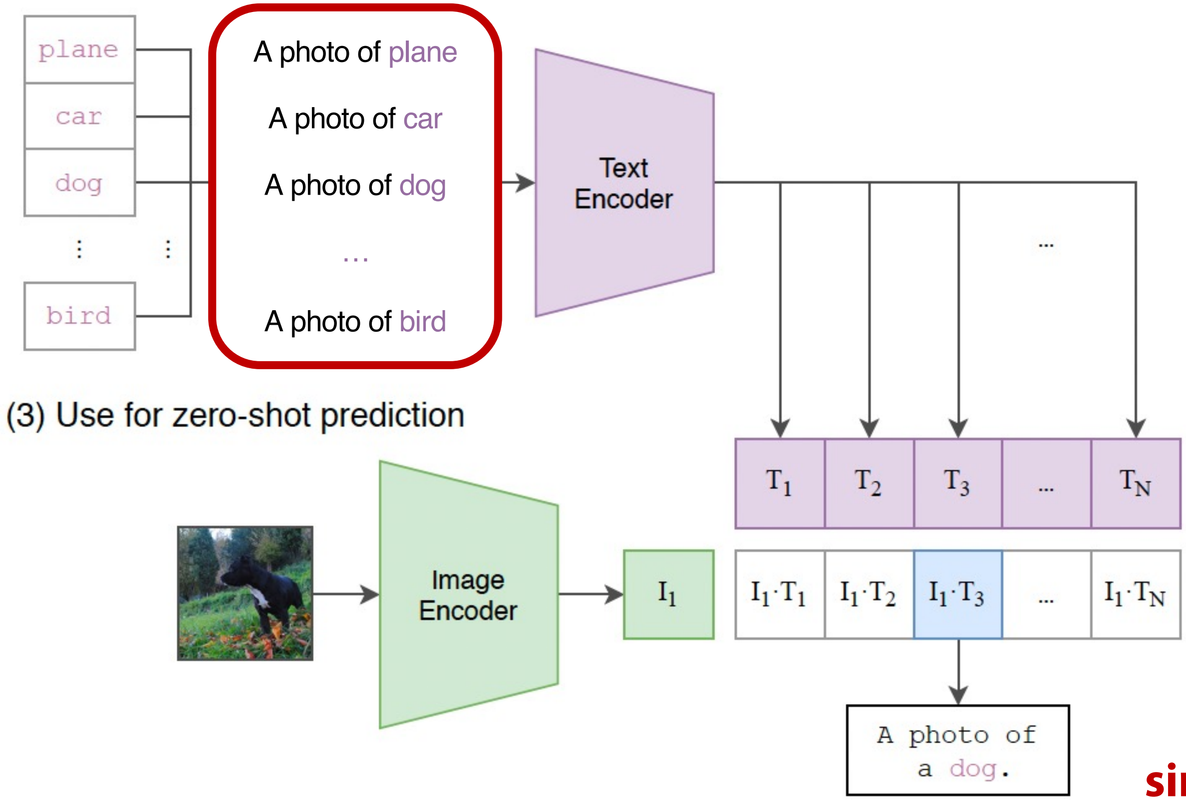
Method: Zero-Shot Testing



Method: Zero-Shot Testing

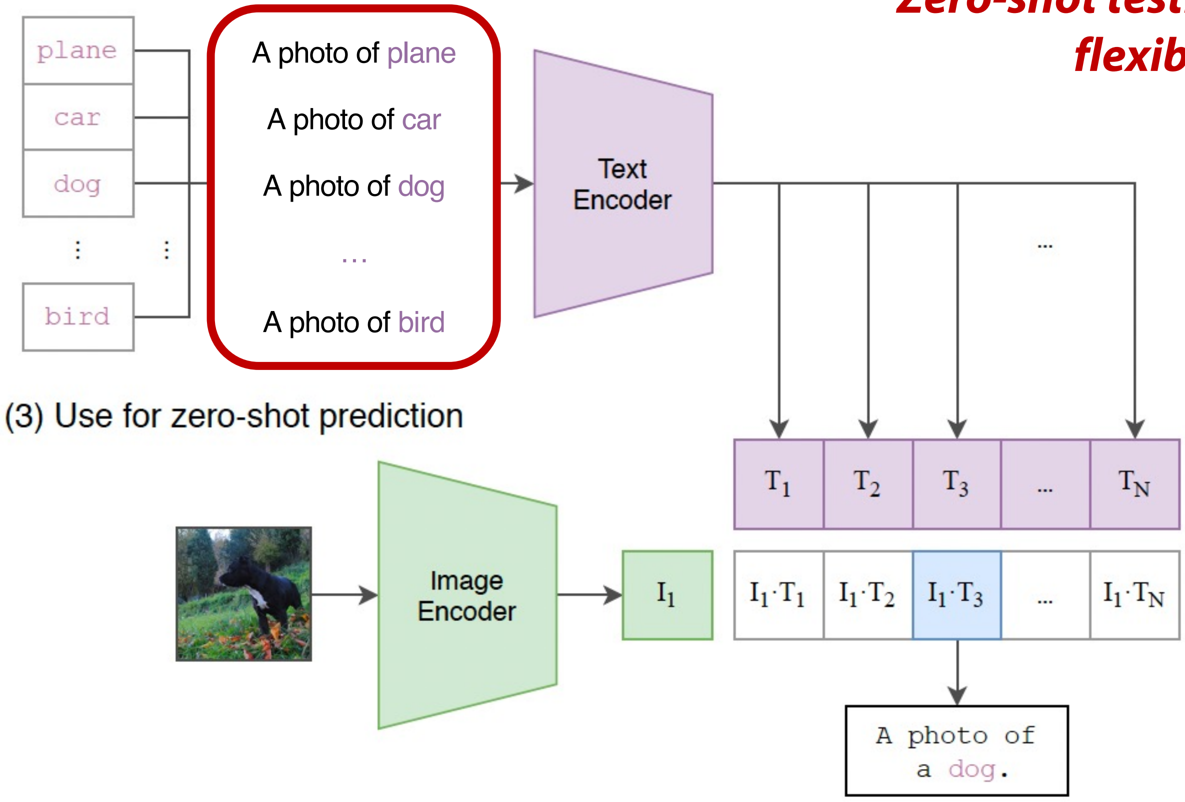


Method: Zero-Shot Testing



Compute similarities and pick the best class

Method: Zero-Shot Testing



Zero-shot testing is super flexible!

Method: Zero-Shot Testing – Prompt Engineering

Class names as baseline prompts

Problematic:

- A single word is often ambiguous, *i.e.*, the ***dog 'boxer'*** and the ***athlete 'boxer'***
- *It is rare on the web that a image is paired with a single word*

Method: Zero-Shot Testing – Prompt Engineering

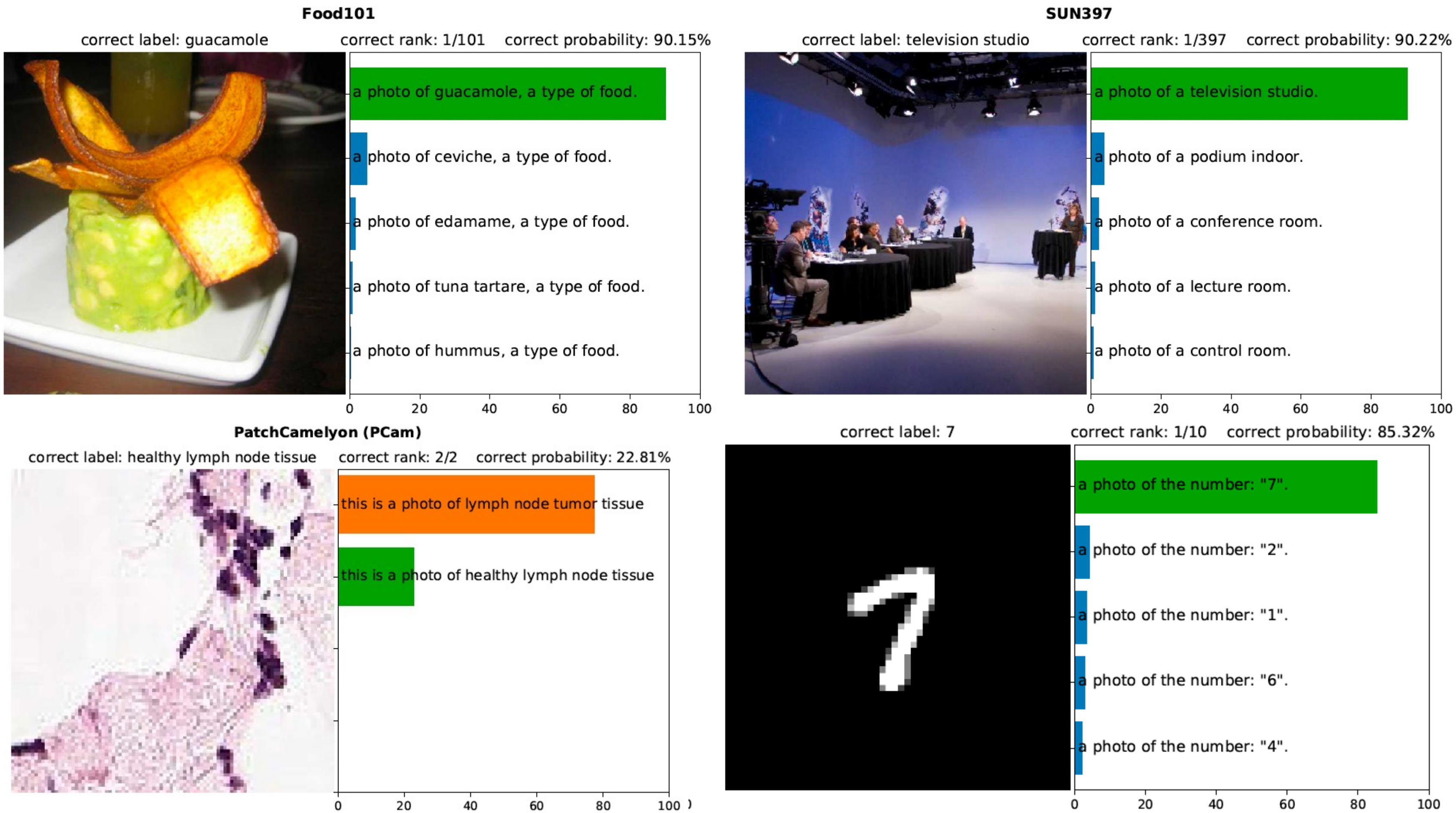
Class names as baseline prompts

Problematic:

- A single word is often ambiguous, *i.e.*, the ***dog 'boxer'*** and the ***athlete 'boxer'***
- *It is rare on the web that a image is paired with a single word*

Prompt engineering examples:

`A photo of a {label}.`	(For general classification)
`This is a {label}.`	(For general classification)
`A photo of a {label}, a type of pet.`	(For pet classification)
`A photo of a {label}, a type of food.`	(For food classification)
`A satellite photo of a {label}.`	(For satellite image classification)
`A digit "{label}".`	(For digit classification)


Food101
correct label: guacamole
correct rank: 1/101
correct probability: 90.15%
a photo of guacamole, a type of food.
a photo of ceviche, a type of food.
a photo of edamame, a type of food.
a photo of tuna tartare, a type of food.
a photo of hummus, a type of food.
0
20
40
60
80
100
SUN397
correct label: television studio
correct rank: 1/397
correct probability: 90.22%
a photo of a television studio.
a photo of a podium indoor.
a photo of a conference room.
a photo of a lecture room.
a photo of a control room.
0
20
40
60
80
100
PatchCamelyon (PCam)
correct label: healthy lymph node tissue
correct rank: 2/2
correct probability: 22.81%
this is a photo of lymph node tumor tissue
this is a photo of healthy lymph node tissue
0
20
40
60
80
100
correct label: 7
correct rank: 1/10
correct probability: 85.32%
a photo of the number: "7".
a photo of the number: "2".
a photo of the number: "1".
a photo of the number: "6".
a photo of the number: "4".
0
20
40
60
80
100

Method: Zero-Shot Testing – Prompt Engineering

Class names as baseline prompts

Problematic:

- A single word is often ambiguous, *i.e.*, the ***dog 'boxer'*** and the ***athlete 'boxer'***
- *It is rare on the web that a image is paired with a single word*

Prompt engineering examples:

`A photo of a {label}.`	(For general classification)
`This is a {label}.`	(For general classification)
`A photo of a {label}, a type of pet.`	(For pet classification)
`A photo of a {label}, a type of food.`	(For food classification)
`A satellite photo of a {label}.`	(For satellite image classification)
`A digit "{label}".`	(For digit classification)

Prompt ensemble examples (average the prompt features)**:**

```
A photo of a {label}.
A photo of a small {label}.
A photo of a big {label}.
```

(This could match the object no matter its size)

Method: Zero-Shot Testing – Prompt Engineering

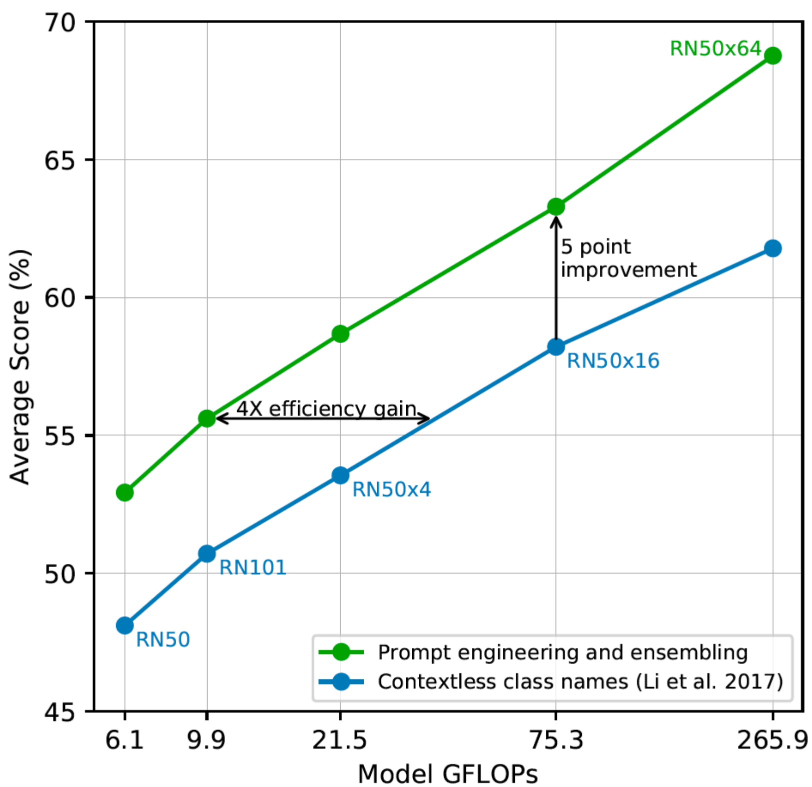
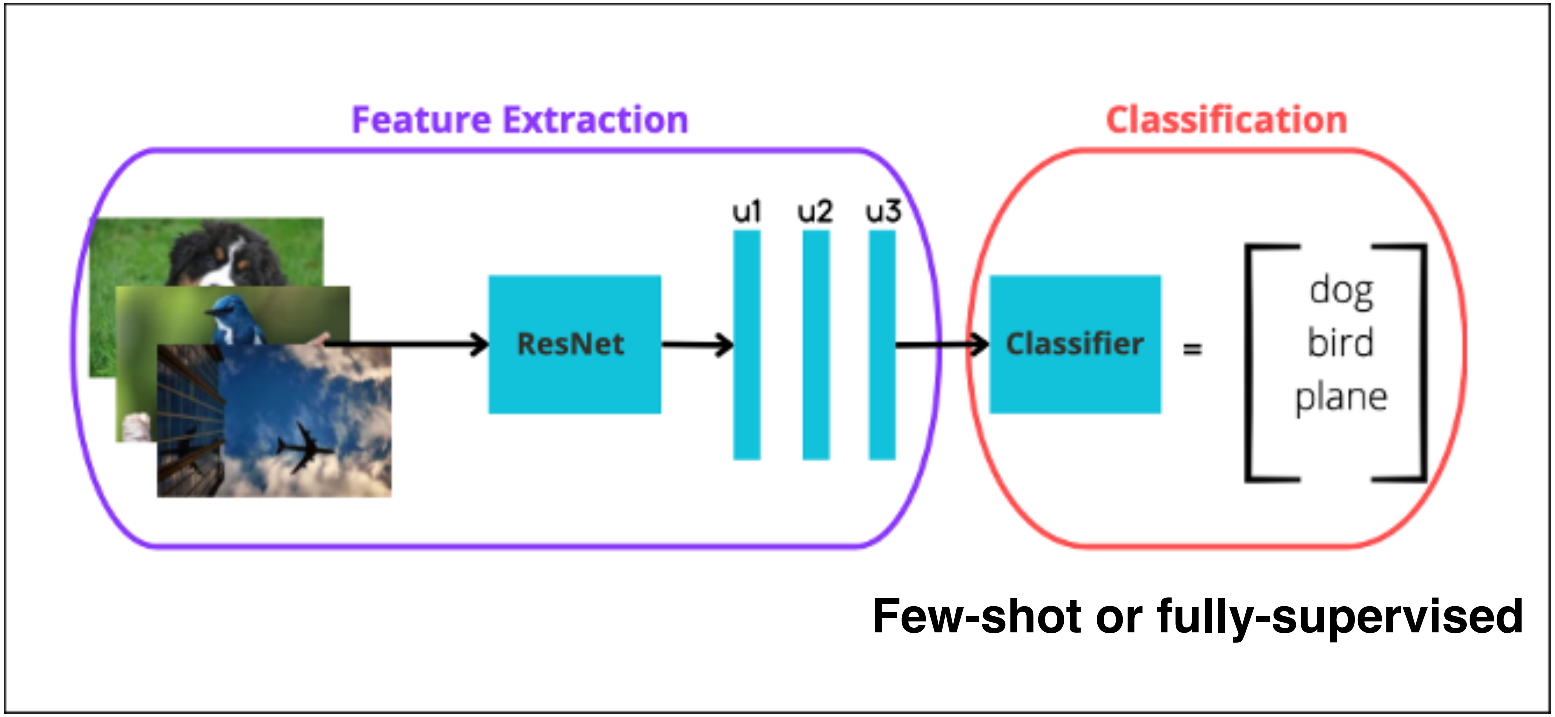


Figure 4. **Prompt engineering and ensembling improve zero-shot performance.**

Compare with decidated image classifier?

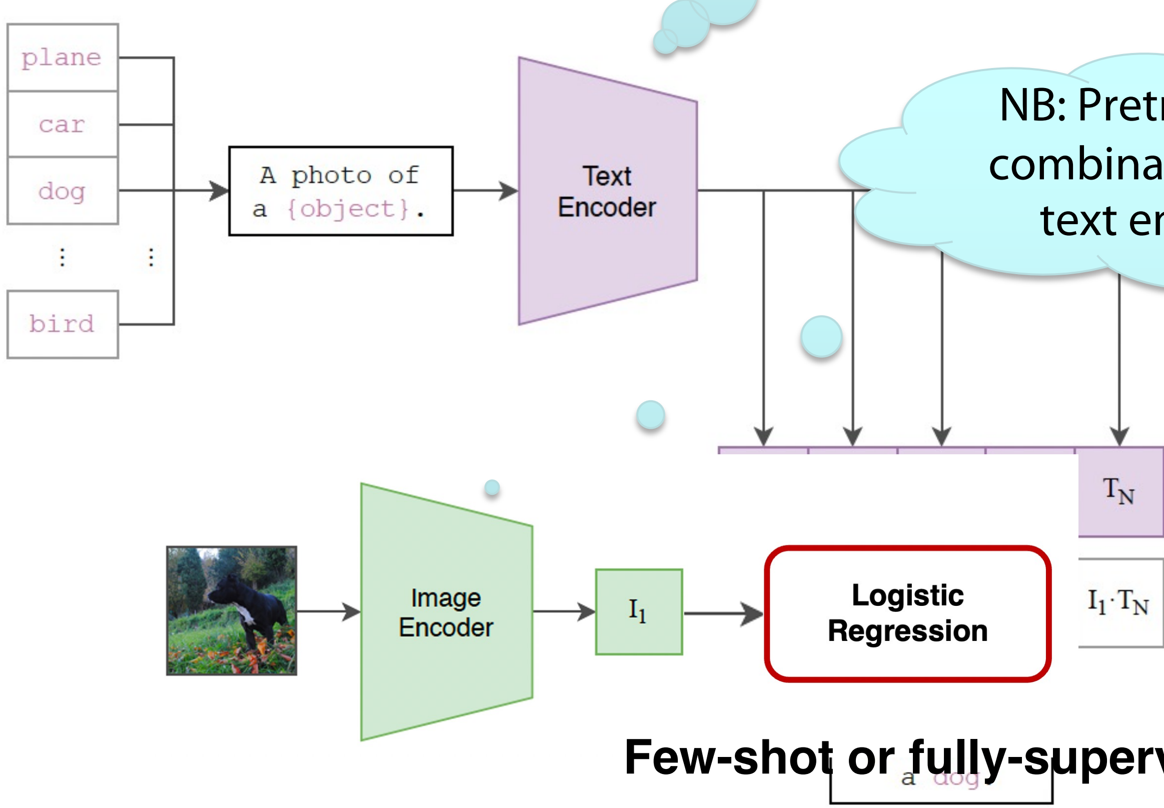


- For training, class labels must be known beforehand
- Using an image extractor paired with a classifier is also known as **linear probe evaluation**

Linear Probe CLIP

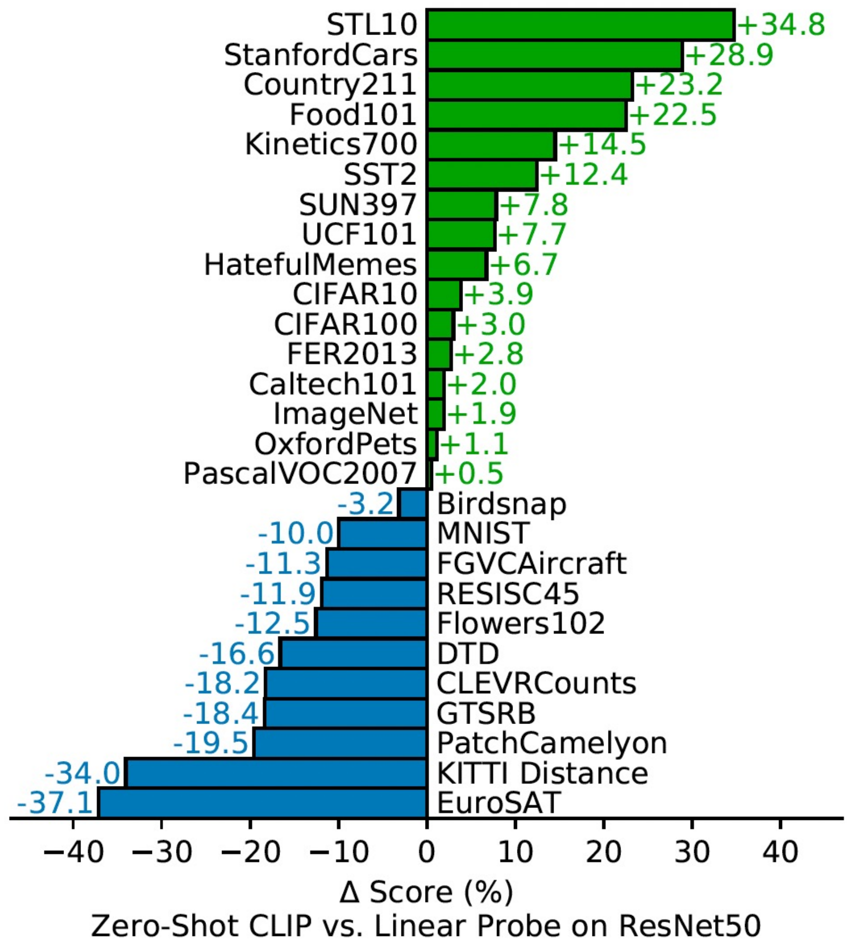
Use only the **CLIP's** *Image Encoder* to get the image features and fed them into a linear classifier. Even with this setup, **CLIP's** few-shot-learning capabilities are outstanding.

NB: Pretrained in combination with text encoder



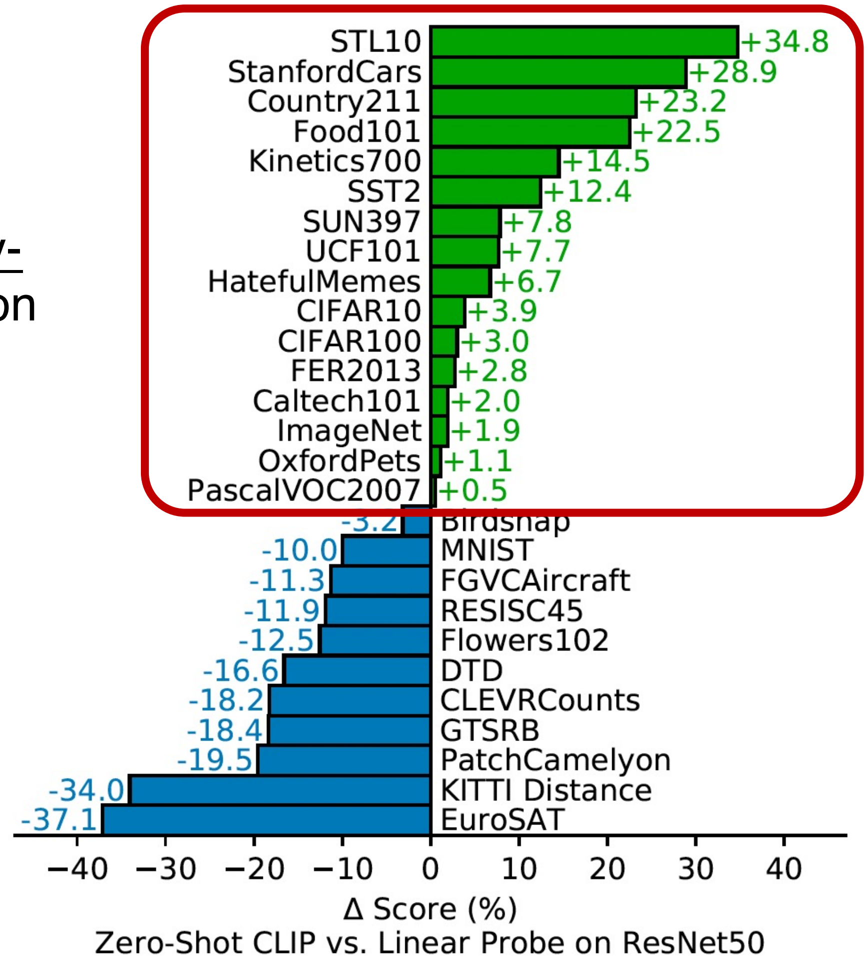
Few-shot or fully-supervised

Experiments: Zero-shot



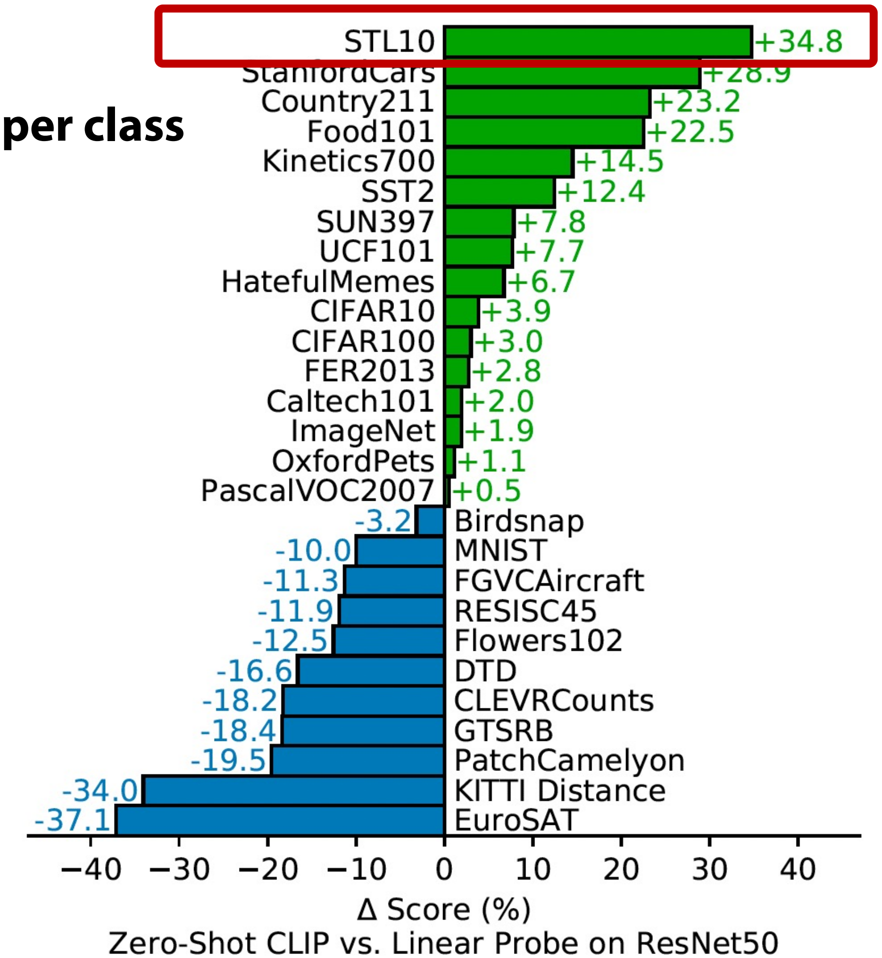
Experiments: Zero-shot

Zero-shot CLIP outperforms fully-supervised ResNet linear probe on 16 datasets



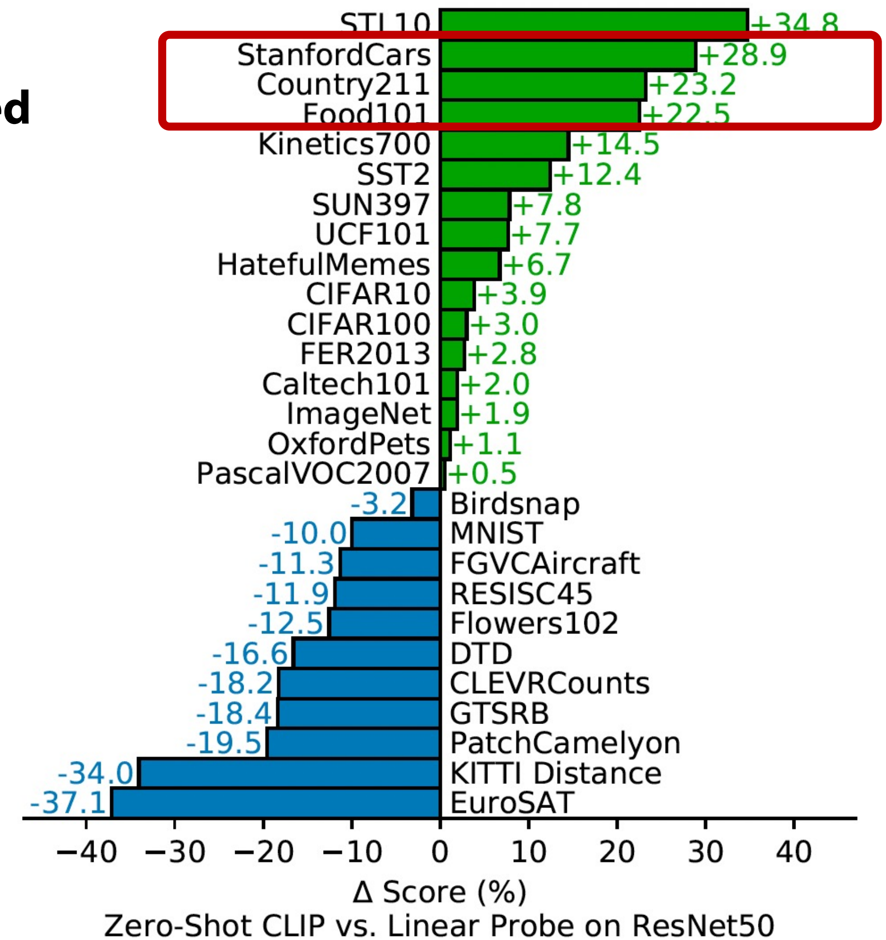
Experiments: Zero-shot

Limited examples per class



Experiments: Zero-shot

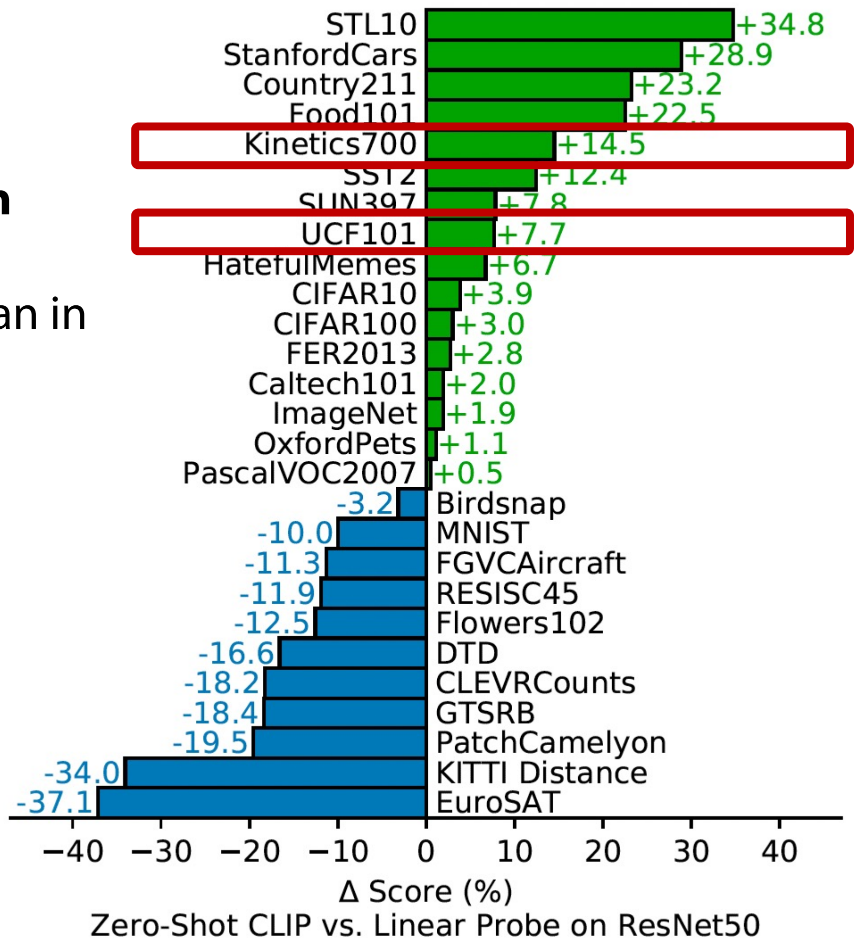
Fine-grained



Experiments: Zero-shot

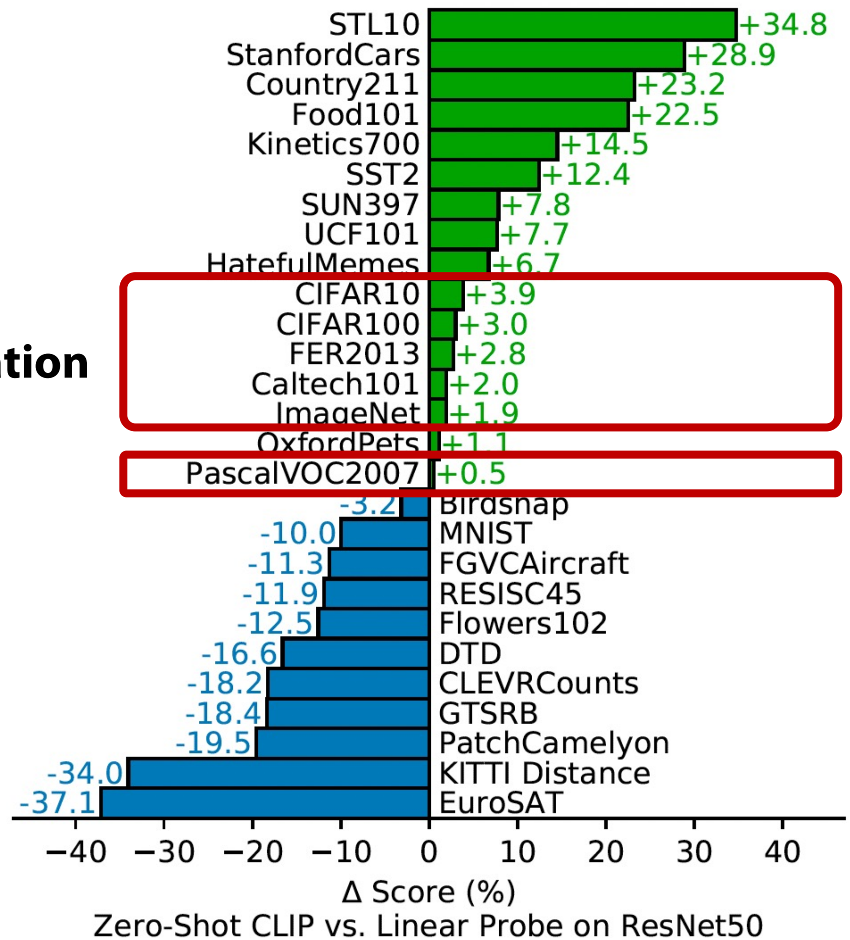
Action recognition

(More verbs on web than in ImageNet)



Experiments: Zero-shot

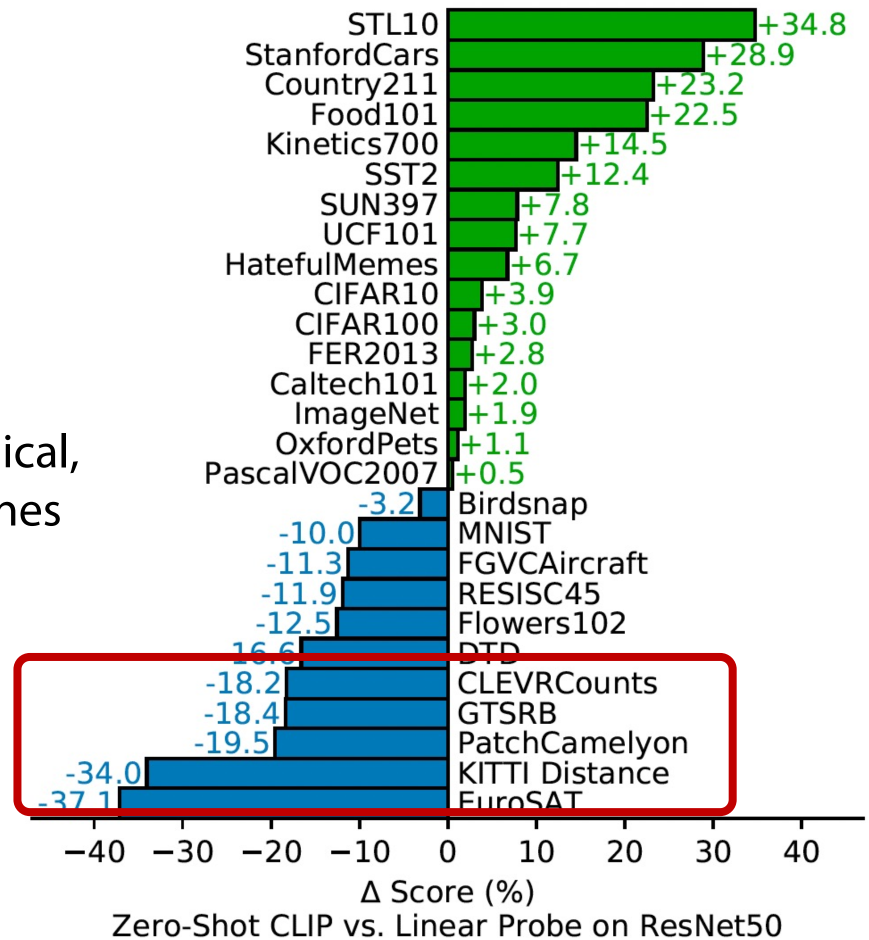
General classification



Experiments: Zero-shot

Specialized: satellite, medical, self-driving, synthetic scenes

(Rare on web)



Experiments: Zero-shot

Still large room for zero-shot CLIP

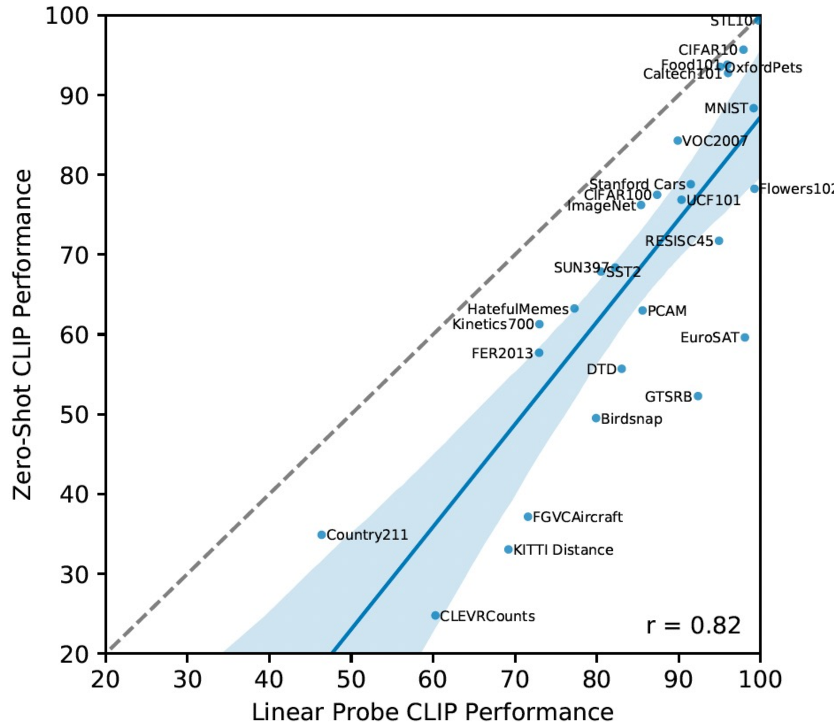


Figure 8. **Zero-shot performance is correlated with linear probe performance but still mostly sub-optimal.**

Experiments: Zero-shot

More compute power could help

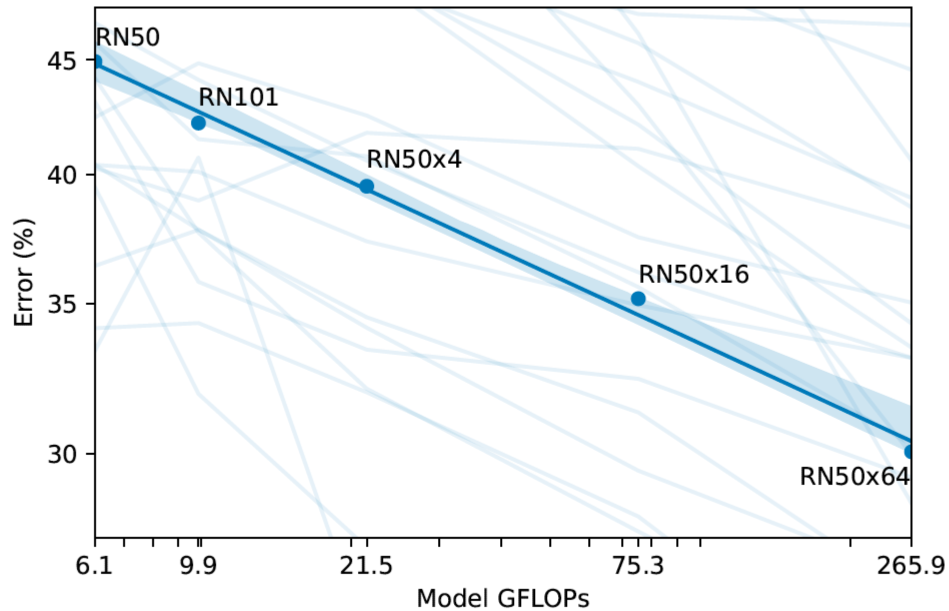


Figure 9. **Zero-shot CLIP performance scales smoothly as a function of model compute power.**

Experiments: Few-shot

Zero-shot CLIP = 4-shot Linear CLIP

Few-shot Linear CLIP > Others

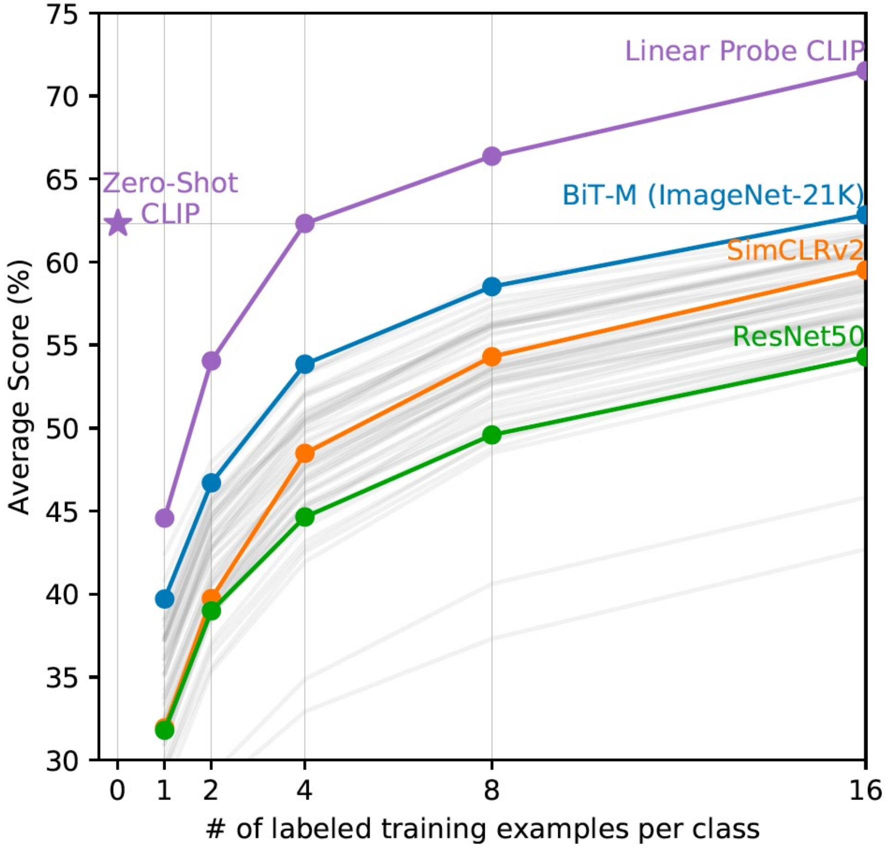


Figure 6. **Zero-shot CLIP outperforms few-shot linear probes.**

Experiments: Linear probe

Linear probe CLIP is STOA

Event better on more diverse datasets

Transformer is better than ConvNet with enough data

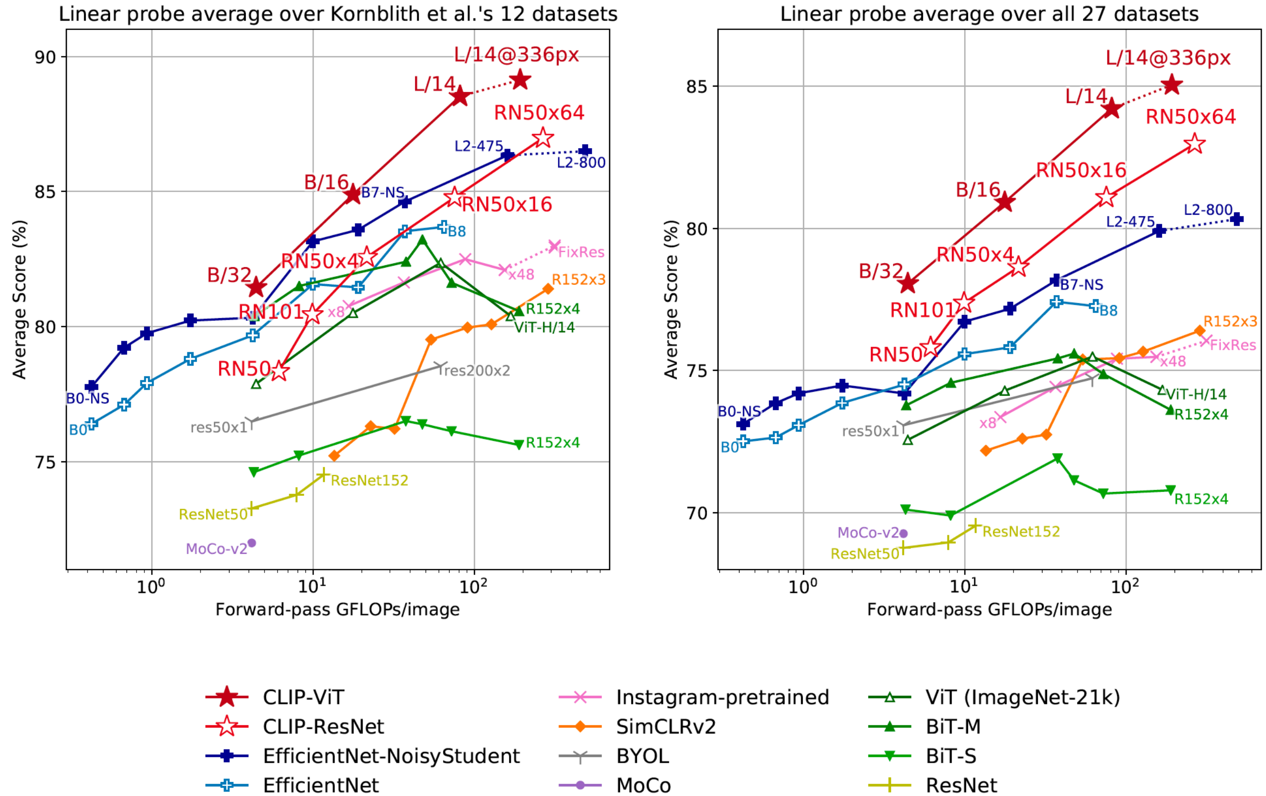


Figure 10. **Linear probe performance of CLIP models in comparison with state-of-the-art computer vision models**, including

ImageNet-like datasets **More diverse datasets**

Experiments: CLIP is more robust to domain shift

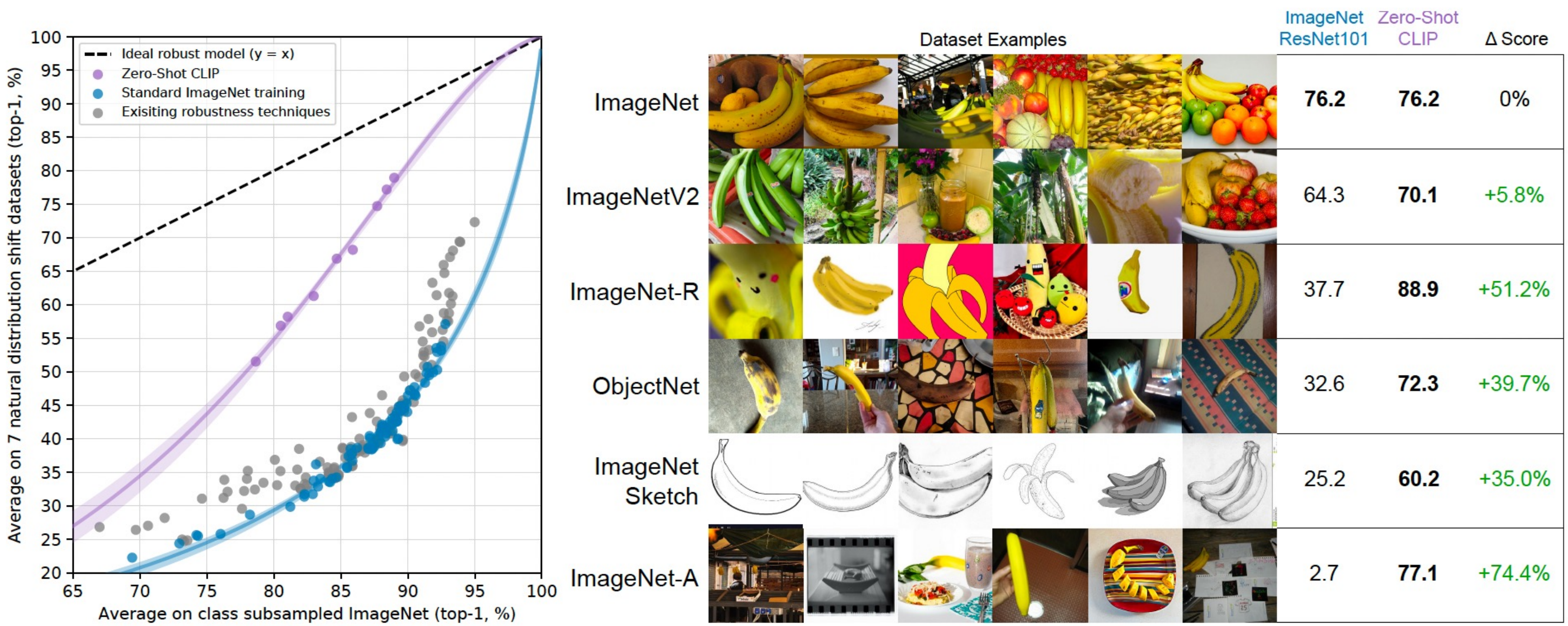


Figure 13. **Zero-shot CLIP is much more robust to distribution shift than standard ImageNet models.** (Left) An ideal robust model

Experiments: CLIP is more robust to domain shift

Semantically similar datasets
in similar or distinct domains

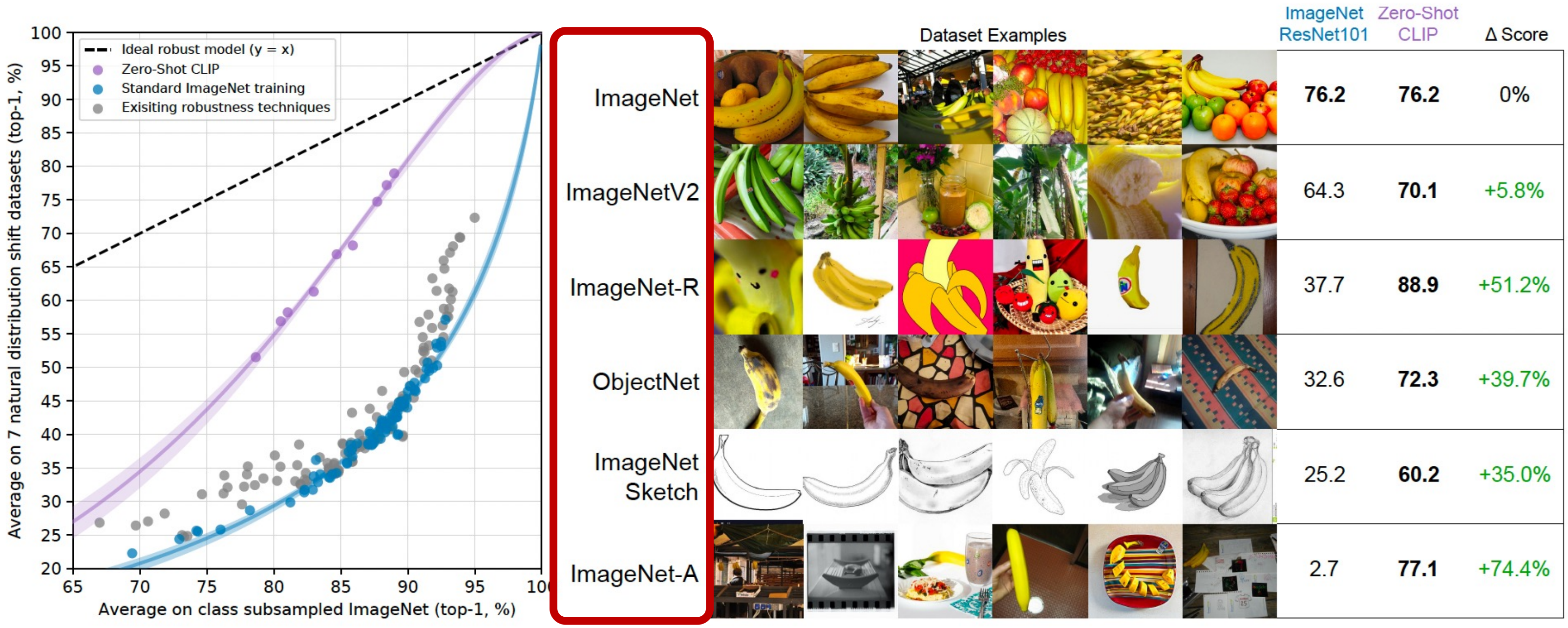
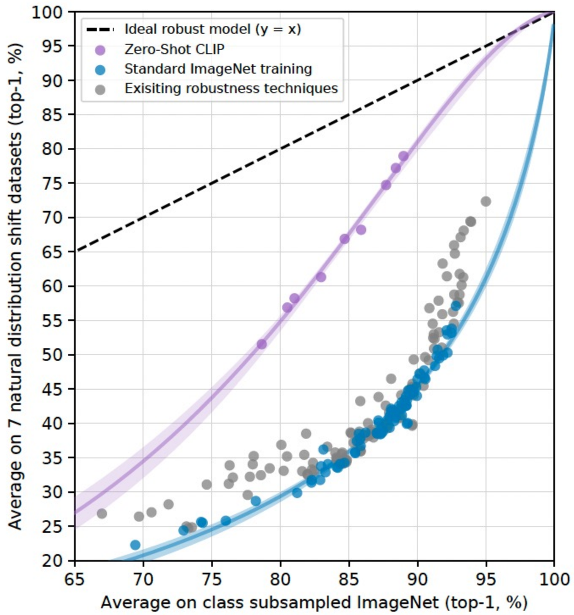
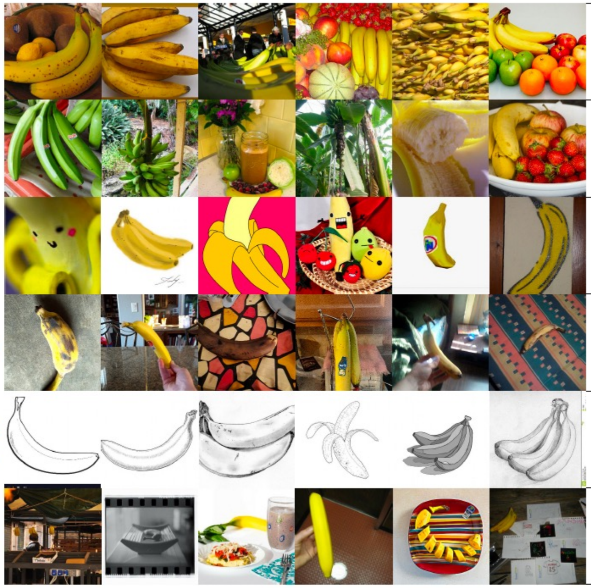


Figure 13. **Zero-shot CLIP is much more robust to distribution shift than standard ImageNet models.** (Left) An ideal robust model

Experiments: CLIP is more robust to domain shift



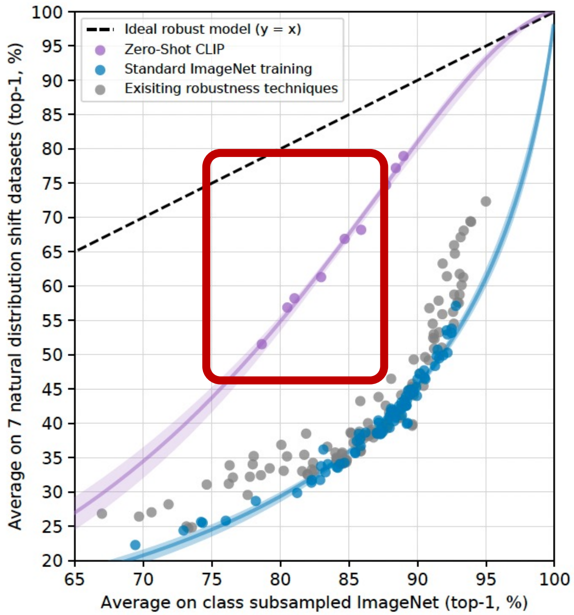


Dataset	ImageNe ResNet10	Zero-Shot CLIP	Δ Score
ImageNet	**76.2**	**76.2**	0%
ImageNetV2	64.3	**70.1**	+5.8%
ImageNet-R	37.7	**88.9**	+51.2%
ObjectNet	32.6	**72.3**	+39.7%
ImageNet Sketch	25.2	**60.2**	+35.0%
ImageNet-A	2.7	**77.1**	+74.4%

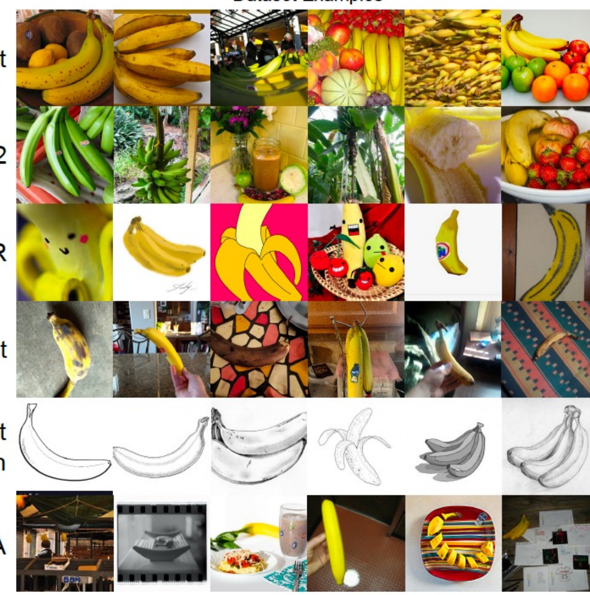
Figure 13. **Zero-shot CLIP is much more robust to distribution shift than standard ImageNet models.** (Left) An ideal robust model

Zero-shot CLIP is robust

Experiments: CLIP is more robust to domain shift



Dataset Examples



	ImageNet ResNet101	Zero-Shot 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%

Figure 13. **Zero-shot CLIP is much more robust to distribution shift than standard ImageNet models.** (Left) An ideal robust model

Zero-shot CLIP is robust

Code Released

```
import torch
import clip
from PIL import Image

device = "cuda" if torch.cuda.is_available() else "cpu"
model, preprocess = clip.load("ViT-B/32", device=device)

image = preprocess(Image.open("CLIP.png")).unsqueeze(0).to(device)
text = clip.tokenize(["a diagram", "a dog", "a cat"]).to(device)

with torch.no_grad():
    image_features = model.encode_image(image)
    text_features = model.encode_text(text)

    logits_per_image, logits_per_text = model(image, text)
    probs = logits_per_image.softmax(dim=-1).cpu().numpy()

print("Label probs:", probs)  # prints: [[0.9927937  0.00421068 0.00299572]]
```

Code Released

```
import torch
import clip
from PIL import Image

device = "cuda" if torch.cuda.is_available() else "cpu"
model, preprocess = clip.load("ViT-B/32", device=device)

image = preprocess(Image.open("CLIP.png")).unsqueeze(0).to(device)
text = clip.tokenize(["a diagram", "a dog", "a cat"]).to(device)

with torch.no_grad():
    image_features = model.encode_image(image)
    text_features = model.encode_text(text)

    logits_per_image, logits_per_text = model(image, text)
    probs = logits_per_image.softmax(dim=-1).cpu().numpy()

print("Label probs:", probs)  # prints: [[0.9927937  0.00421068 0.00299572]]
```

Easy to get CLIP features

Conclusion CLIP

Multi-modal pre-training on a web scale gives STOA performances

Zero-shot may enable a new paradigm to develop vision systems

- No data annotation, model training, hyper-parameter tuning is needed
- Only '`import clip`' and design the prompts
- Especially for non-specialized tasks
- At least, CLIP features are more accurate and robust than ResNet features

Images and languages are mapped into a common space

- This is how human understand concepts
- Towards general intelligence
- But currently, more like a super fuzzy reverse search engine

Easy to use:

- Released codes and models
- Unreleased data and prompts