AI504: Programming for Artificial Intelligence

Week 15: Image-Text Multimodal Learning

Edward Choi

Grad School of AI

edwardchoi@kaist.ac.kr

Today’s Topic

- Image-to-text (a.k.a Image captioning)
  - Show and Tell
  - Show, Attend and Tell
- Text-to-Image
  - Text-conditioned GAN
  - DALL-E
  - CLIP + DALL-E 2
- Image-text pretraining
  - Pre-trained vision-language models

Image Captioning

Image-to-Text

- Sequence to sequence
  - Text in, text out
  - e.g. Translate French to English
- Image to sequence
  - Image in, text out
  - e.g. Describe a given image in text (i.e. Image Captioning)

Image Captioning

A person riding a motorcycle on a dirt road.

A group of young people playing a game of frisbee.

A herd of elephants walking across a dry grass field.

Encoder-Decoder Architecture

- Sequence to sequence
  - Encoder: RNN
  - Decoder: RNN
- Image to sequence
  - Encoder: ???
  - Decoder: ???

Encoder-Decoder Architecture

- Sequence to sequence
  - Encoder: RNN
  - Decoder: RNN
- Image to sequence
  - Encoder: CNN
  - Decoder: RNN

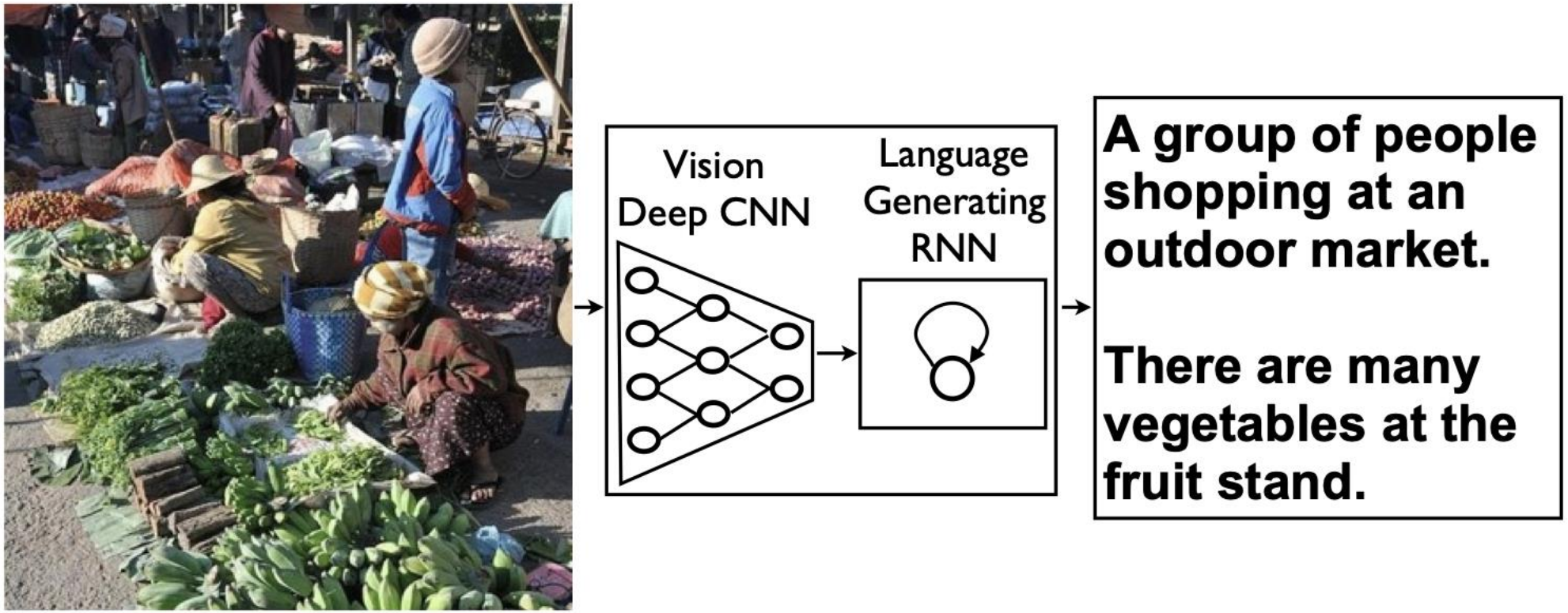
Show and Tell

Show and Tell

- Show and Tell: A Neural Image Caption Generator
  - Vinyals et al. CVPR 2015
- First paper to perform neural image captioning without any domain knowledge
  - No object detection, language modeling, description templates
  - Not text ranking, but pure generation
  - End-to-end training

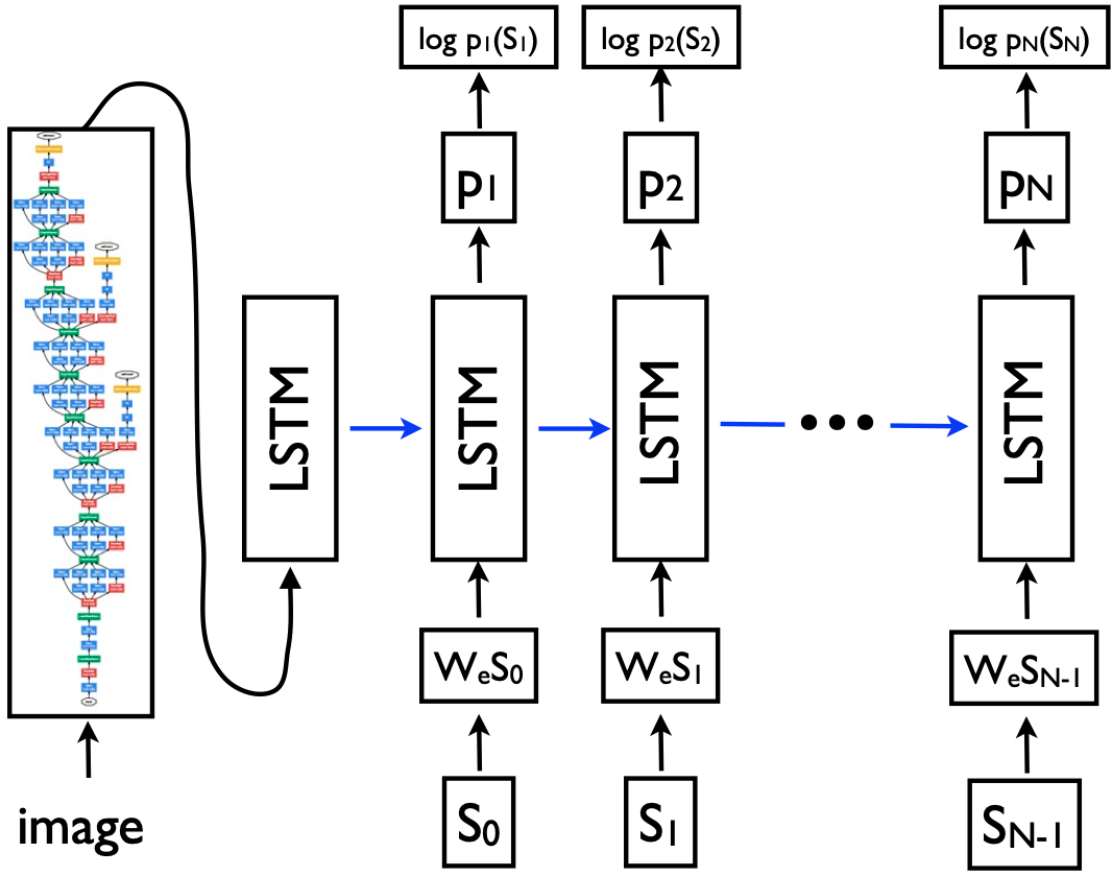
Show and Tell

- Very simple architecture



Show and Tell

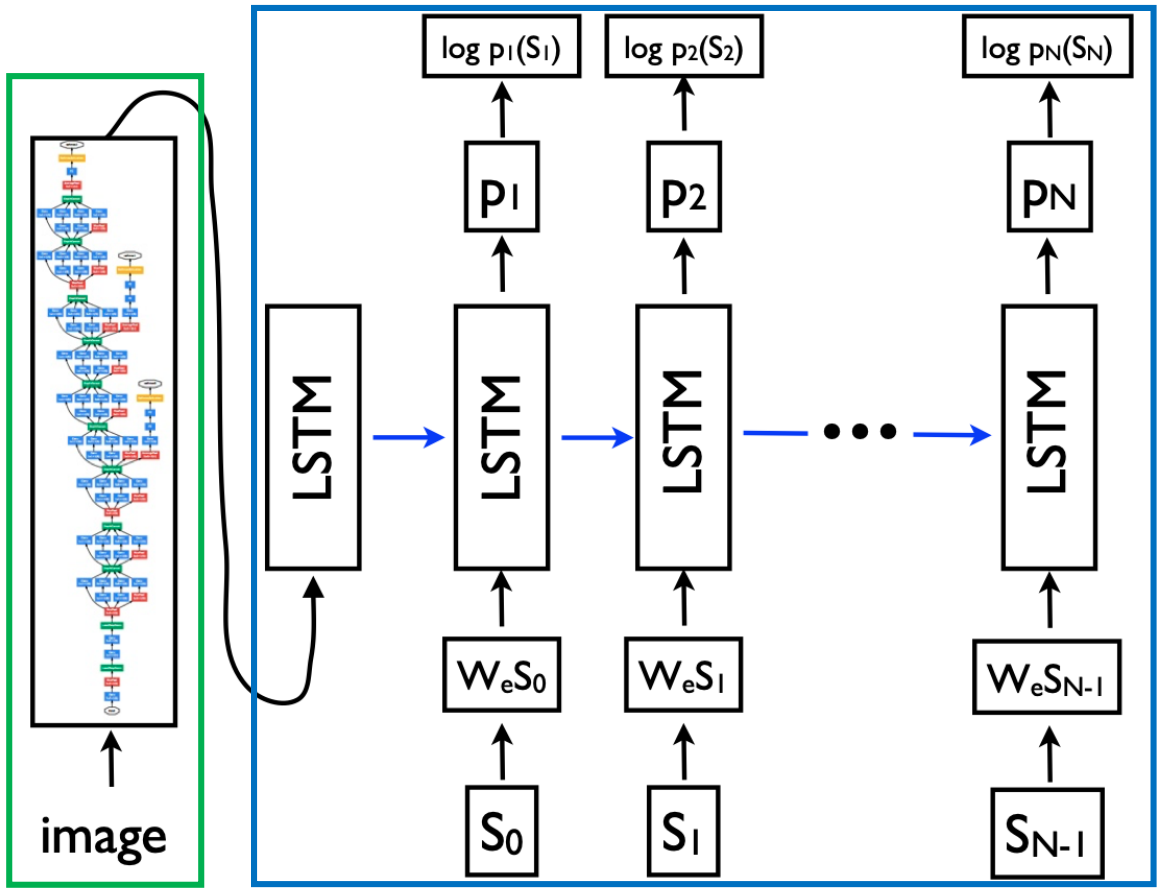
- A bit more detailed architecture depiction



Show and Tell

- A bit more detailed architecture depiction

Image Encoder (Inception v1 pretrained on ImageNet)



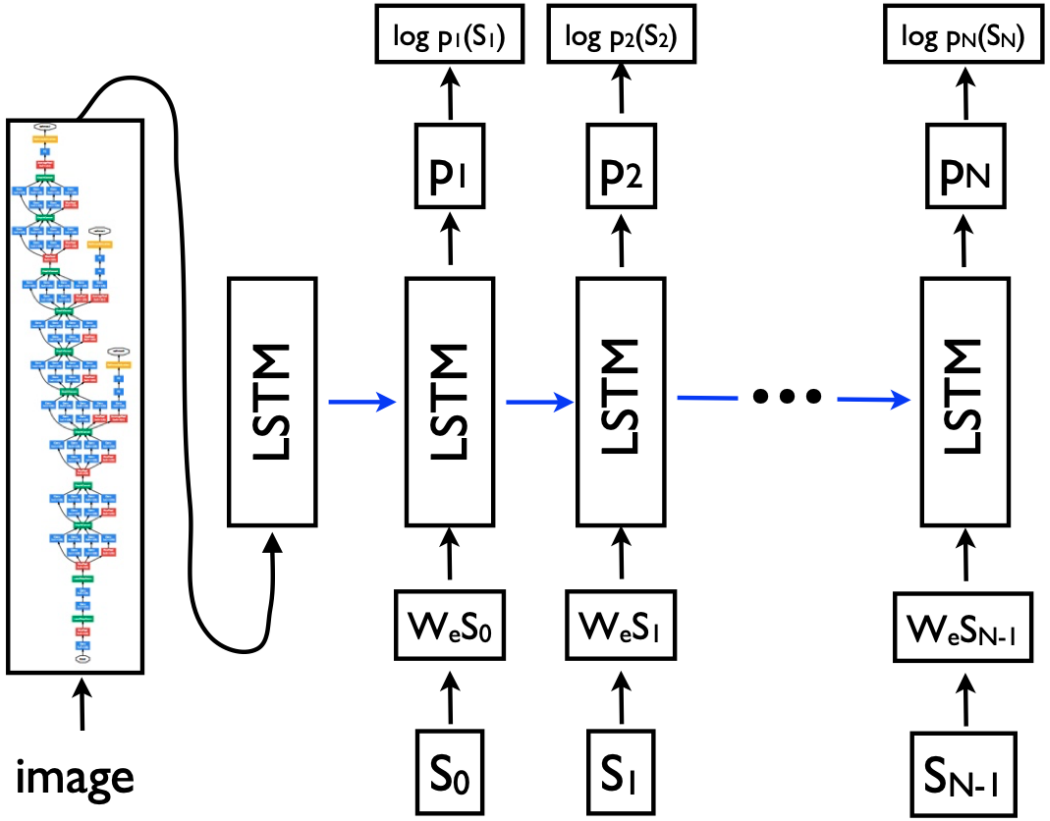
Text Decoder (LSTM)

Show and Tell

- Each S_i is predicted based on p_i
  - $S_i = \text{Softmax}(W_s p_i + b)$
- Each p_i is derived based on p_{i-1}, S_{i-1}
  - $p_i = \text{RNN}(p_{i-1}, W_e S_{i-1})$

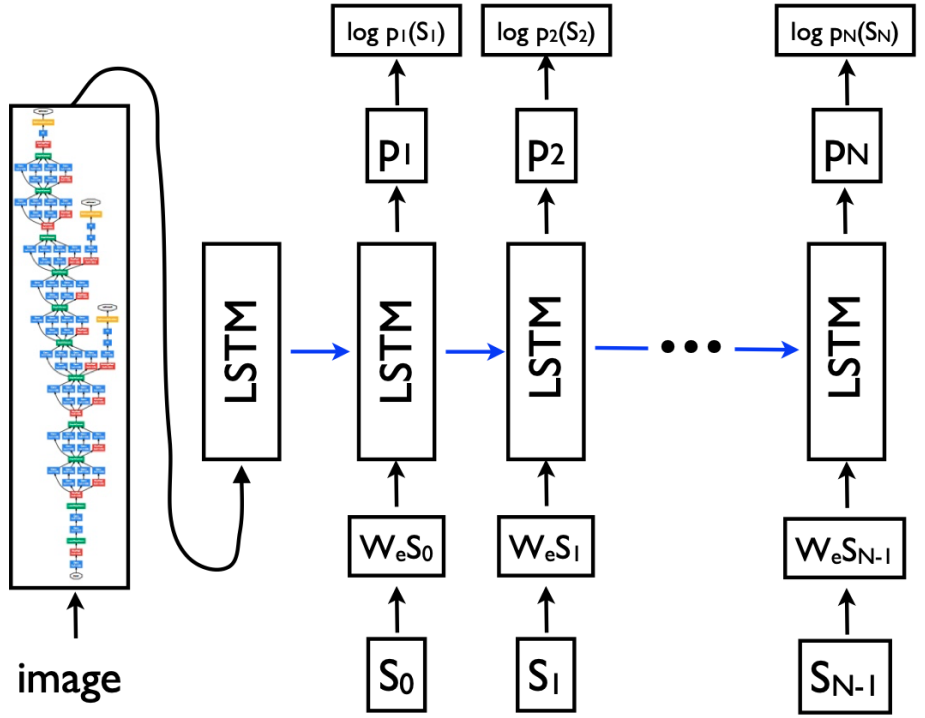
- W_e = Word embedding
- S_{-1} = CNN(Image)

- S_0:<START>, S_N:<END>



Show and Tell

- **Some technical details**
- 512 embedding size & RNN size
  - Output of CNN is also 512-dimensional
- Image embedding is “fed” into LSTM at time -1
  - Not used to initialized the LSTM hidden vector.
  - Hidden layers are probably initialized to 0
- Pretrained word embeddings didn’t help much
  - Specifically, Word2Vec
- Beam search is used with beam size 20
- Trained with negative log-likelihood



Popular Datasets

Dataset name	size		
	train	valid.	test
Pascal VOC 2008 [6]	-	-	1000
Flickr8k [26]	6000	1000	1000
Flickr30k [33]	28000	1000	1000
MSCOCO [20]	82783	40504	40775
SBU [24]	1M	-	-

Model Performance

Metric	BLEU-4	METEOR	CIDER
NIC	**27.7**	**23.7**	**85.5**
Random	4.6	9.0	5.1
Nearest Neighbor	9.9	15.7	36.5
Human	21.7	25.2	85.4

Table 1. Scores on the MSCOCO development set.

Approach	PASCAL (xfer)	Flickr 30k	Flickr 8k	SBU
Im2Text [24]				11
TreeTalk [18]				19
BabyTalk [16]	25			
Tri5Sem [11]			48	
m-RNN [21]		55	58	
MNLM [14][5]		56	51	
SOTA	25	56	58	19
NIC	**59**	**66**	**63**	**28**
Human	69	68	70	

Table 2. BLEU-1 scores. We only report previous work results when available. SOTA stands for the current state-of-the-art.

Evaluation Results (grouped by human rating)

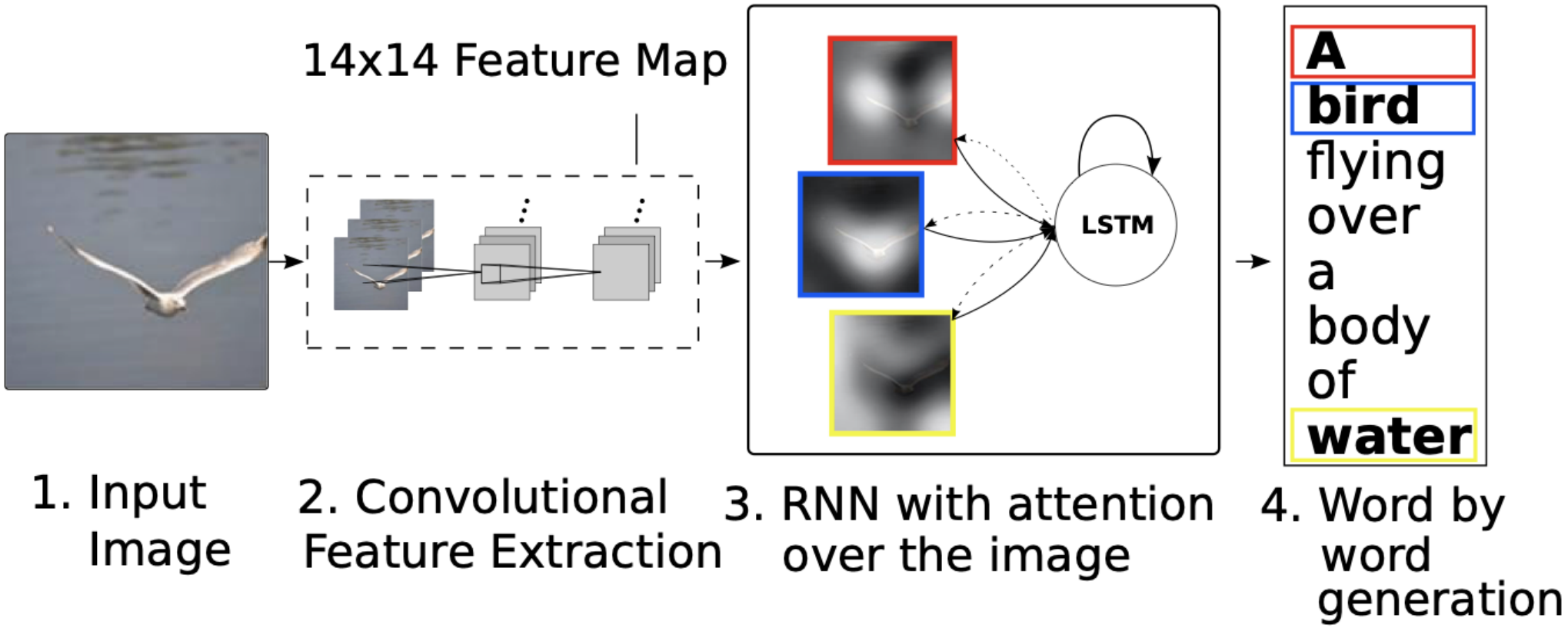
Show, Attend and Tell

Show, Attend and Tell

- Show, Attend and Tell: Neural Image Caption Generation with Visual Attention
  - Xu et al. ICML 2015
- Mixing attention mechanism with image captioning

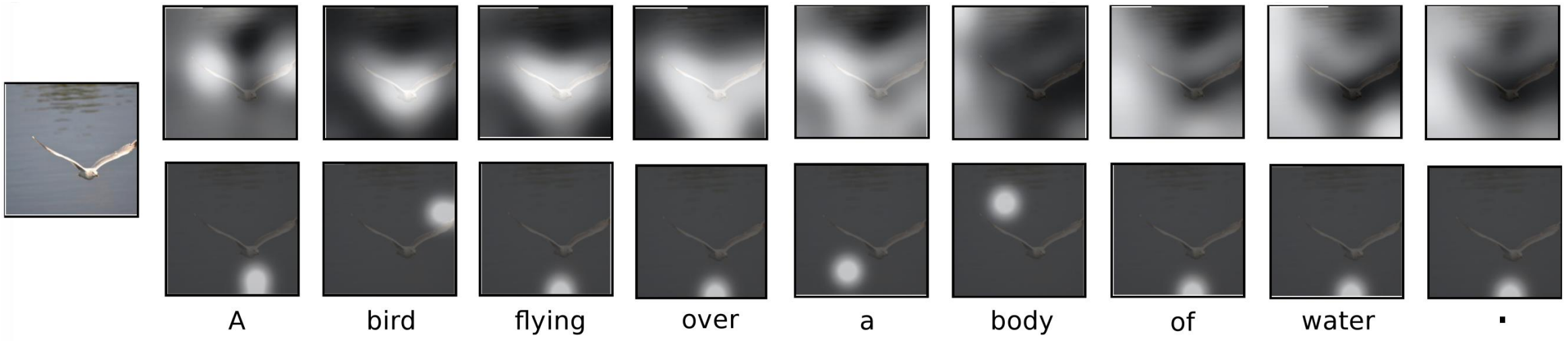
Show, Attend and Tell

- High-level architecture



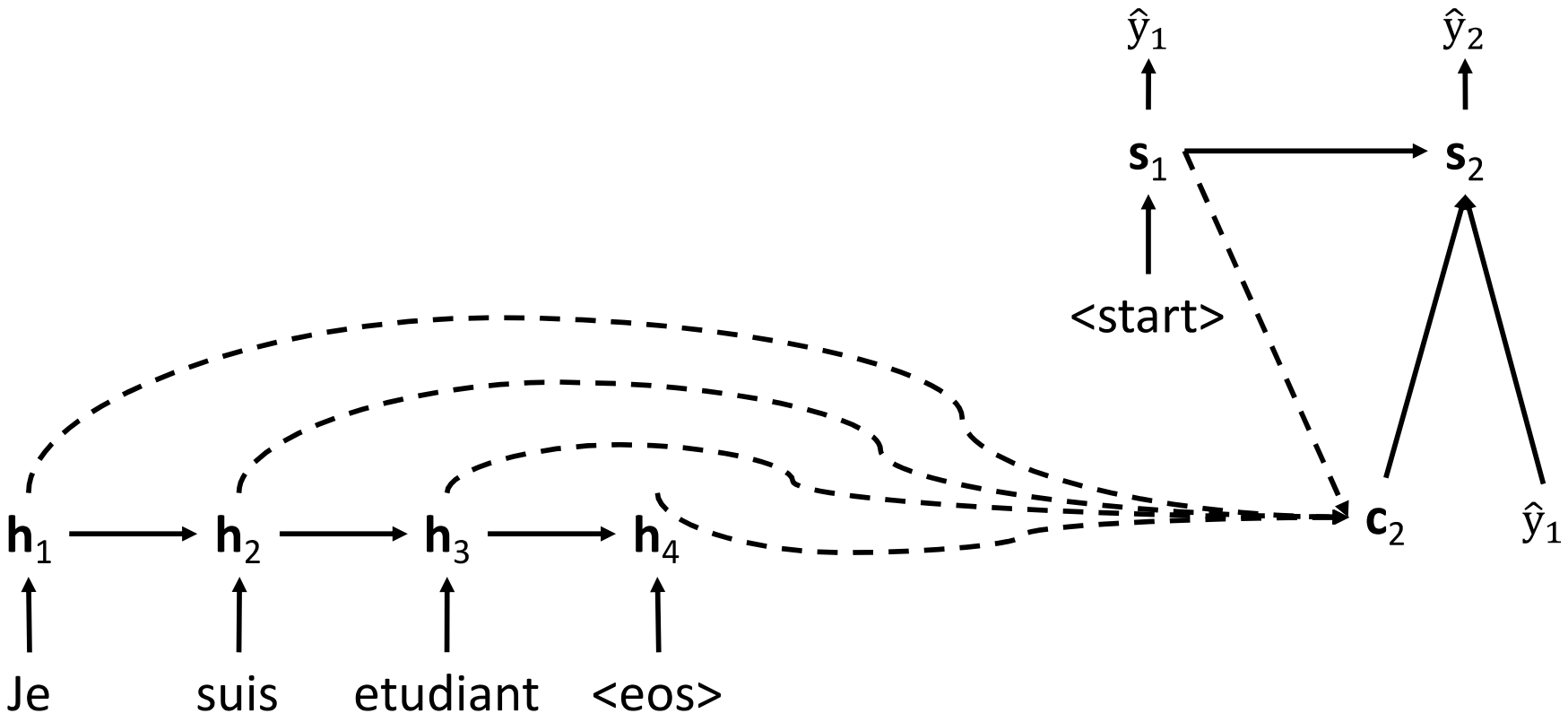
Show, Attend and Tell

- Example: “A bird flying over a body of water .”
  - Top row is “soft” attention, bottom row is “hard” attention.
- Model is “attending” to relevant part of image when generating word



Encoder-Decoder Architecture

- Seq2seq with attention



Encoder-Decoder Architecture

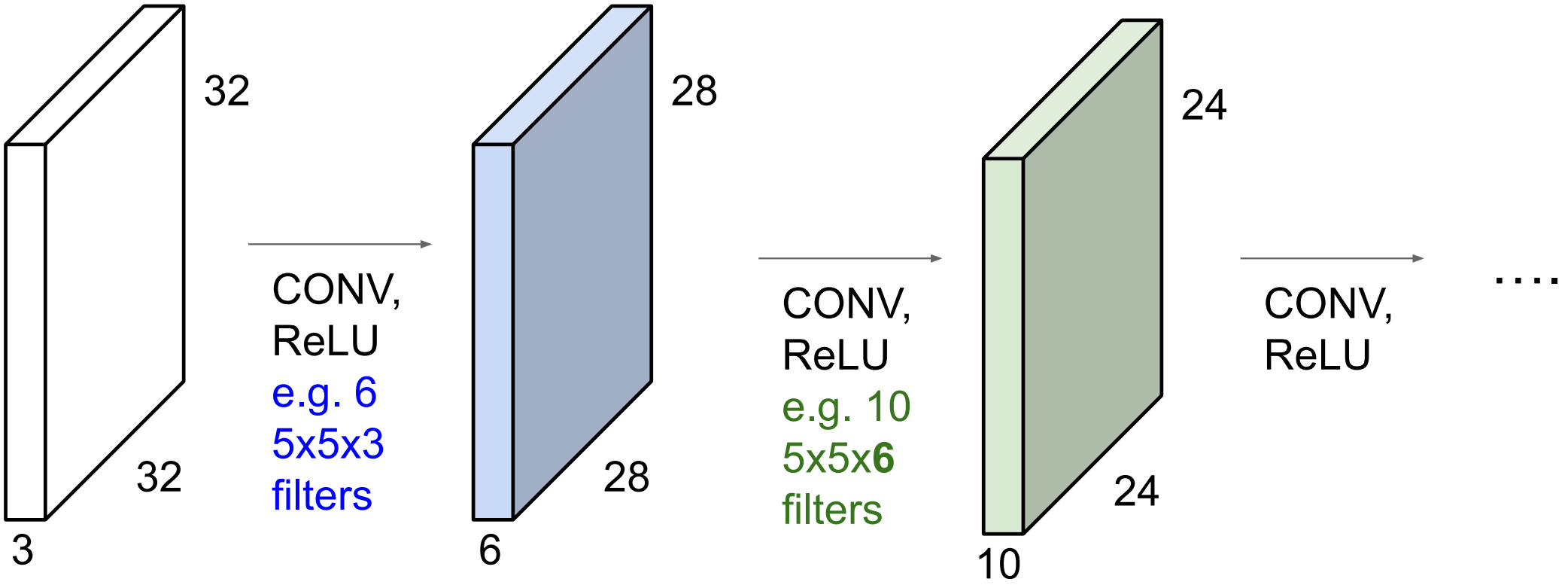
- **What we need:**
- Encoder to obtain image representation
- Decoder to generate caption
- Attention module to calculate attention weights

Encoder-Decoder Architecture

- **What we need:**
- Encoder to obtain image representation
  - Oxford VGGnet
- Decoder to generate caption
  - LSTM
- Attention module to calculate attention weights
  - MLP

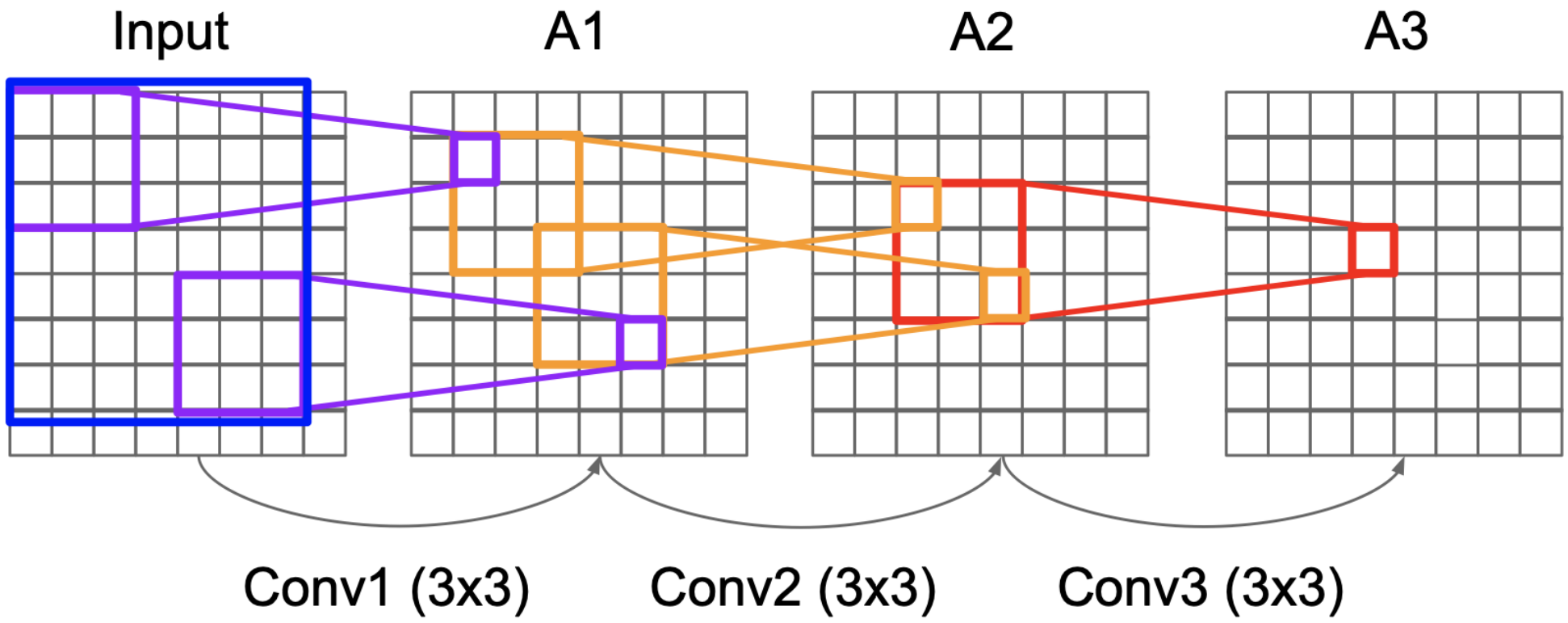
How to Attend to Part of Image

- Remember Convolution?



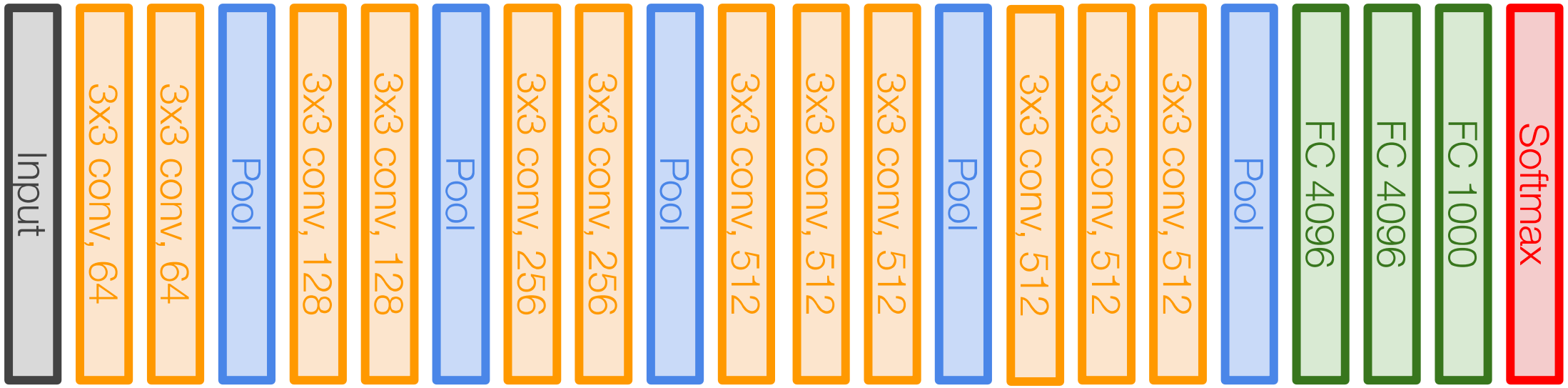
How to Attend to Part of Image

- Remember receptive field?



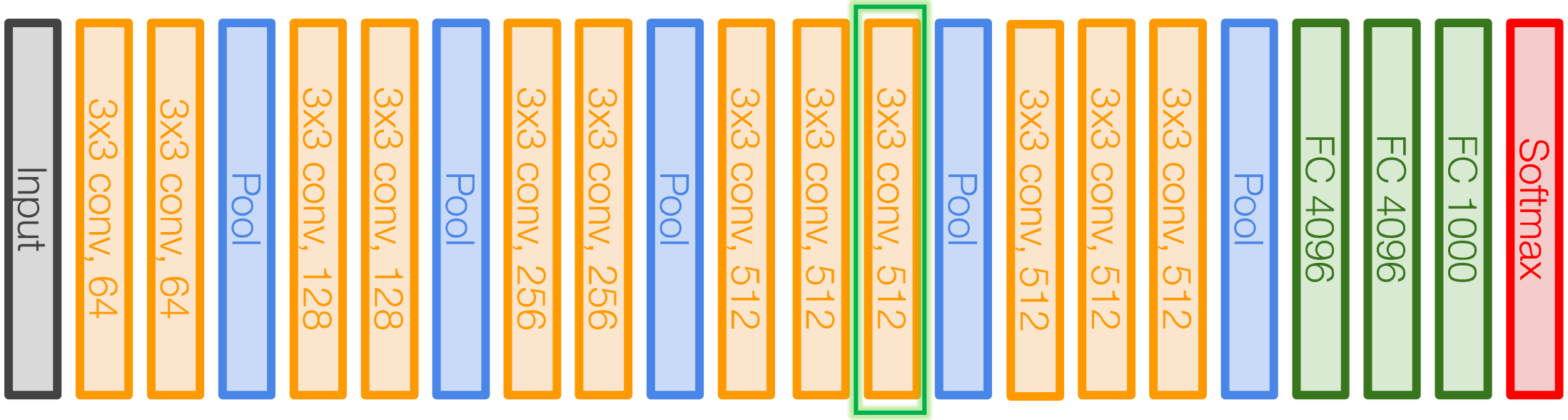
How to Attend to Part of Image

- Remember VGG 16?



How to Attend to Part of Image

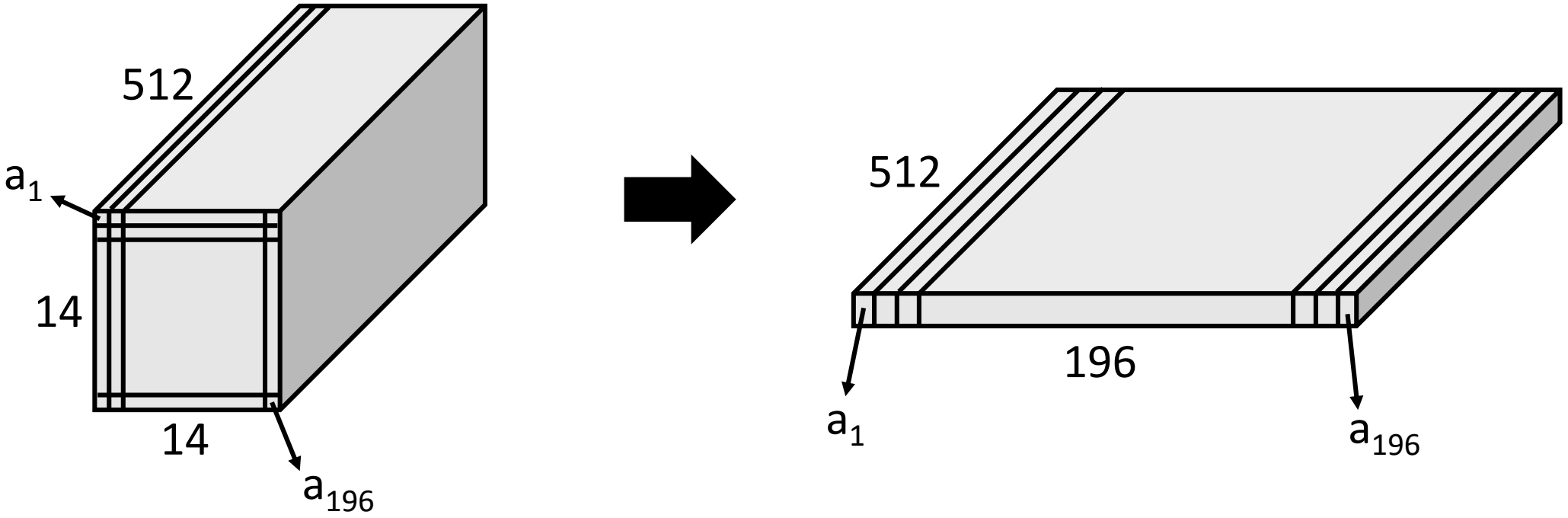
- Remember VGG 16?



Output of this convolution layer:
14 x 14 x 512 feature map
➔ 196 x 512 image representation vector

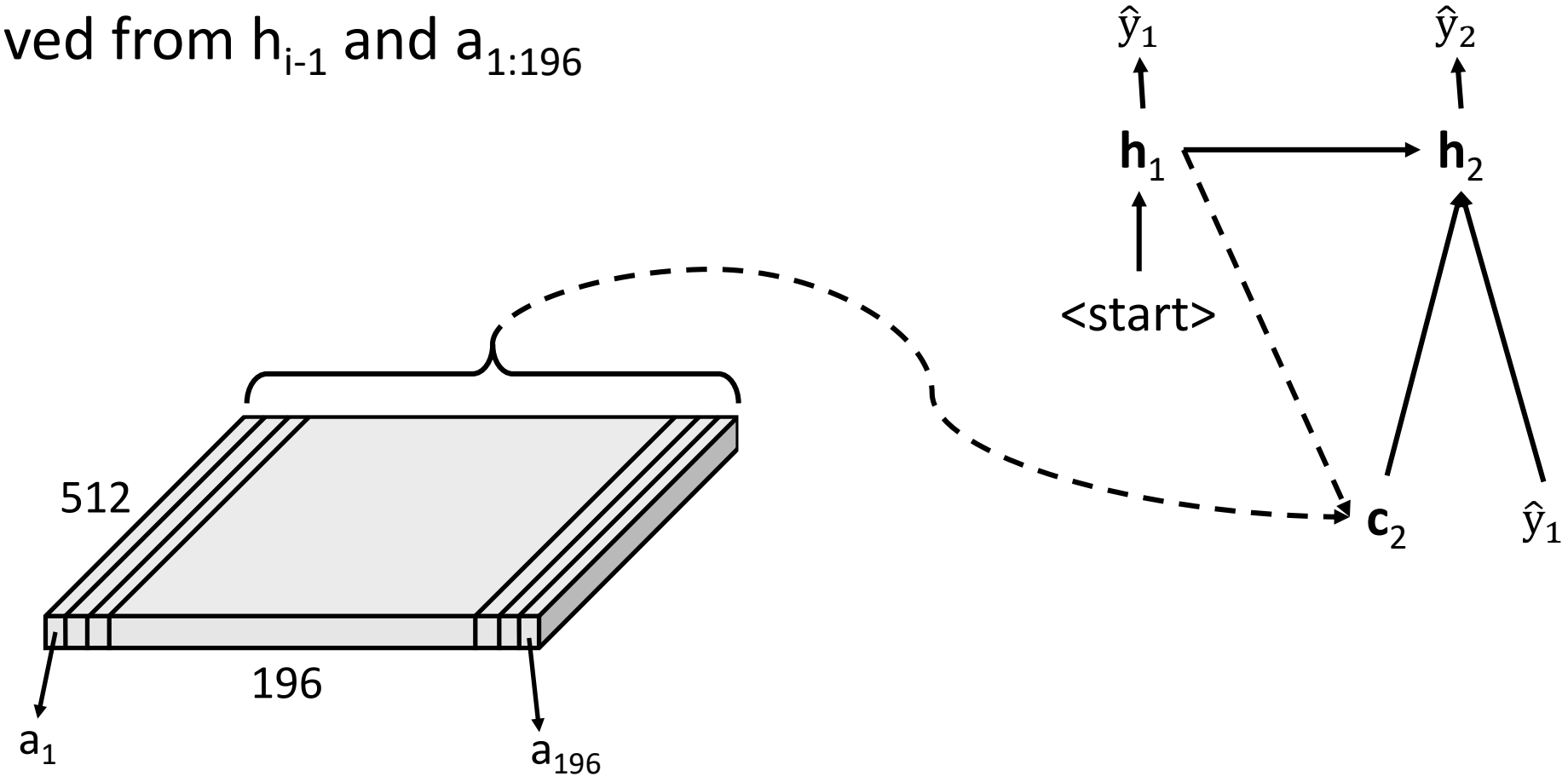
Model Architecture

- Flattening the image feature maps



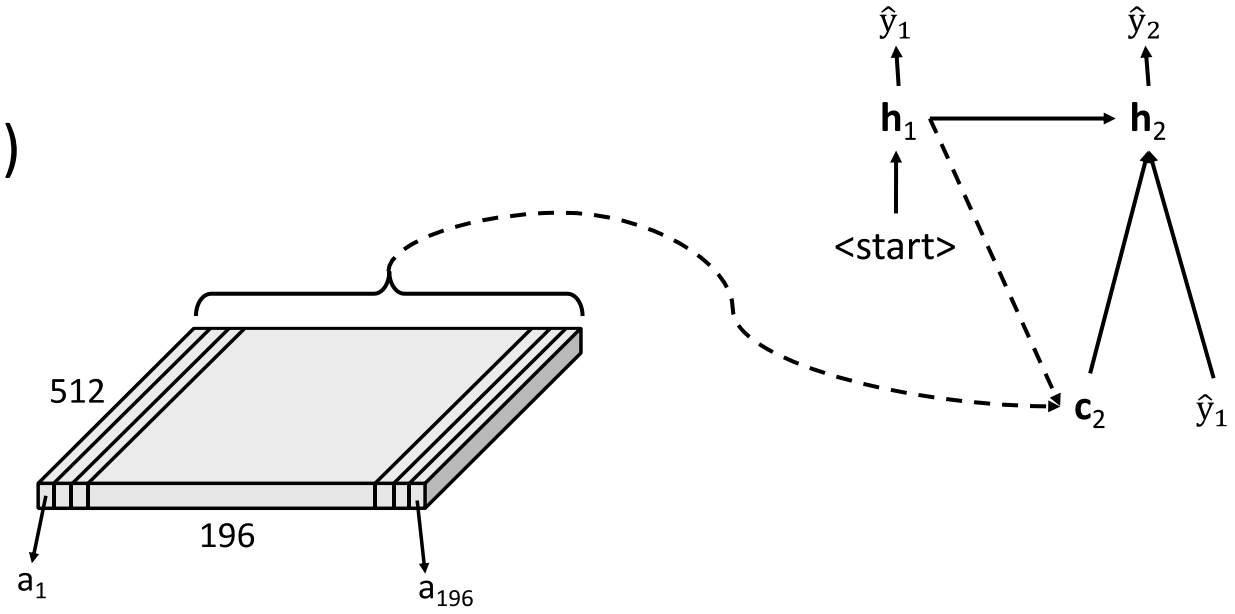
Show, Attend and Tell

- Each y_i is predicted based on h_i
- Each h_i is derived based on h_{i-1}, y_{i-1}, c_i
- c_i is derived from h_{i-1} and $a_{1:196}$



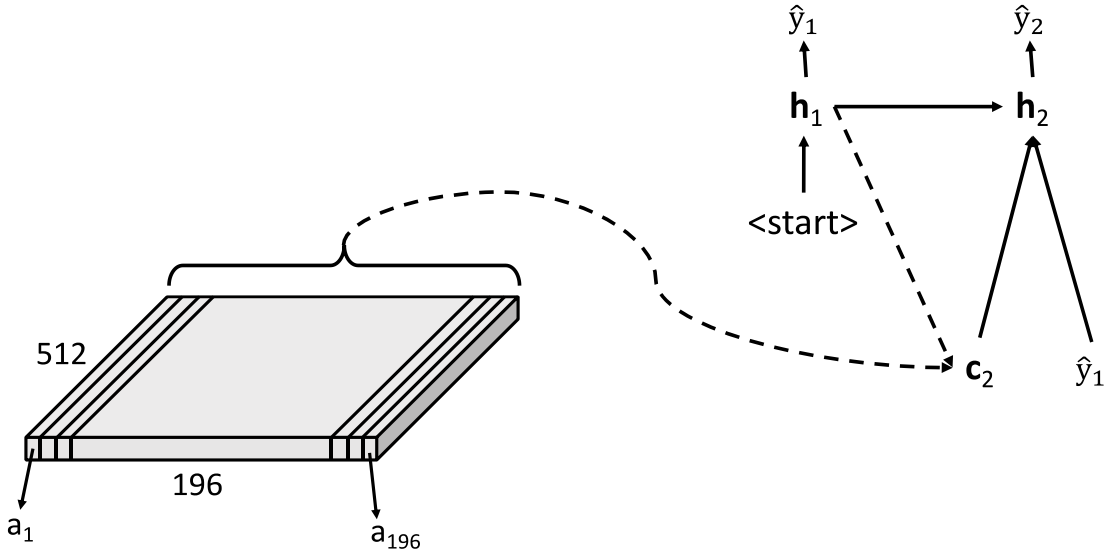
Show, Attend and Tell

- Each y_i is predicted based on h_i
  - $\hat{y}_1$ = Softmax($W_w h_i$ + b)
- Each h_i is derived based on h_{i-1}, y_{i-1}, c_i
  - h_i = RNN(h_{i-1}, $[y_{i-1};c_i]_{concat}$)
- c_i is derived from h_{i-1} and $a_{1:196}$
  - c_i = sum(α_i*a_i)
  - α_i = Softmax(f(h_{i-1}, a_1), … , f(h_{i-1}, a_{196}))
  - f(h_{i-1}, a_j) = $h_{i-1}^T W_f a_j$



Show, Attend and Tell

- **Some technical details**
- RNN's initial hidden state is learned
  - $h_0 = \text{MLP}\left(\frac{1}{L}\sum_{i=1}^{L} a_{1:L}\right)$
- Authors also tried "hard" attention.
  - Stochastically select only one a_i at each step.
  - Use reinforcement learning to train.
- Encourage $\sum_t \alpha_{ti} \approx 1$
  - Make the model pay equal attention to every part of image during text generation.



Model Performance

		BLEU				
Dataset	Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR
Flickr8k	Google NIC(Vinyals et al., 2014)$^{\dagger\Sigma}$	63	41	27	—	—
	Log Bilinear (Kiros et al., 2014a)$^{\circ}$	65.6	42.4	27.7	17.7	17.31
	Soft-Attention	**67**	44.8	29.9	19.5	18.93
	Hard-Attention	**67**	**45.7**	**31.4**	**21.3**	**20.30**
Flickr30k	Google NIC$^{\dagger\circ\Sigma}$	66.3	42.3	27.7	18.3	—
	Log Bilinear	60.0	38	25.4	17.1	16.88
	Soft-Attention	66.7	43.4	28.8	19.1	**18.49**
	Hard-Attention	**66.9**	**43.9**	**29.6**	**19.9**	18.46
COCO	CMU/MS Research (Chen & Zitnick, 2014)^{a}	—	—	—	—	20.41
	MS Research (Fang et al., 2014)$^{\dagger a}$	—	—	—	—	20.71
	BRNN (Karpathy & Li, 2014)$^{\circ}$	64.2	45.1	30.4	20.3	—
	Google NIC$^{\dagger\circ\Sigma}$	66.6	46.1	32.9	24.6	—
	Log Bilinear$^{\circ}$	70.8	48.9	34.4	24.3	20.03
	Soft-Attention	70.7	49.2	34.4	24.3	**23.90**
	Hard-Attention	**71.8**	**50.4**	**35.7**	**25.0**	23.04

Correction Attention Examples

Figure 3. Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)

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.

Incorrect Attention Examples

Figure 5. Examples of mistakes where we can use attention to gain intuition into what the model saw.

A large white bird standing in a forest.

A woman holding a clock in her hand.

A man wearing a hat and
a hat on a skateboard.

A person is standing on a beach
with a surfboard.

A woman is sitting at a table
with a large pizza.

A man is talking on his cell phone
while another man watches.

A woman is throwing a frisbee in a park.

Text-to-Image

Text-to-Image

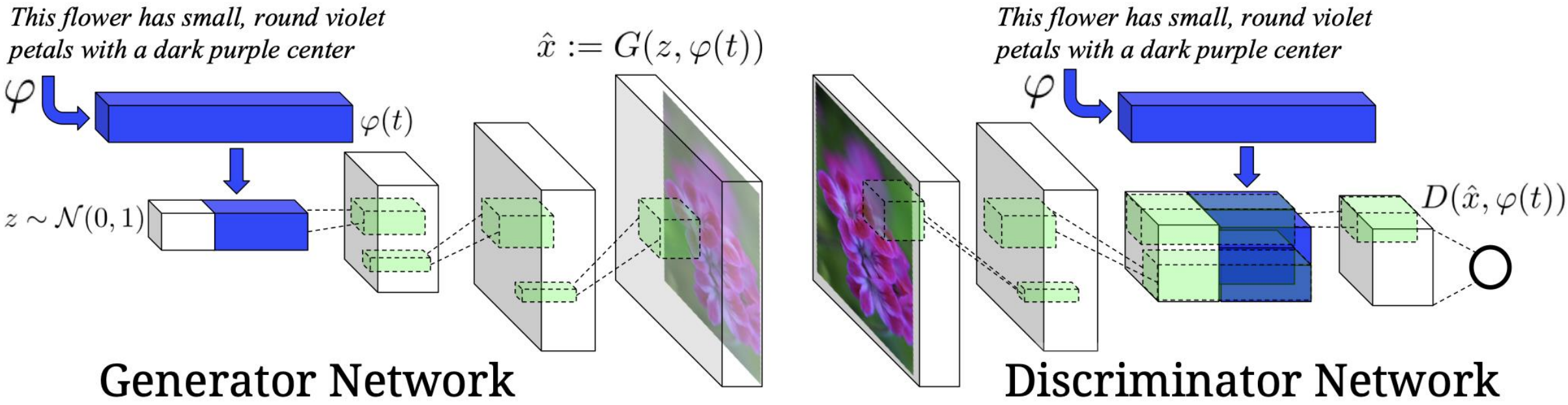
- Generative Adversarial Text to Image Synthesis
  - Reed et al. ICML 2016
- Text-conditioned image generation with GAN

this small bird has a pink breast and crown, and black primaries and secondaries.

the flower has petals that are bright pinkish purple with white stigma

Model Architecture

- Encode text with RNN
- Decode (i.e. generate) image with GAN
  - Use deconvolution (like DC-GAN) to upsample.



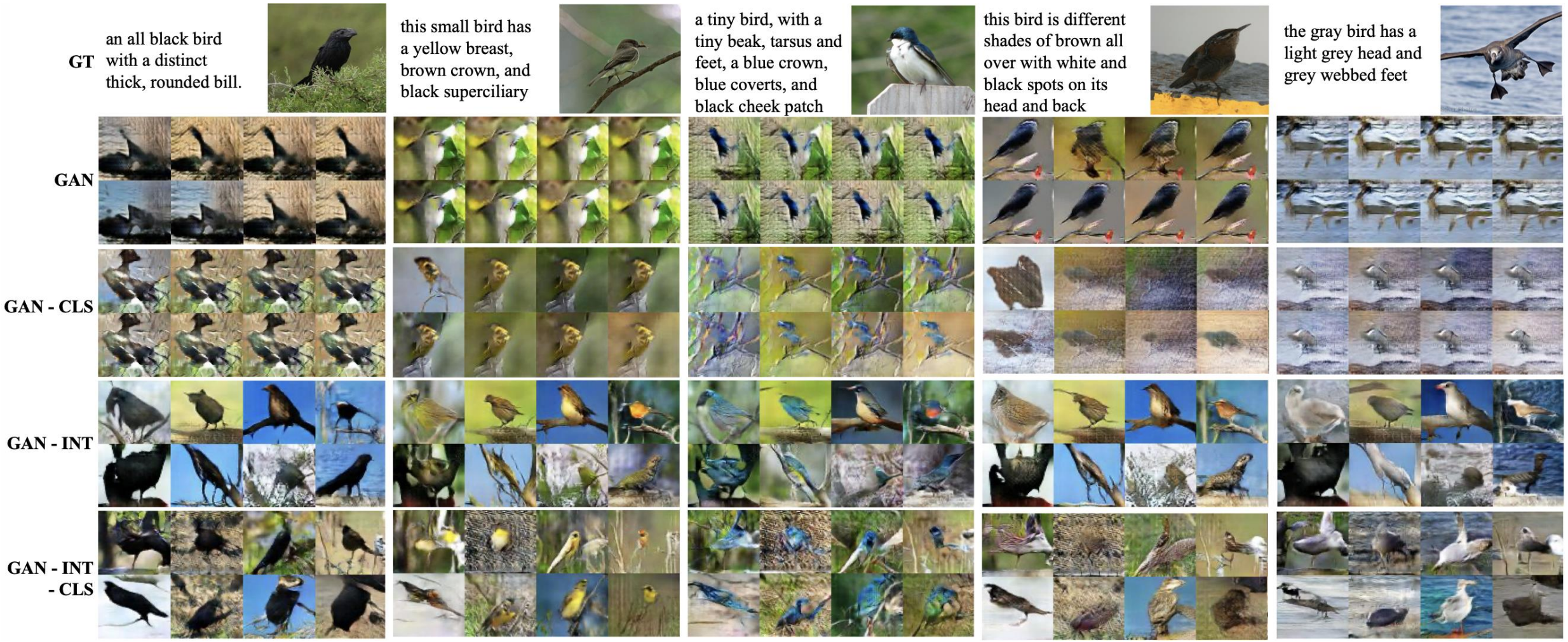
Training Strategy

- Discriminator's job is complicated
  - Real image with right text? ➔ Real!
  - Fake image with right text? ➔ Fake!
  - Real image with wrong text? ➔ Fake!
  - Fake image with wrong text? ➔ Fake!
- Discriminator is fed three cases
  - Real image, right text
  - Real image, wrong text
  - Fake image, right text

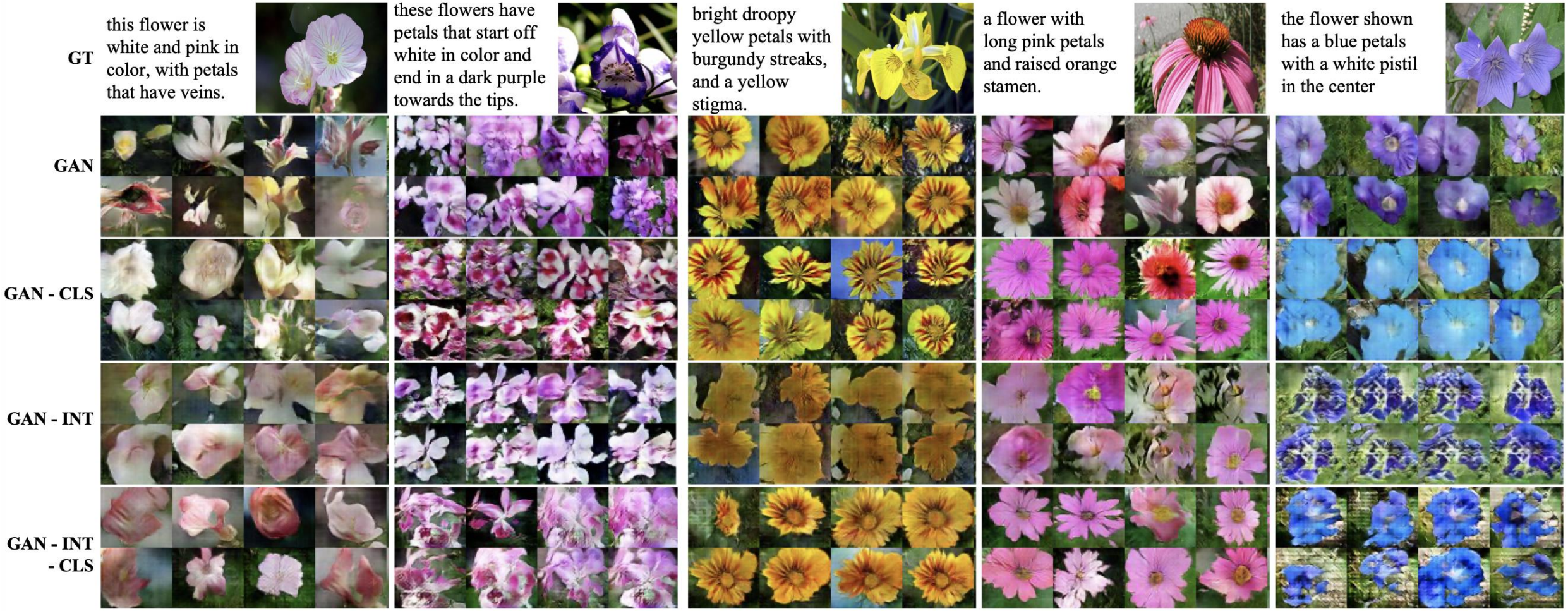
Algorithm 1 GAN-CLS training algorithm with step size α, using minibatch SGD for simplicity.

Input: minibatch images x, matching text t, mismatching $\hat{t}$, number of training batch steps S
for $n = 1$ **to** S **do**
  $h \leftarrow \varphi(t)$ {Encode matching text description}
  $\hat{h} \leftarrow \varphi(\hat{t})$ {Encode mis-matching text description}
  $z \sim \mathcal{N}(0, 1)^Z$ {Draw sample of random noise}
  $\hat{x} \leftarrow G(z, h)$ {Forward through generator}
  $s_r \leftarrow D(x, h)$ {real image, right text}
  $s_w \leftarrow D(x, \hat{h})$ {real image, wrong text}
  $s_f \leftarrow D(\hat{x}, h)$ {fake image, right text}
  $\mathcal{L}_D \leftarrow \log(s_r) + (\log(1 - s_w) + \log(1 - s_f))/2$
  $D \leftarrow D - \alpha \partial \mathcal{L}_D / \partial D$ {Update discriminator}
  $\mathcal{L}_G \leftarrow \log(s_f)$
  $G \leftarrow G - \alpha \partial \mathcal{L}_G / \partial G$ {Update generator}
end for

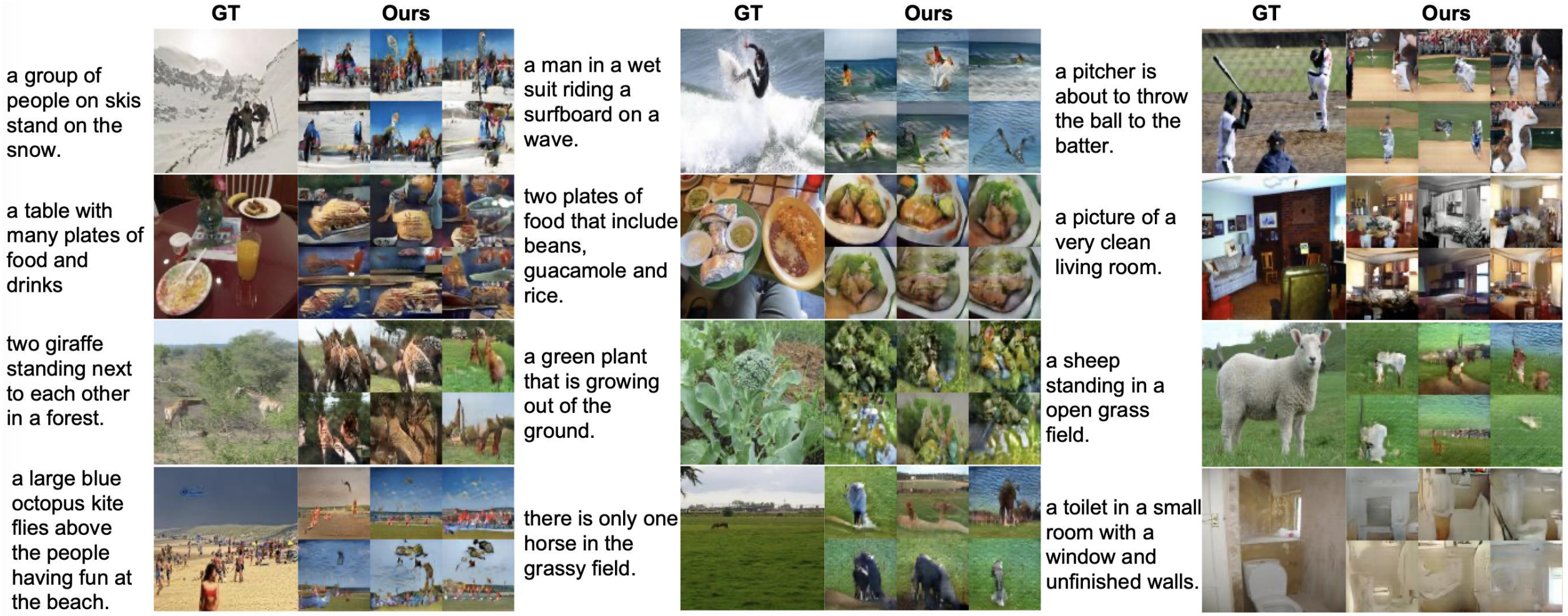
Examples


GT
an all black bird with a distinct thick, rounded bill.
this small bird has a yellow breast, brown crown, and black superciliary
a tiny bird, with a tiny beak, tarsus and feet, a blue crown, blue coverts, and black cheek patch
this bird is different shades of brown all over with white and black spots on its head and back
the gray bird has a light grey head and grey webbed feet
GAN
GAN - CLS
GAN - INT
GAN - INT - CLS

Examples


GT
this flower is white and pink in color, with petals that have veins.
these flowers have petals that start off white in color and end in a dark purple towards the tips.
bright droopy yellow petals with burgundy streaks, and a yellow stigma.
a flower with long pink petals and raised orange stamen.
the flower shown has a blue petals with a white pistil in the center
GAN
GAN - CLS
GAN - INT
GAN - INT - CLS

Examples


GT
Ours
a group of people on skis stand on the snow.
a table with many plates of food and drinks
two giraffe standing next to each other in a forest.
a large blue octopus kite flies above the people having fun at the beach.
a man in a wet suit riding a surfboard on a wave.
two plates of food that include beans, guacamole and rice.
a green plant that is growing out of the ground.
there is only one horse in the grassy field.
a pitcher is about to throw the ball to the batter.
a picture of a very clean living room.
a sheep standing in a open grass field.
a toilet in a small room with a window and unfinished walls.

DALL-E

- Zero-Shot Text-to-Image Generation
  - Ramesh et al. (OpenAI), 2021
- Purely based on Transformers + Vector Quantization
  - No GAN, no VAE
  - 64 layers, 62 attention heads, 12 billion params
  - 250 million text-image pairs collected from the Internet

TEXT PROMPT an illustration of a baby daikon radish in a tutu walking a dog

AI-GENERATED IMAGES

TEXT & IMAGE PROMPT the exact same cat on the top as a sketch on the bottom

AI-GENERATED IMAGES

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

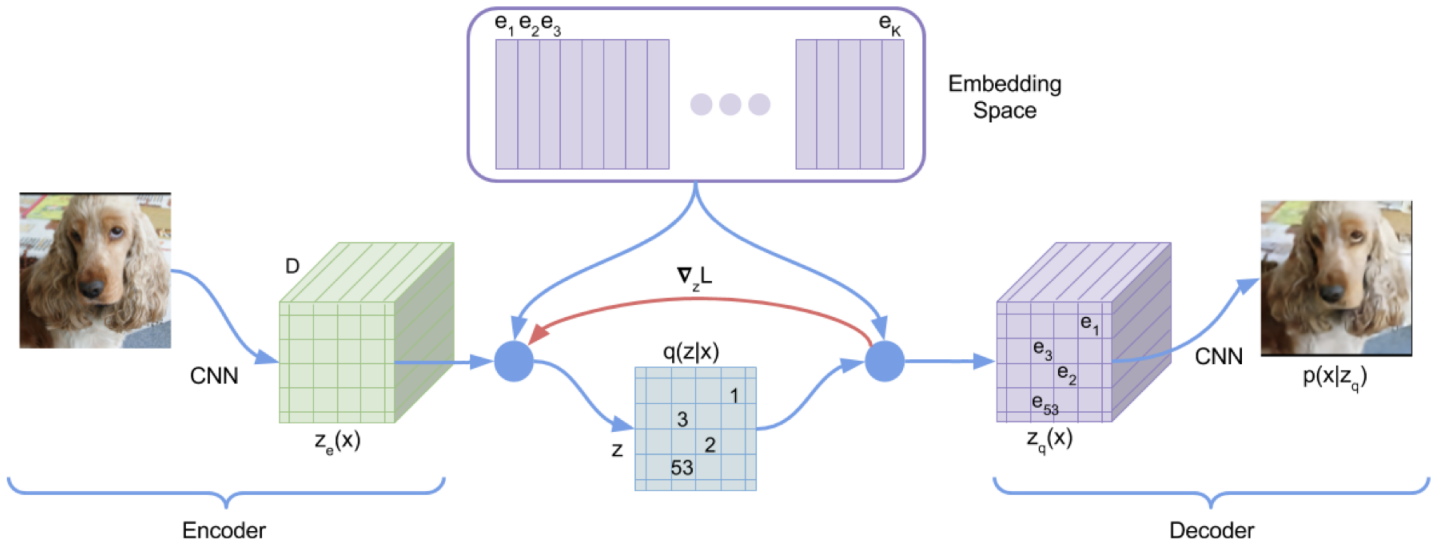
AI-GENERATED IMAGES

TEXT PROMPT a store front that has the word ‘openai’ written on it. . . .

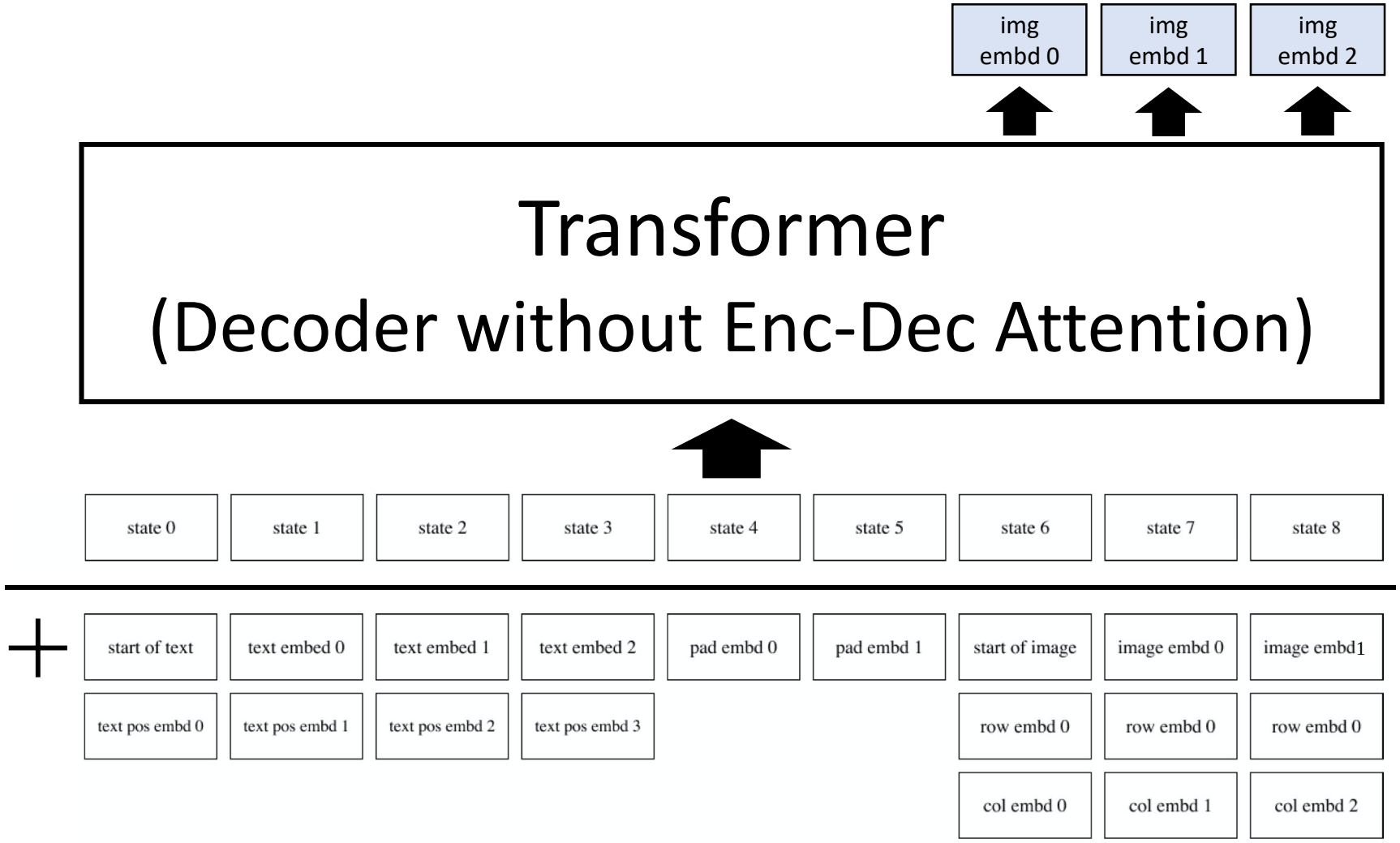
AI-GENERATED IMAGES

Image Tokens

- Use “vector quantization”
  - “Neural Discrete Representation Learning”, van den Oord et al. (DeepMind), 2017
- Replace each image feature with a image token
  - There is a predefined dictionary of image tokens
  - Now an image can be represented as a sequence of tokens (like text!)

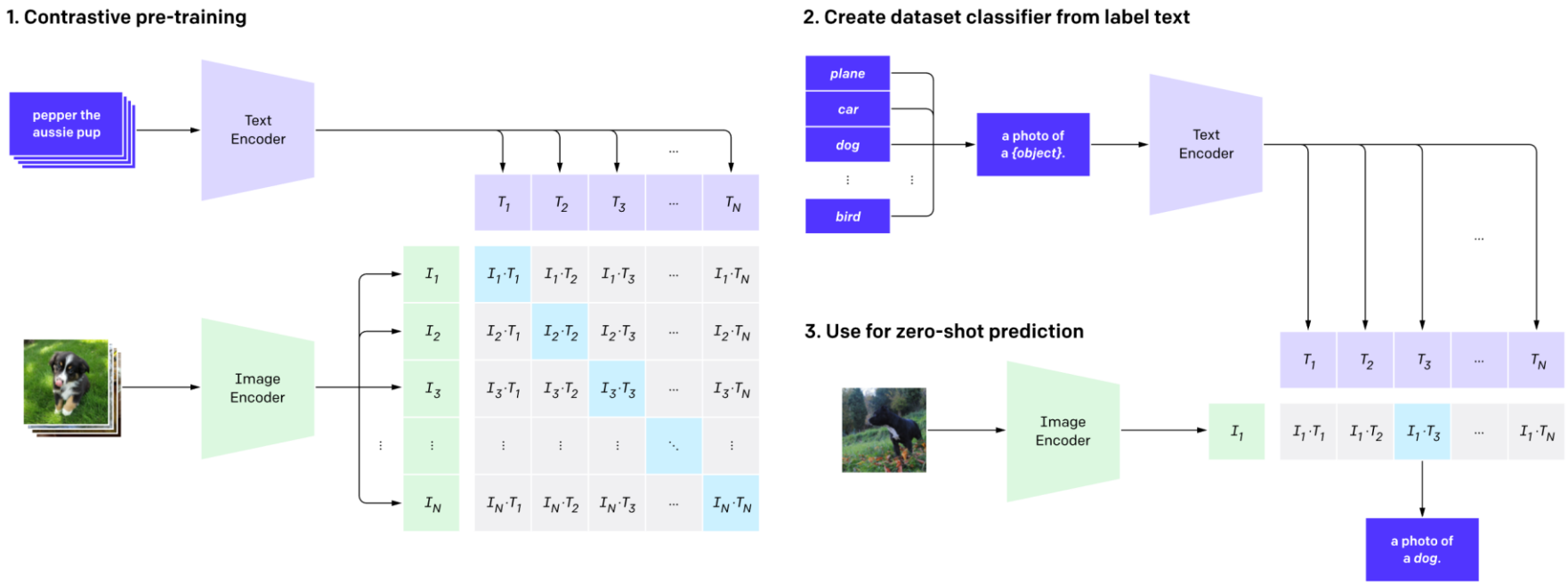


DALL-E Architecture



CLIP

- Learning Transferable Visual Models From Natural Language Supervision
  - Radford, Kim et al. 2021 (OpenAI)
  - Contrastive learning between text and image
  - Great zero-shot performance
  - Understands the relationship between text and image very well



DALL-E 2

- Hierarchical Text-Conditional Image Generation with CLIP Latents
  - Ramesh et al. 2022 (OpenAI)
  - Text-to-image generation using CLIP priors and classifier-free guided diffusion
  - Two-step upsampling (also diffusion)

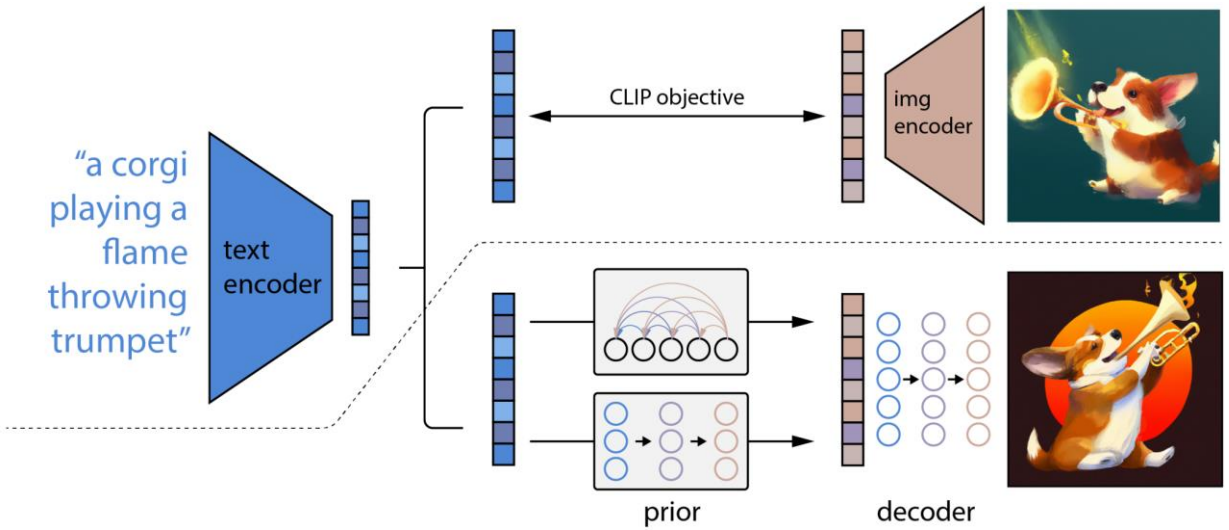


Image-Text Multi-modal Pre-training

Image-Text Multi-modal Pretraining

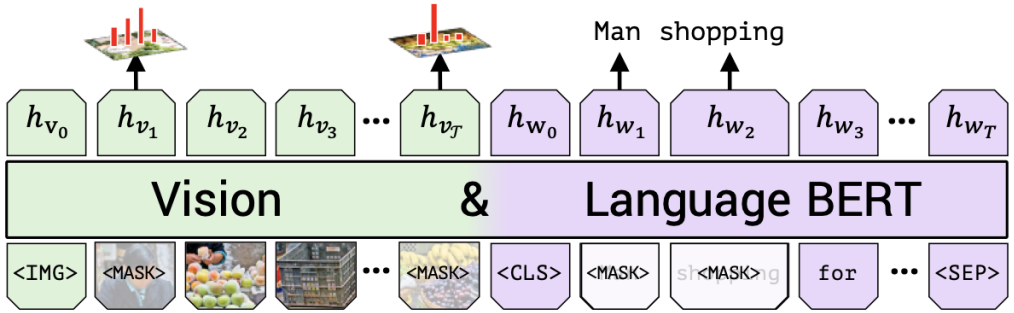
- Very active since 2019
  - VideoBERT, ViLBERT, InterBERT, LXMERT, UNITER, Unified VLP, PixelBERT, CoCa, Flamingo, BEiT v3
- Objective
  - Pre-train a model to “understand” the relationship between images and text
- Downstream tasks
  - Image retrieval
  - Visual question answering
  - Image captioning
  - Image generation
  - …

Common Strategy

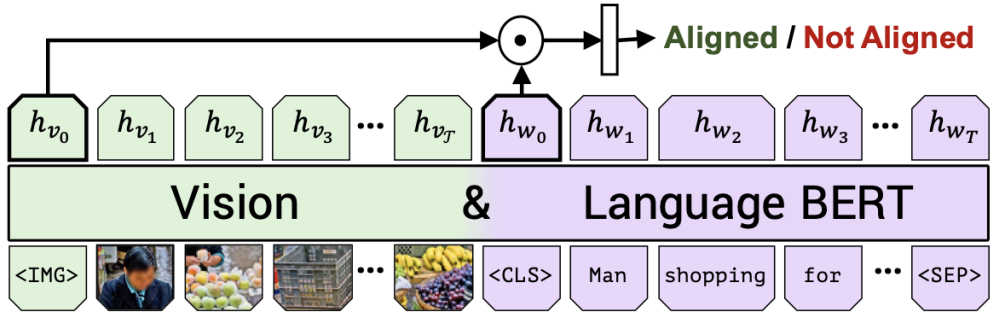
- Extract image features from the image
  - Pre-trained object detectors (e.g. Fast R-CNN, Mask R-CNN)
  - Directly feed pixel feature maps
  - Use VQVAE to quantize images into code
- Feed image features and text to BERT
- Optimize for some pre-training objective
  - Masked language modeling
  - Masked image predictiong
  - Image-text alignment
  - …

ViLBERT

- ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks
  - Lu et al, NeurIPS 2019
- Masked image modeling
  - Predict the class distribution from Mask R-CNN
- Masked language modeling
  - Same as BERT
- Image-Text alignment prediction
  - Predict whether the given pair is a matching pair



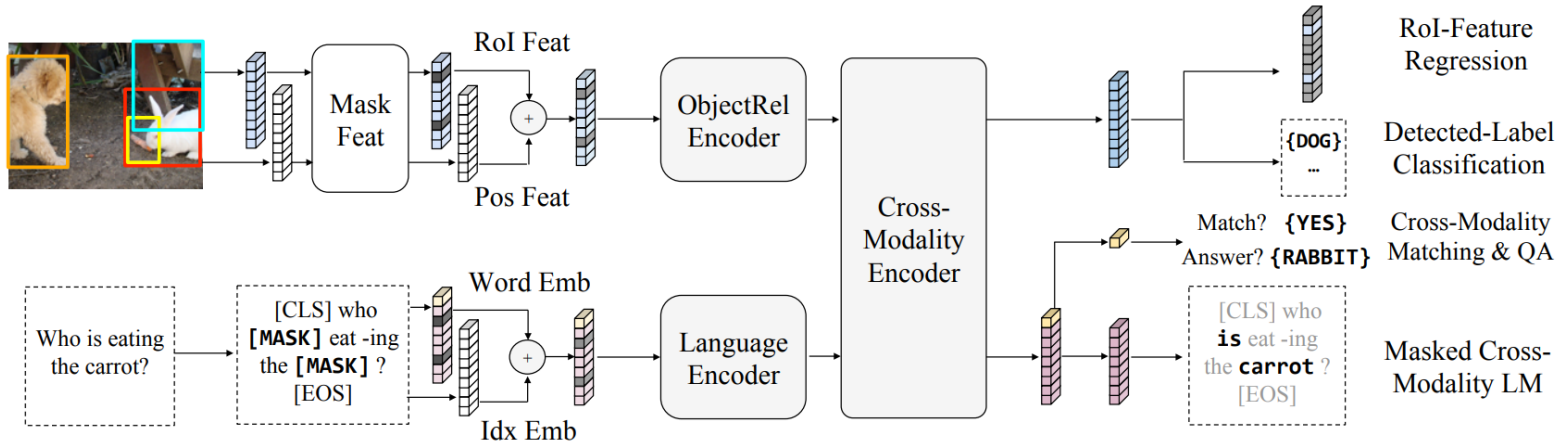
(a) Masked multi-modal learning



(b) Multi-modal alignment prediction

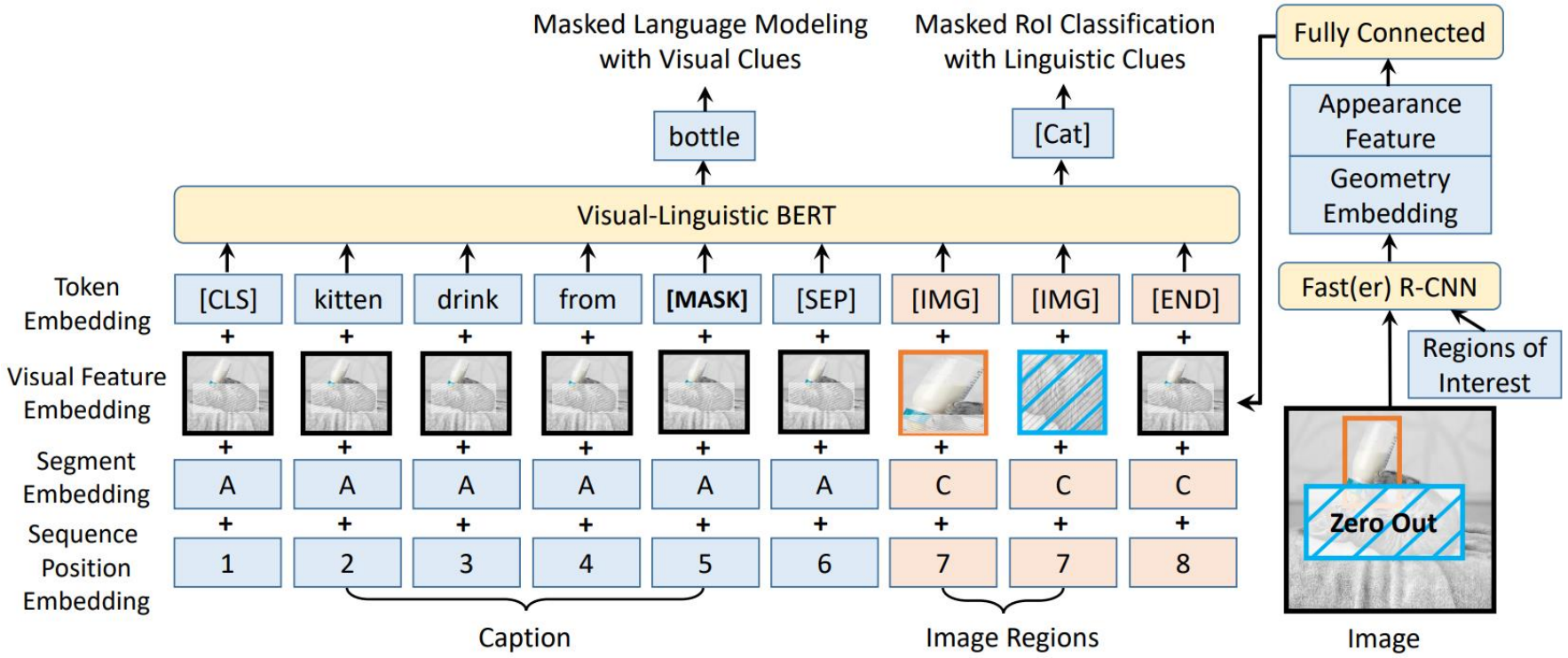
LXMERT

- LXMERT: Learning Cross-Modality Encoder Representations from Transformers
  - Tan and Bansal, EMNLP 2019
- Masked image modeling
  - Feature regression
  - Label classification
- Masked language modeling
- Image-Text alignment prediction



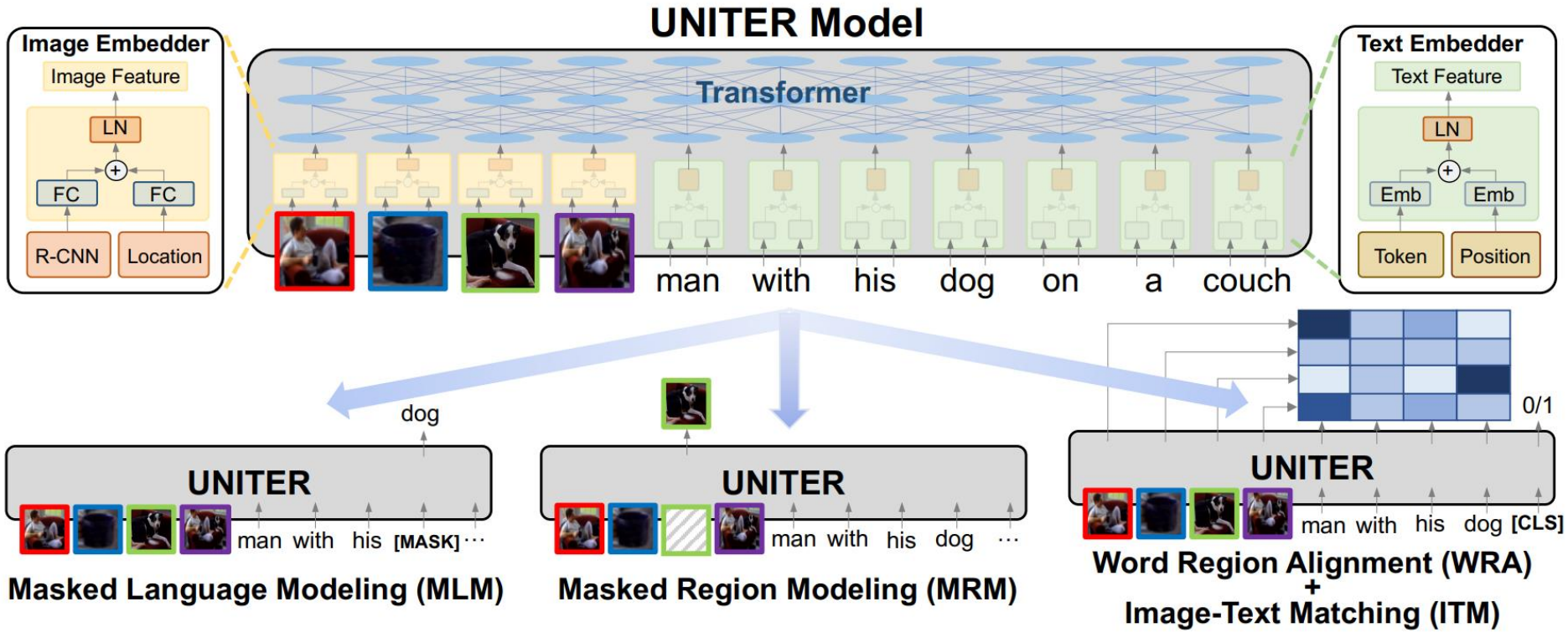
VL-BERT

- VL-BERT: Pre-training of Generic Visual-Linguistic Representations
  - Su et al., ICLR 2020



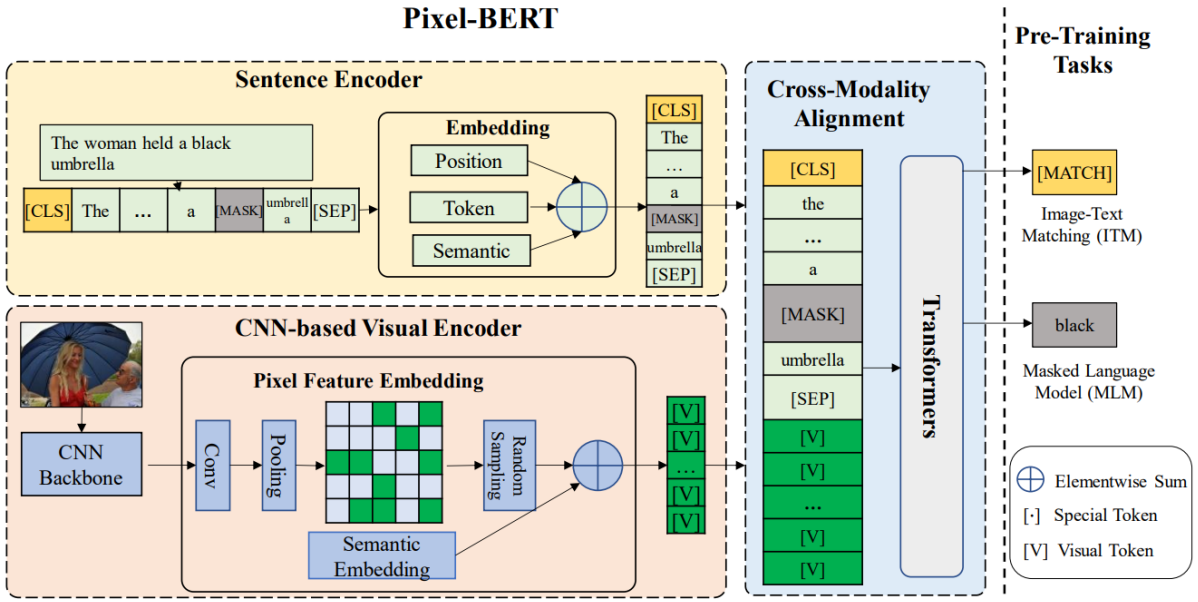
UNITER

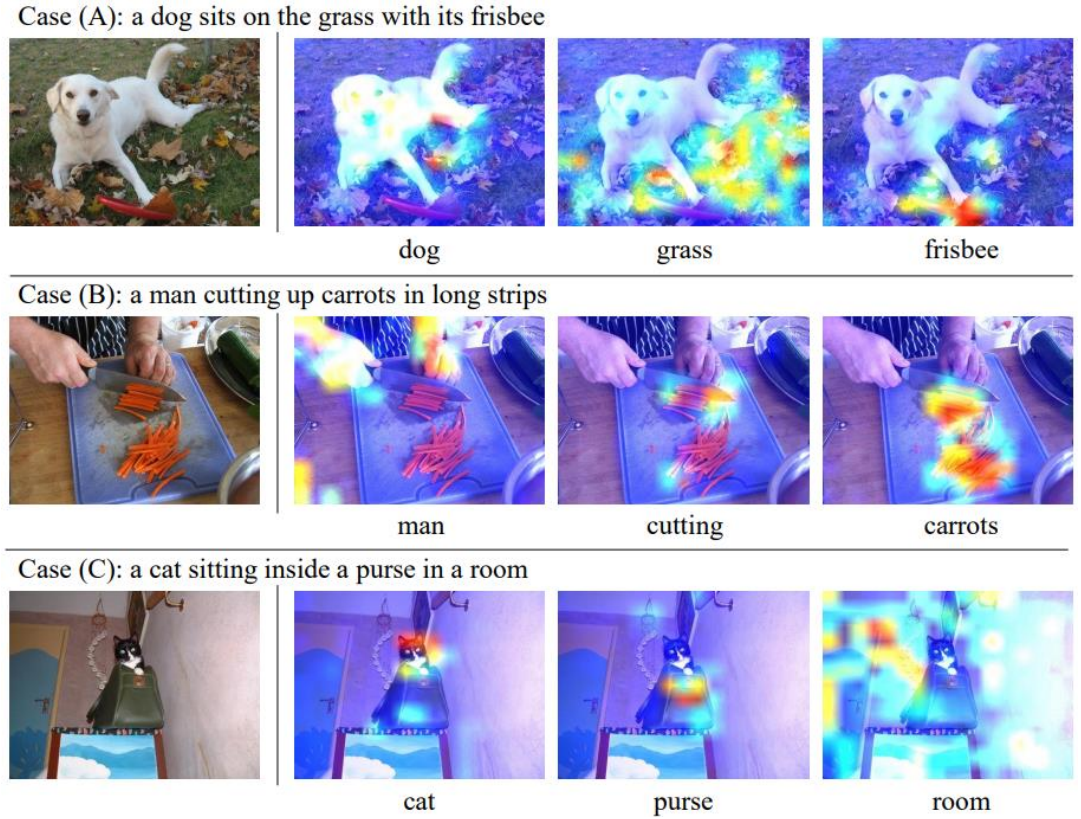
- UNITER: UNiversal Image-TExt Representation Learning
  - Chen et al., ECCV 2020



Pixel-BERT

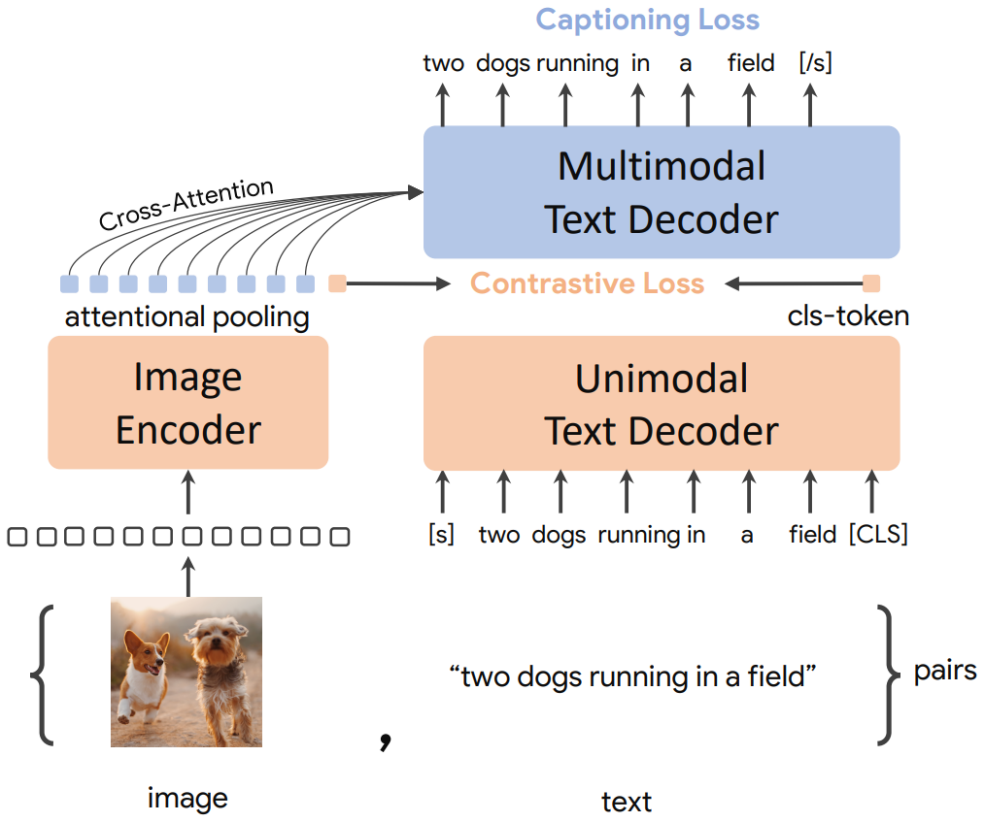
- Pixel-BERT: Aligning Image Pixels with Text by Deep Multi-Modal Transformers
  - Huang et al. 2020
  - Simple architecture (only CNN + Transformer, NO Object detector)





CoCa

- CoCa: Contrastive Captioners are Image-Text Foundation Models
  - Yu et al. 2022 (Google)
  - Contrastive loss + captioning loss



Flamingo

- Flamingo: a Visual Language Model for Few-Shot Learning
  - Alayrac et al. 2022 (DeepMind)
  - Inter-leaved text-image sequences for prompt-based few-shot tasks

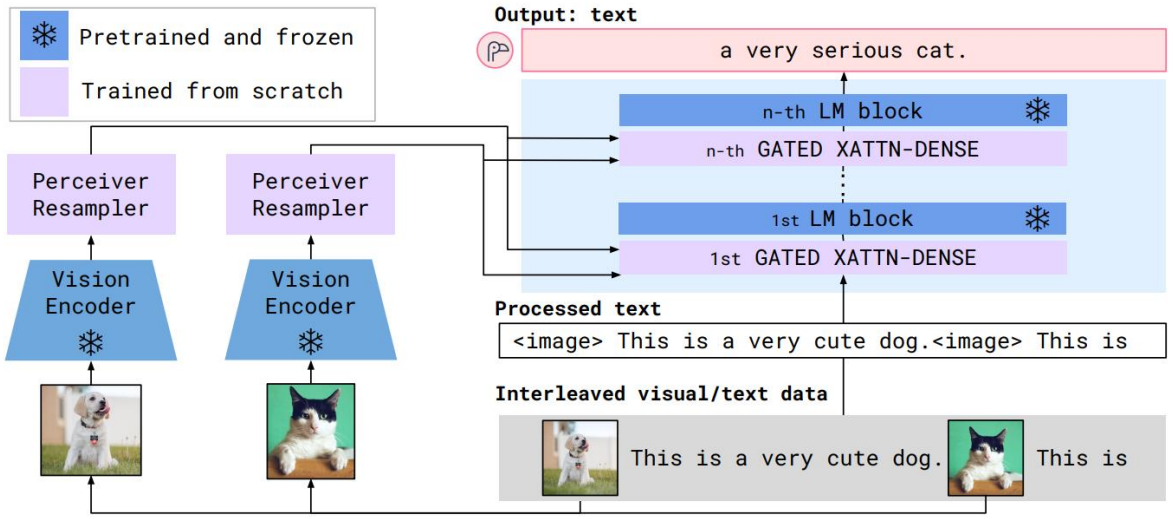
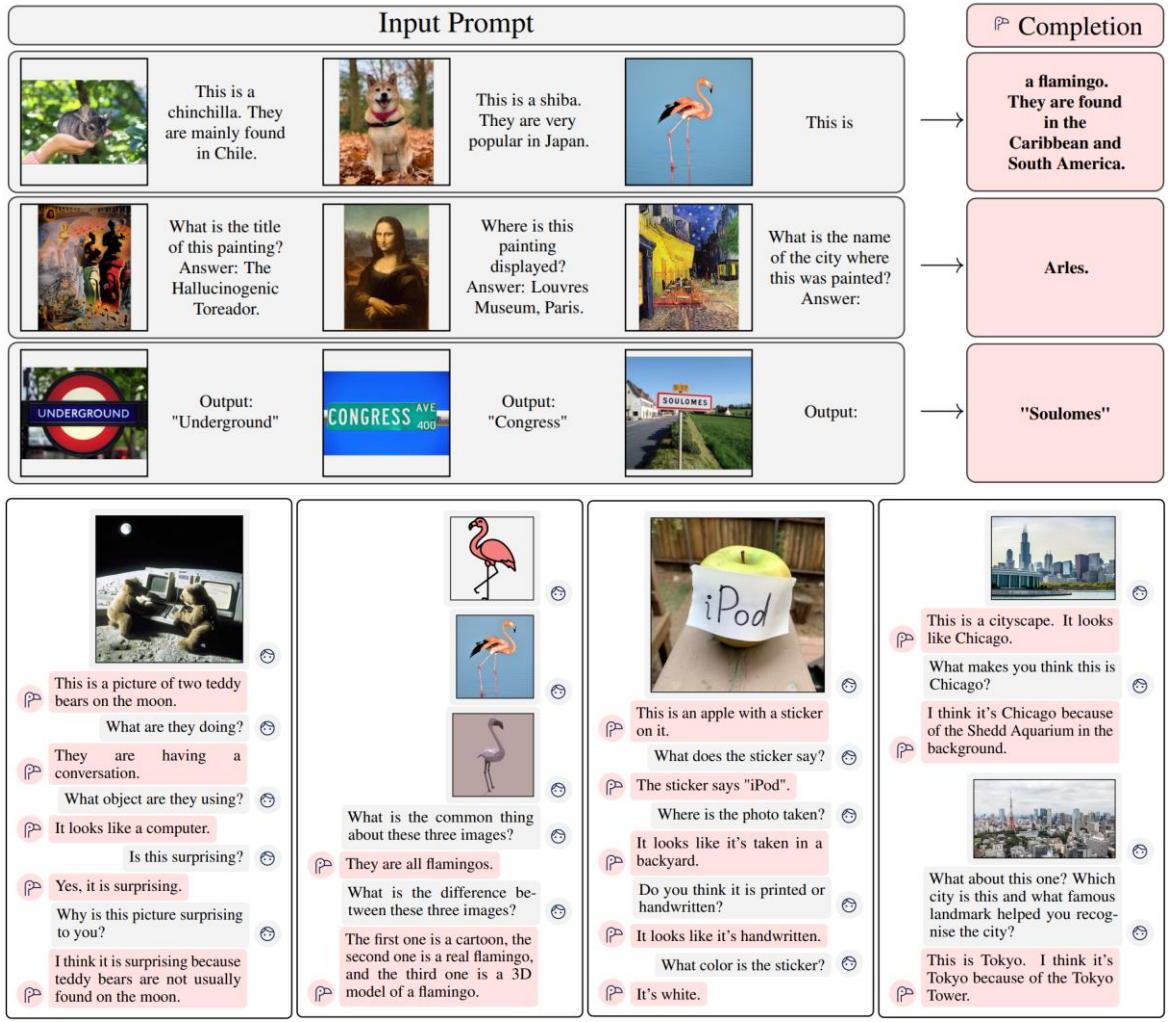


Figure 3: **Flamingo architecture overview.** Flamingo is a family of visual language models (VLMs) that take as input visual data interleaved with text and produce free-form text as output.



AI504: Programming for Artificial Intelligence

Week 15: Image-Text Multimodal Learning

Edward Choi

Grad School of AI

edwardchoi@kaist.ac.kr