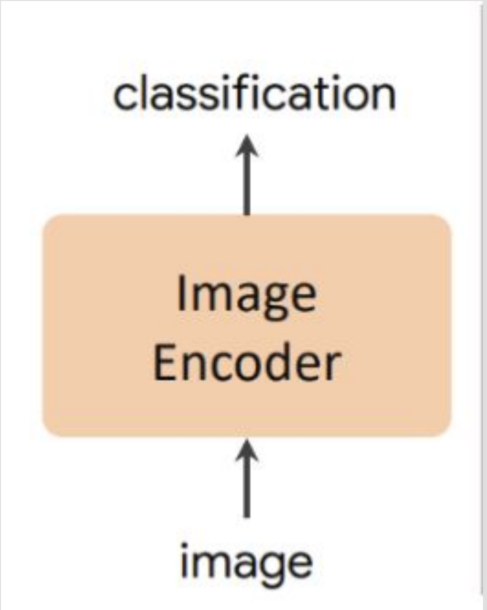
CoCa: Contrastive Captioners are Image-Text Foundation Models

Machine Learning Seminar, group 192
Ganzhara Vladimir, Akulov Dmitrii, Ismagilov Artem

Single Encoder Classification

- Image Encoder: Image -> features
  - Examples: ConvNet, ViT (Vision Transformers)
- Classifier: features -> class
  - Full layer.

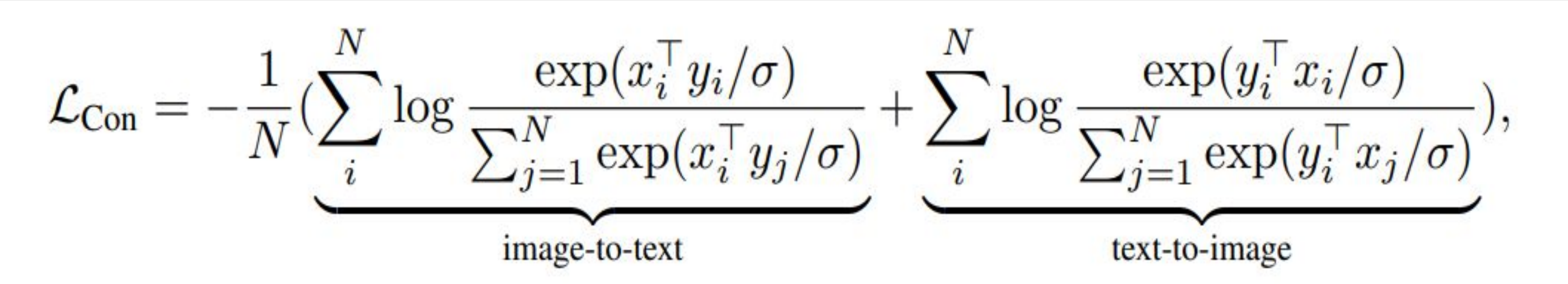


Simple, direct, profit.

Accuracy on ImageNet:
#3 (ensamble) Model Soup Vit-G/14 (2022) - 90.9%
#8 Vit-G/14 (2022) - 90.45%,
#19 NFNet-F4+ (2021) - 89.2%

Dual-Encoder Contrastive Learning

- Image Encoder: Image -> embedding
- Text Encoder: Text -> embedding



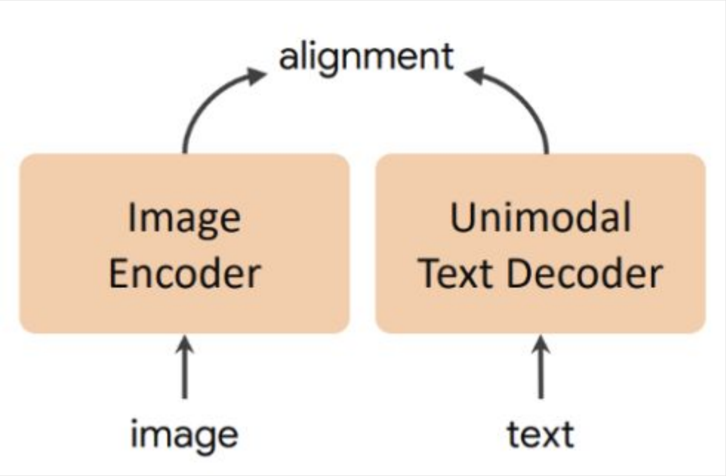
$$\mathcal{L}_{\text{Con}} = -\frac{1}{N}\Big(\underbrace{\sum_{i}^{N} \log \frac{\exp(x_i^\top y_i/\sigma)}{\sum_{j=1}^{N} \exp(x_i^\top y_j/\sigma)}}_{\text{image-to-text}} + \underbrace{\sum_{i}^{N} \log \frac{\exp(y_i^\top x_i/\sigma)}{\sum_{j=1}^{N} \exp(y_i^\top x_j/\sigma)}}_{\text{text-to-image}}\Big),$$

x_i - embedding of i-th image   y_i - embedding of i-th text.

Logic: Maximize distance between text and image embeddings from different indexes, while minimizing the same.

Dual-Encoder Contrastive Learning

Q: What are the advantages of dual encoding?

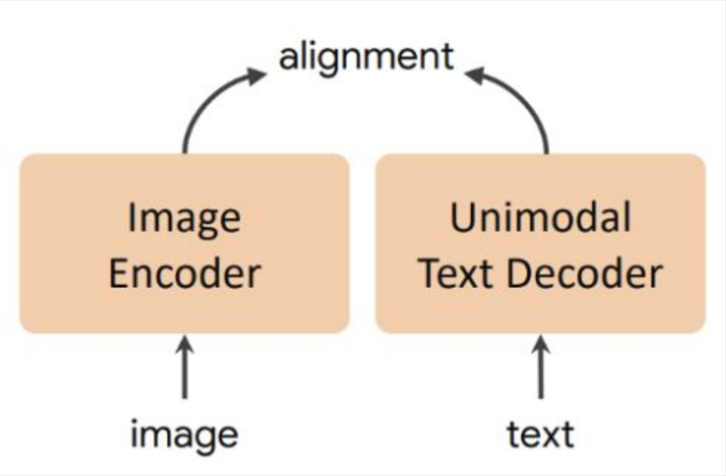


Dual-Encoder Contrastive Learning

Data: Image-Text.
Text is a description of an image. (For example: JFT)

Q: What are the advantages of dual encoding?

A: Instead of working with precise labels, we can use broad (noisy) texts instead.



Zero-Shot Image Classification

Zero-Shot Learning -
pre-trained model encounters data from unseen classes, but still has to predict the class.

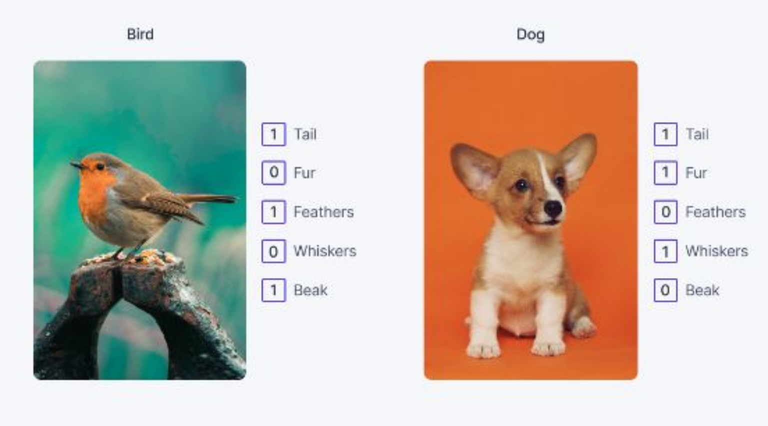
Q: How can we do this with dual encoding?

Zero-Shot Image Classification

Zero-Shot Learning -
pre-trained model encounters data from unseen classes, but still has to predict the class.

Q: How can we do this with dual encoding paradigm?

A: Our texts are descriptions, from which we can infer, what is important to classify this object.

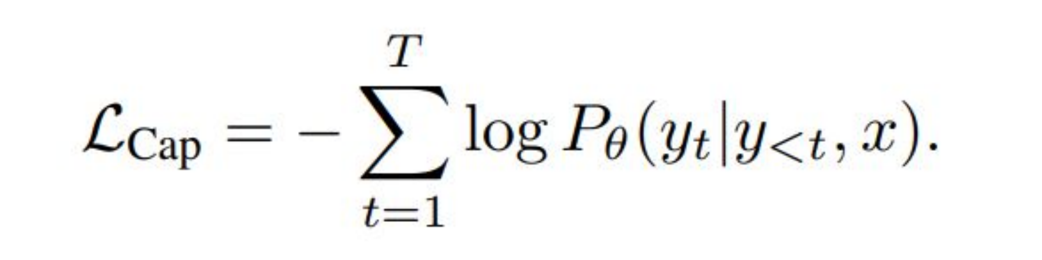


For more info on Zero Shot Image Classification check out CLIP or Learning Transferable Visual Models From Natural Language Supervision.

Encoder-Decoder Captioning

- Image Encoder: Image -> embedding
- Unimodal Text Decoder: Text -> embedding
- Multimodal Text Decoder: Image embedding + previous text tokens embedding -> next word

- Loss: maximizing probability of a word conditioned on image embedding and previous words.



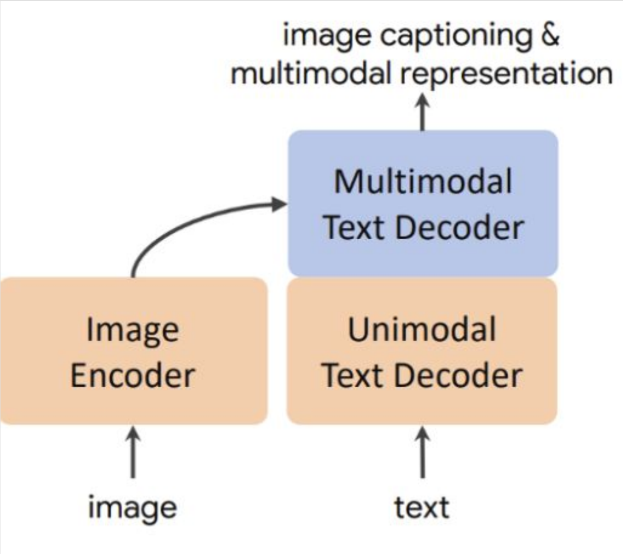
$$\mathcal{L}_{\text{Cap}} = -\sum_{t=1}^{T} \log P_{\theta}(y_t | y_{<t}, x).$$

Encoder-Decoder Captioning

Difference between Multimodal and Unimodal decoders: cross-attention layers.

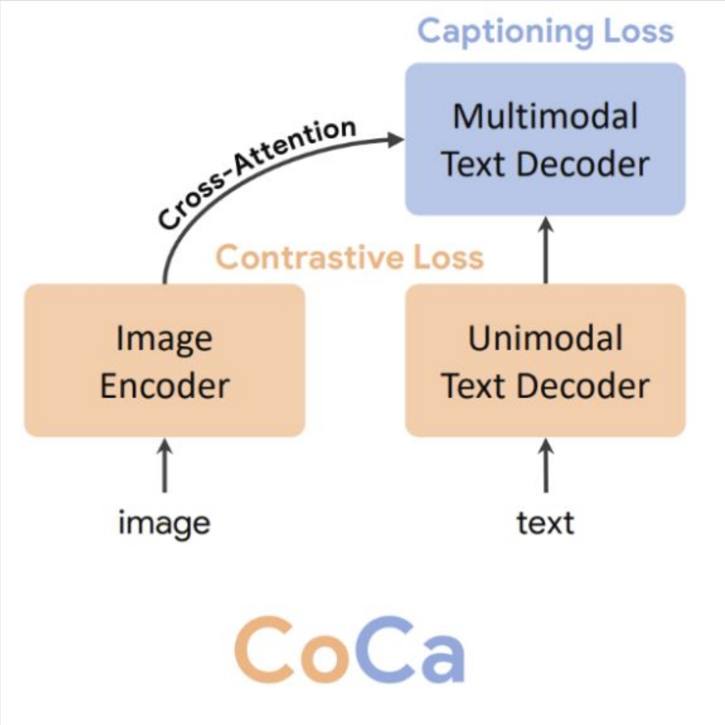
New skill: Image-based text generation and overall image-language understanding.

Task examples: answering questions to images or absolutely effortlessly using it for zero-shot learning btw.



Contrastive Captioners (CoCa)

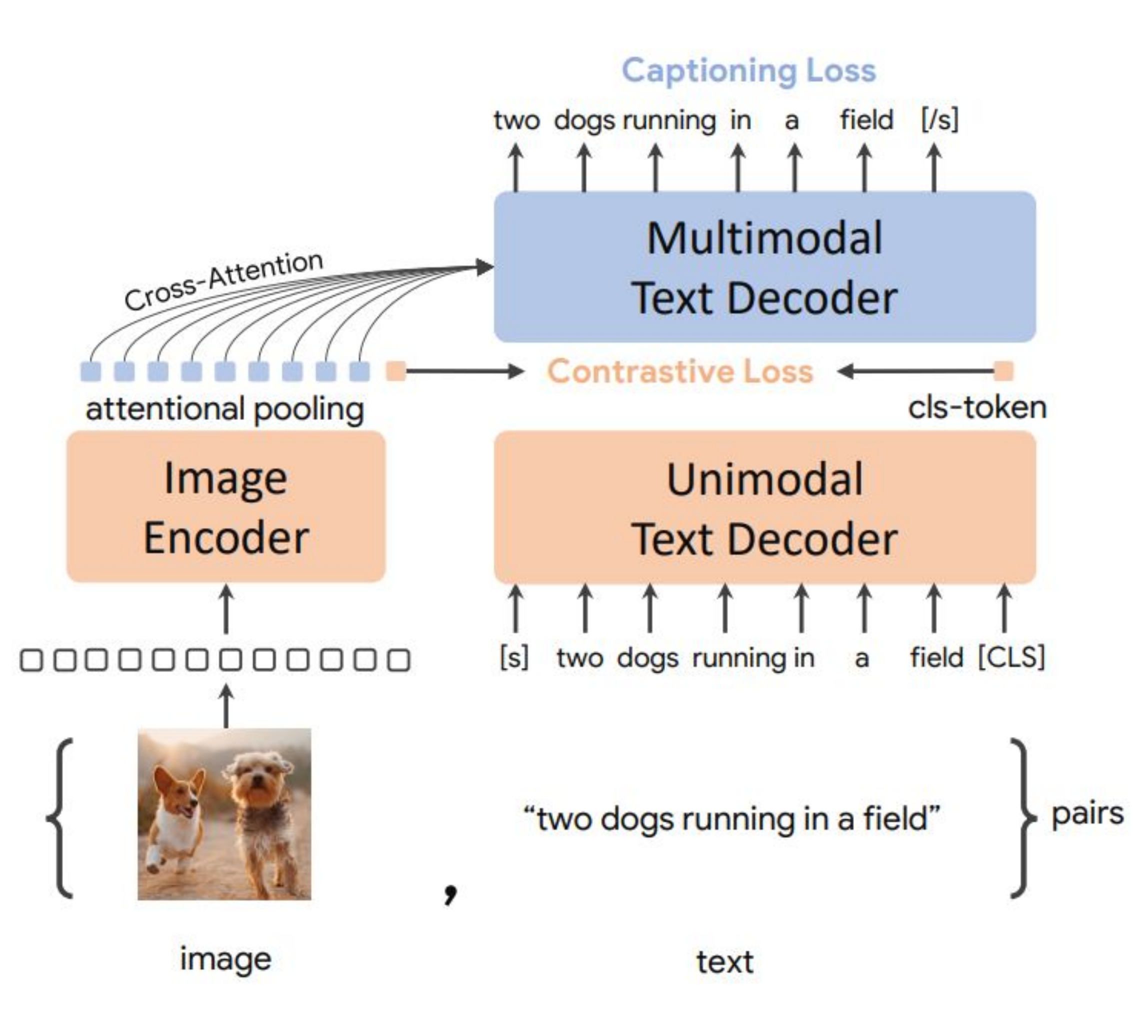
- Image Encoder: Image -> embedding
- Unimodal Text Decoder: Text -> embedding
- Multimodal Text Decoder: Image embedding + previous text tokens embedding -> next word
- Loss: Weighted sum of Co Loss and Ca Loss



$$\mathcal{L}_{\text{CoCa}} = \lambda_{\text{Con}} \cdot \mathcal{L}_{\text{Con}} + \lambda_{\text{Cap}} \cdot \mathcal{L}_{\text{Cap}},$$

Contrastive Captioners

Question:
For what task has CoCa been designed?

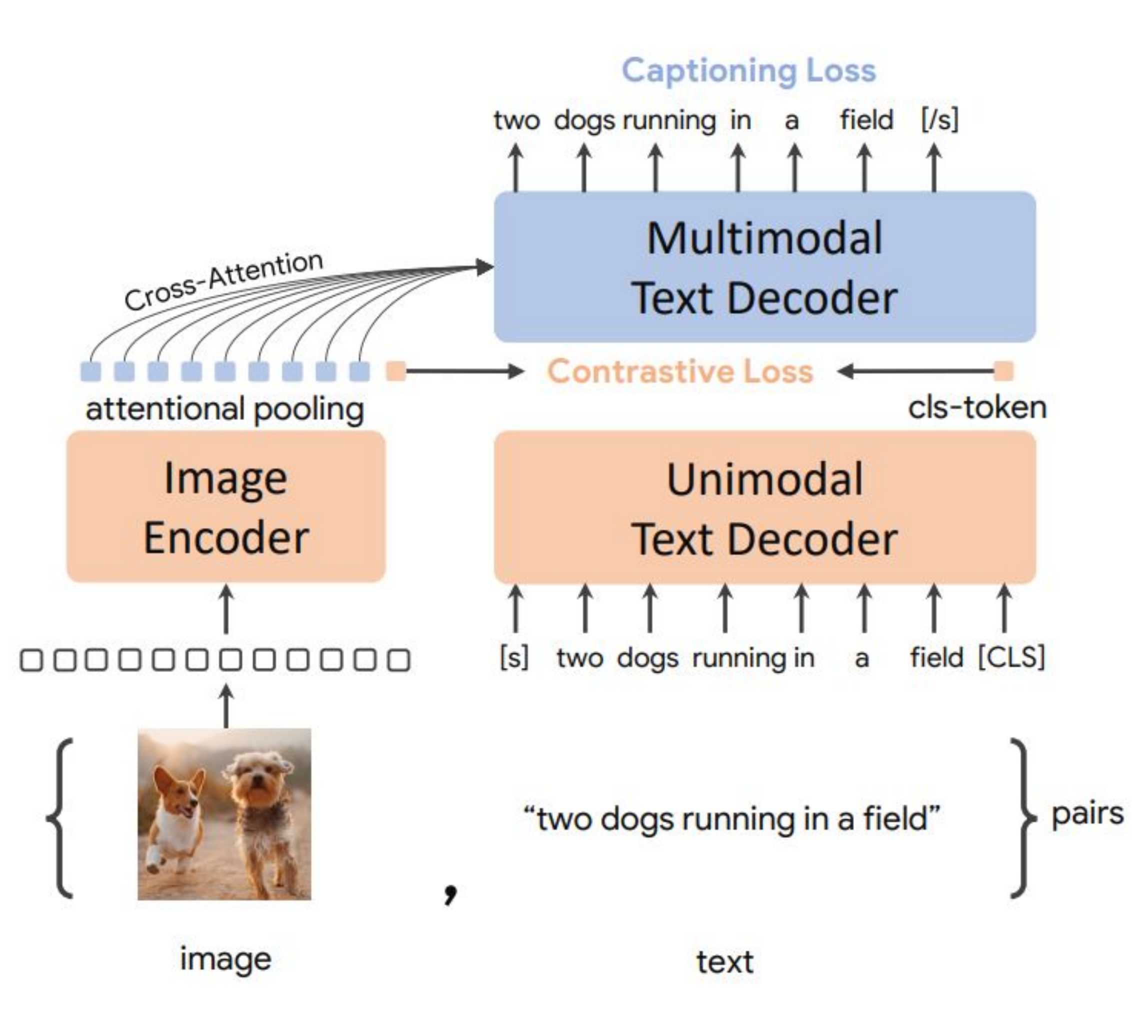


Contrastive Captioners

Question:
For what task has CoCa been designed?

Answer:
Trick question. Training of these three modules.

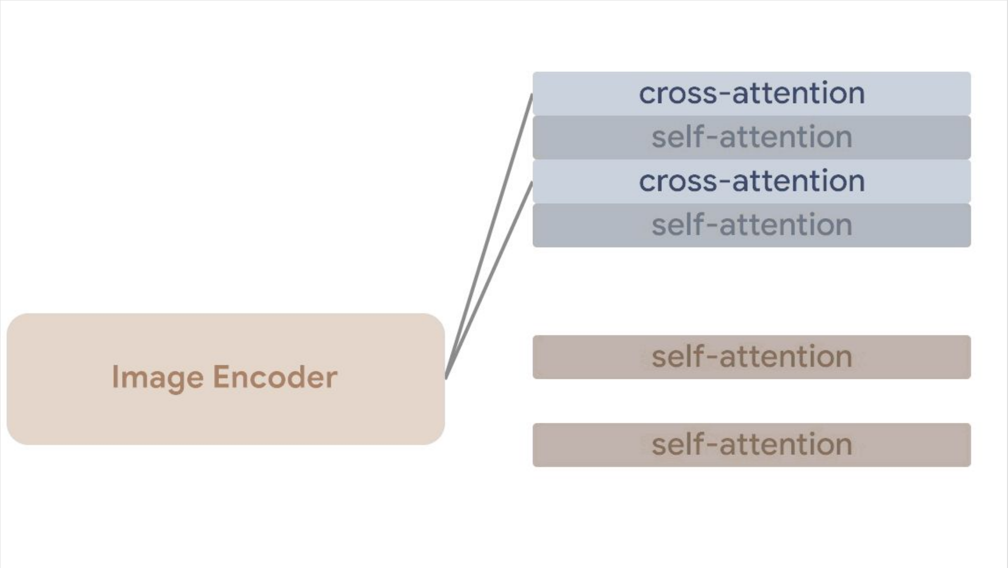
CoCa is a massive pretrained model for image-text tasks.
Data: JFT-3B and ALIGN.



CoCa structure

Overall: 2.1B parameters.

- Image Encoder: ViT, 40 layers, 1B parameters.
- Text Decoders: 18 self-attention layers both, 1.1B parameters.
- Multimodal also has cross-attention layers in-between



How to use CoCa (for classification)

Step 1: Download pretrained model.
Step 2: Add a classifier layer.

(Frozen Evaluation)
Step 3a: Freeze all weights in encoder and add an additional layer after encoder to balance feature weights.

(Finetuning)
Step 3b: Finetune encoder with task-specific data.

Step 4:Profit.

How to use CoCa (for classification)

Step 1: Download pretrained model.
Step 2: Add a classifier layer.

(Frozen Evaluation)
Step 3a: Freeze all weights in encoder and add an additional layer after encoder to balance feature weights.

(Finetuning)
Step 3b: Finetune encoder with task-specific data.

Step 4:Profit.

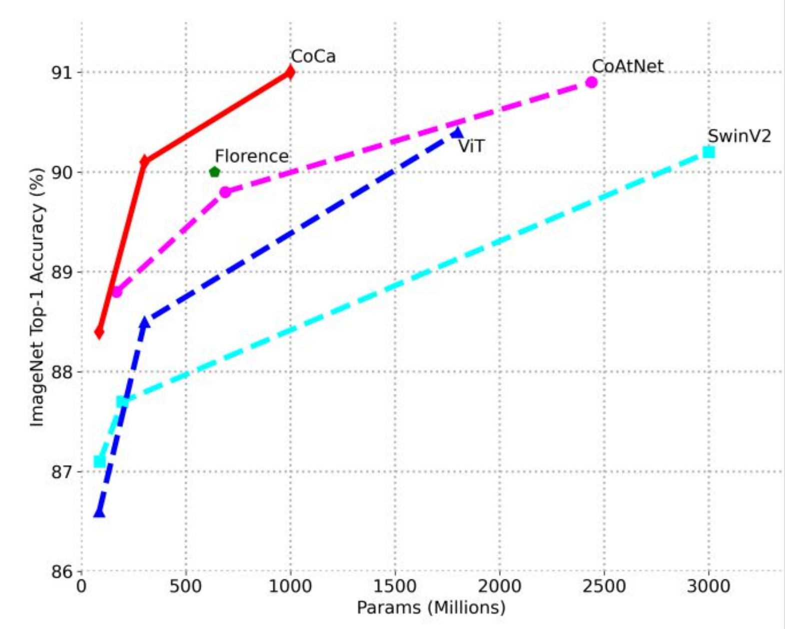
ImageNet Accuracy:

#1 Finetuned Coca (2022) - 91%
#6 Frozen Coca (2022) - 90.6%

Why?

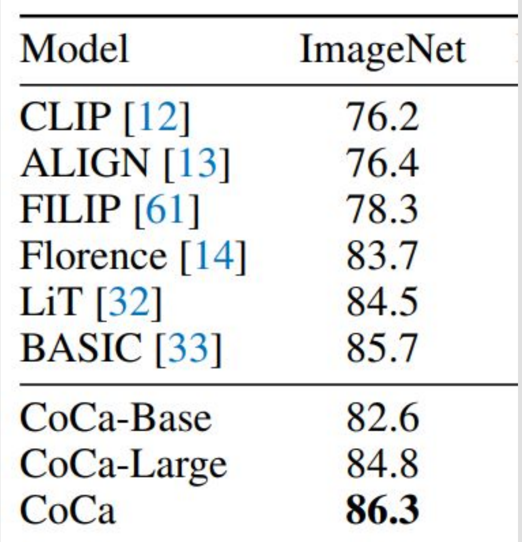
Why does Coca perform so well?

Different way of learning: Coca uses text training signals, while training the image encoder, thus learning better image representations.



(a) Finetuned ImageNet Top-1 Accuracy.

ImageNet zero-shot classification accuracy table.



Model	ImageNet
CLIP [12]	76.2
ALIGN [13]	76.4
FILIP [61]	78.3
Florence [14]	83.7
LiT [32]	84.5
BASIC [33]	85.7
CoCa-Base	82.6
CoCa-Large	84.8
CoCa	**86.3**

State-of-the-art zero-shot learning.

CoCa-Base and CoCa-Large - less layers in modules.

Types of tasks CoCa can solve

1. Image Classification.
2. Describing Images and Videos.
3. VQA (Visual Question Answering)
4. Basically anything with image and text in it.

a hand holding a san francisco 49ers football

a row of cannons with the eiffel tower in the background

a white van with a license plate that says we love flynn

a person sitting on a wooden bridge holding an umbrella

a truck is reflected in the side mirror of a car

How to use CoCa (for everything else)

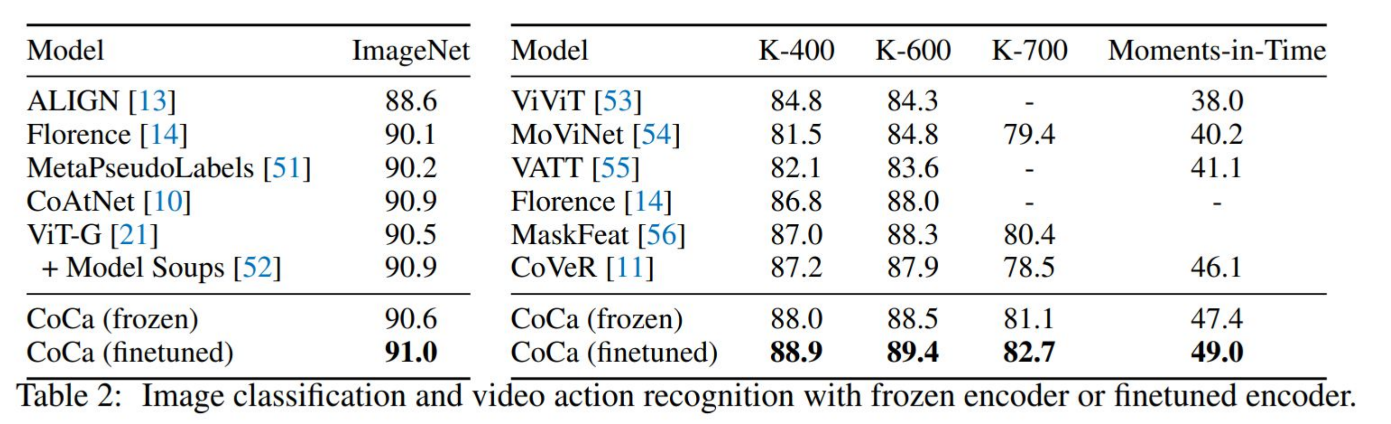
Step 1: Download pretrained model.

VQA: Step 2: Train a linear classifier: final decoder output -> classes of most common answers in training data

Video: Step 2: Get several frames from the video.
Step 3: Run them through encoder.
Step 4: Attention pooling + Cross-entropy

Step Whatever:Profit.

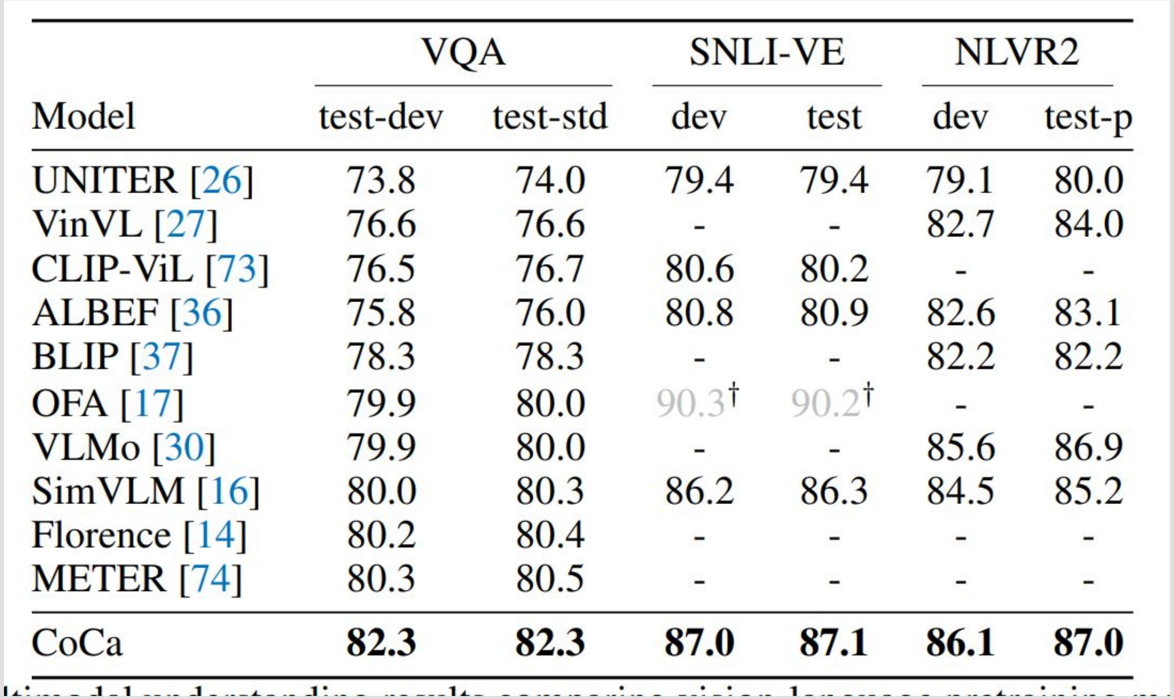
Fun tables with CoCa



Model	ImageNet
ALIGN [13]	88.6
Florence [14]	90.1
MetaPseudoLabels [51]	90.2
CoAtNet [10]	90.9
ViT-G [21]	90.5
+ Model Soups [52]	90.9
CoCa (frozen)	90.6
CoCa (finetuned)	**91.0**

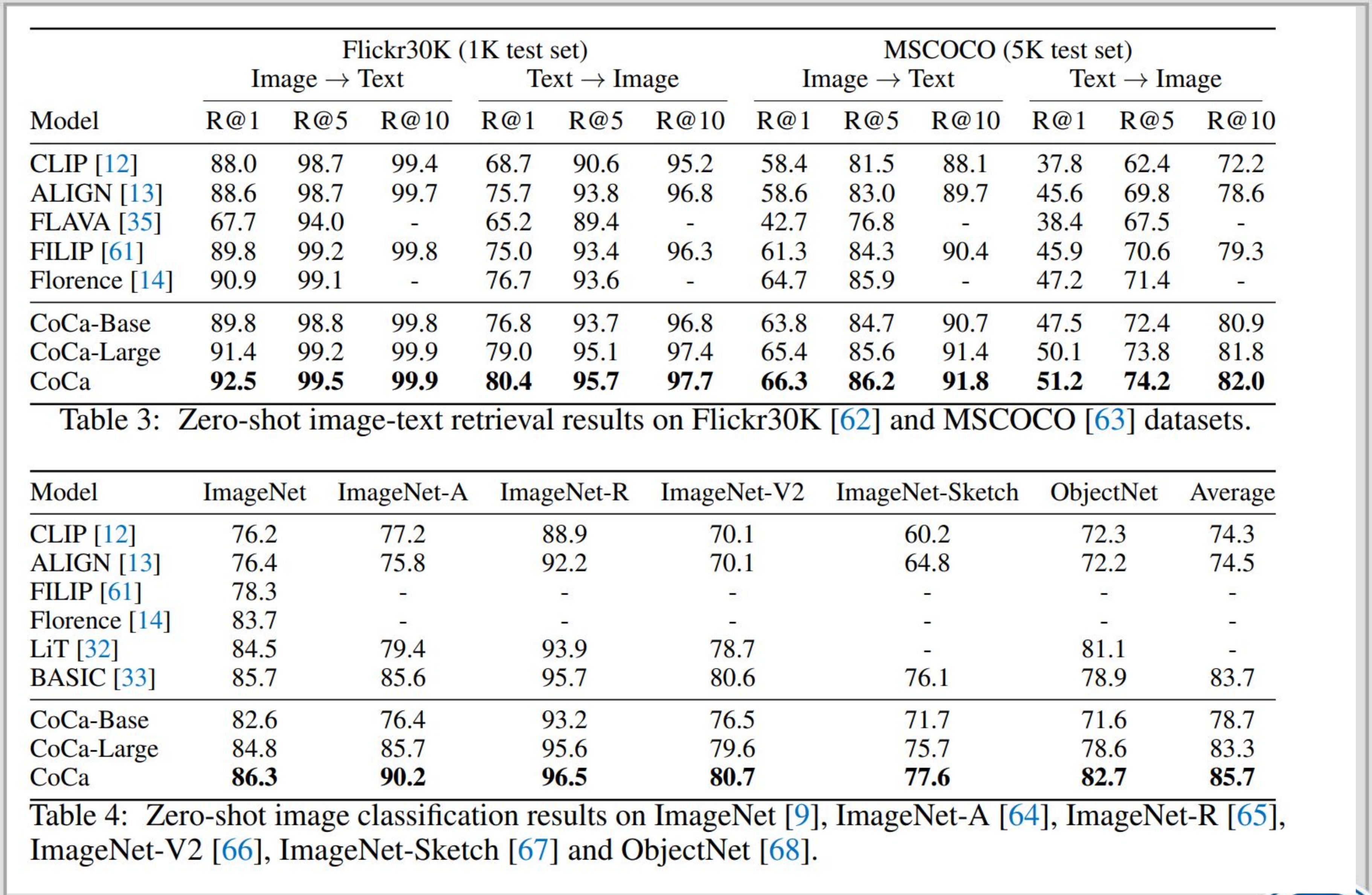
Model	K-400	K-600	K-700	Moments-in-Time
ViViT [53]	84.8	84.3	-	38.0
MoViNet [54]	81.5	84.8	79.4	40.2
VATT [55]	82.1	83.6	-	41.1
Florence [14]	86.8	88.0	-	-
MaskFeat [56]	87.0	88.3	80.4	
CoVeR [11]	87.2	87.9	78.5	46.1
CoCa (frozen)	88.0	88.5	81.1	47.4
CoCa (finetuned)	**88.9**	**89.4**	**82.7**	**49.0**

Table 2: Image classification and video action recognition with frozen encoder or finetuned encoder.



	VQA		SNLI-VE		NLVR2	
Model	test-dev	test-std	dev	test	dev	test-p
UNITER [26]	73.8	74.0	79.4	79.4	79.1	80.0
VinVL [27]	76.6	76.6	-	-	82.7	84.0
CLIP-ViL [73]	76.5	76.7	80.6	80.2	-	-
ALBEF [36]	75.8	76.0	80.8	80.9	82.6	83.1
BLIP [37]	78.3	78.3	-	-	82.2	82.2
OFA [17]	79.9	80.0	$90.3^{\dagger}$	$90.2^{\dagger}$	-	-
VLMo [30]	79.9	80.0	-	-	85.6	86.9
SimVLM [16]	80.0	80.3	86.2	86.3	84.5	85.2
Florence [14]	80.2	80.4	-	-	-	-
METER [74]	80.3	80.5	-	-	-	-
CoCa	**82.3**	**82.3**	**87.0**	**87.1**	**86.1**	**87.0**

ltimodal understanding results comparing vision-language pretraining m



	Flickr30K (1K test set)						MSCOCO (5K test set)					
	Image → Text			Text → Image			Image → Text			Text → Image		
Model	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
CLIP [12]	88.0	98.7	99.4	68.7	90.6	95.2	58.4	81.5	88.1	37.8	62.4	72.2
ALIGN [13]	88.6	98.7	99.7	75.7	93.8	96.8	58.6	83.0	89.7	45.6	69.8	78.6
FLAVA [35]	67.7	94.0	-	65.2	89.4	-	42.7	76.8	-	38.4	67.5	-
FILIP [61]	89.8	99.2	99.8	75.0	93.4	96.3	61.3	84.3	90.4	45.9	70.6	79.3
Florence [14]	90.9	99.1	-	76.7	93.6	-	64.7	85.9	-	47.2	71.4	-
CoCa-Base	89.8	98.8	99.8	76.8	93.7	96.8	63.8	84.7	90.7	47.5	72.4	80.9
CoCa-Large	91.4	99.2	99.9	79.0	95.1	97.4	65.4	85.6	91.4	50.1	73.8	81.8
CoCa	**92.5**	**99.5**	**99.9**	**80.4**	**95.7**	**97.7**	**66.3**	**86.2**	**91.8**	**51.2**	**74.2**	**82.0**

Table 3: Zero-shot image-text retrieval results on Flickr30K [62] and MSCOCO [63] datasets.

Model	ImageNet	ImageNet-A	ImageNet-R	ImageNet-V2	ImageNet-Sketch	ObjectNet	Average
CLIP [12]	76.2	77.2	88.9	70.1	60.2	72.3	74.3
ALIGN [13]	76.4	75.8	92.2	70.1	64.8	72.2	74.5
FILIP [61]	78.3	-	-	-	-	-	-
Florence [14]	83.7	-	-	-	-	-	-
LiT [32]	84.5	79.4	93.9	78.7	-	81.1	-
BASIC [33]	85.7	85.6	95.7	80.6	76.1	78.9	83.7
CoCa-Base	82.6	76.4	93.2	76.5	71.7	71.6	78.7
CoCa-Large	84.8	85.7	95.6	79.6	75.7	78.6	83.3
CoCa	**86.3**	**90.2**	**96.5**	**80.7**	**77.6**	**82.7**	**85.7**

Table 4: Zero-shot image classification results on ImageNet [9], ImageNet-A [64], ImageNet-R [65], ImageNet-V2 [66], ImageNet-Sketch [67] and ObjectNet [68].

Bibliography

CoCa - https://arxiv.org/pdf/2205.01917.pdf

Learning Transferable Visual Models From Natural Language Supervision (CLIP included)- https://arxiv.org/pdf/2103.00020.pdf

Only CLIP - https://openai.com/blog/clip/

ImageNet ranking - https://paperswithcode.com/sota/image-classification-on-imagenet

Cross-Attention - https://vaclavkosar.com/ml/cross-attention-in-transformer-architecture