CLIP Embeddings

Contrastive Language–Image Pre-training

Learning Transferable Visual Models From Natural Language Supervision

---

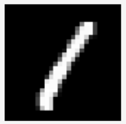
Alec Radford [*1] **Jong Wook Kim** [*1] **Chris Hallacy** [1] **Aditya Ramesh** [1] **Gabriel Goh** [1] **Sandhini Agarwal** [1]
Girish Sastry [1] **Amanda Askell** [1] **Pamela Mishkin** [1] **Jack Clark** [1] **Gretchen Krueger** [1] **Ilya Sutskever** [1]

Presented by:

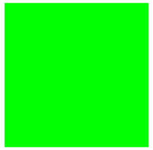
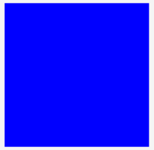
John Tan Chong Min

Demo of CLIP Embeddings

Pros: Broad Category of Image

Image / Text	Best Matched Text	A photo of a guacamole	A picture of a fight	A picture of a girl	A picture of pikachu	a picture of an apple	a picture of a rainbow apple	a picture of one
	A photo of a guacamole	**0.33**	0.20	0.20	0.20	0.22	0.22	0.23
	A picture of a fight	0.13	**0.25**	0.17	0.17	0.16	0.12	0.19
	A picture of a girl	0.13	0.20	**0.25**	0.20	0.19	0.16	0.22
	A picture of pikachu	0.19	0.21	0.22	**0.33**	0.22	0.19	0.22
	a picture of an apple	0.20	0.22	0.22	0.19	**0.31**	0.29	0.24
	a picture of a rainbow apple	0.16	0.19	0.19	0.17	0.27	**0.27**	0.21
	a picture of one	0.20	0.23	0.23	0.21	0.23	0.20	**0.26**

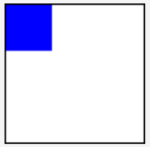
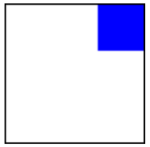
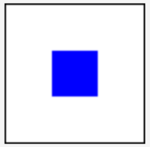
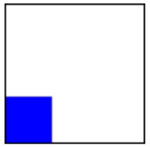
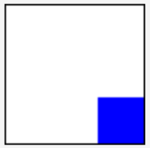
Pros: Color

Image / Text	Best Matched Text	a picture of red background	a picture of green background	a picture of blue background	a picture of black background	a picture of white background
	a picture of red background	**0.31**	0.25	0.26	0.27	0.28
	a picture of green background	0.26	**0.31**	0.26	0.27	0.28
	a picture of blue background	0.26	0.25	**0.31**	0.27	0.27
	a picture of black background	0.25	0.26	0.26	**0.29**	0.28
	a picture of white background	0.22	0.22	0.22	0.22	**0.25**

Pros: Text Detection

Image / Text	Best Matched Text	a picture of "bye"	a picture of "clip"	a picture of "clips"	a picture of "1"	a picture of "2"
bye	a picture of "bye"	**0.32**	0.21	0.20	0.22	0.22
clip	a picture of "clip"	0.24	**0.32**	0.30	0.23	0.22
clips	a picture of "clips"	0.23	0.31	**0.32**	0.23	0.22
1	a picture of "1"	0.24	0.22	0.22	**0.27**	0.25
2	a picture of "2"	0.23	0.22	0.22	0.26	**0.27**

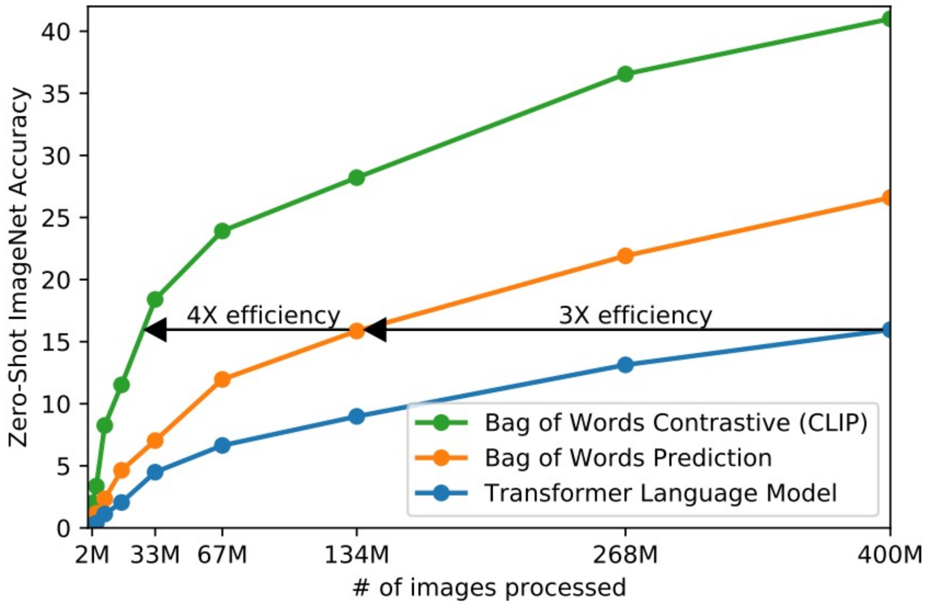
Cons: Position

Image / Text	Best Matched Text	a picture of blue square at top left	a picture of blue square at top right	a picture of blue square at center	a picture of blue square at bottom left	a picture of blue square at bottom right
	a picture of blue square at top right	0.29	**0.30**	0.29	0.29	0.30
	a picture of blue square at top right	0.29	**0.30**	0.29	0.28	0.29
	a picture of blue square at center	0.30	0.31	**0.31**	0.29	0.30
	a picture of blue square at top right	0.27	**0.28**	0.27	0.26	0.27
	a picture of blue square at top right	0.27	**0.28**	0.27	0.27	0.28

Key takeaways

- Large-scale web-scale learning is better than dataset-specific training
- **Text:** LLM systems using unsupervised next-token prediction and can scale without labels
- **Multimodal:** Text-image systems require Caption-Image pairs, but are easily obtainable with internet data

Key insight: Comparing in latent/abstraction space better than predicting in self-supervised manner for cross-domain mapping



- Image domain is of high-dimensionality
- Can be hard to predict image-based caption tokens exactly
- Bag of words / Contrastive learning may help reduce demands on prediction by abstracting in latent space

Dataset

- 400 million text-image pairs

- Text must contain one of 500,000 query words
  - Query words are those occurring at least 100 times in English version of Wikipedia
  - **My Opinion: May mean cross-language support and rare domain-specific words may not be covered**

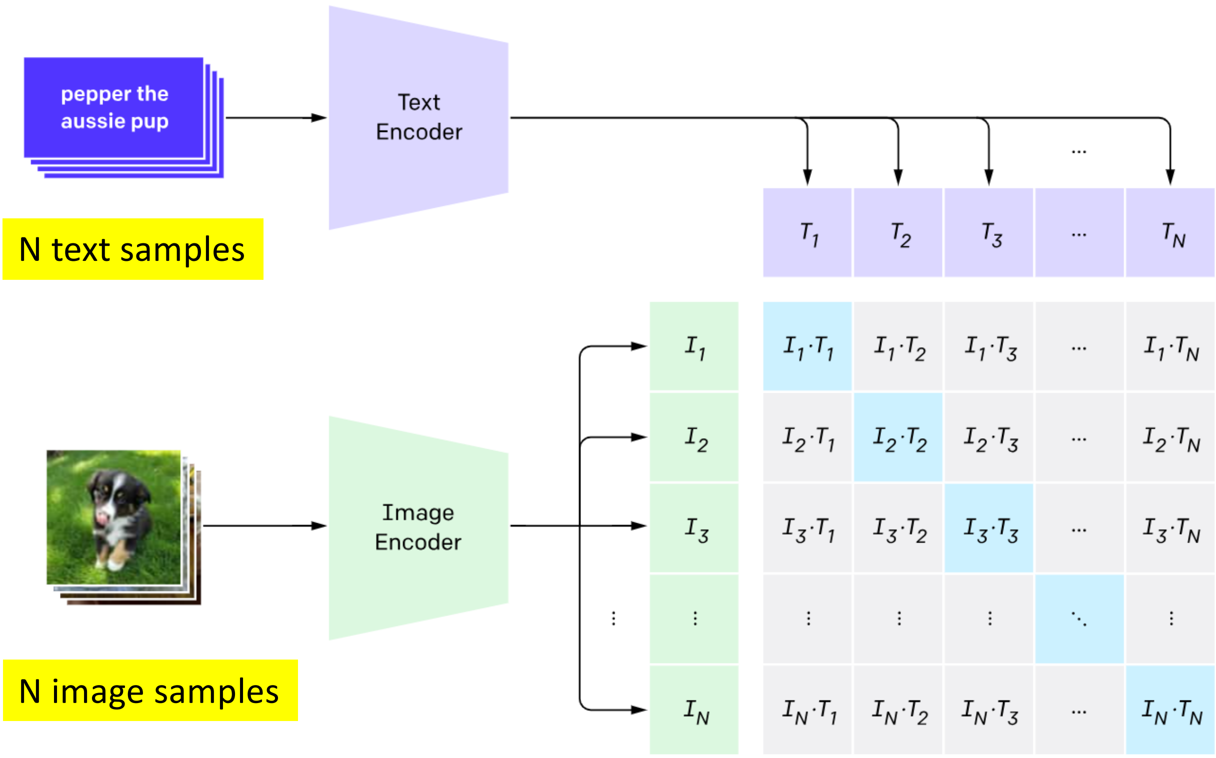
- Class-balance results by including up to 20000 (image, text) pairs per query

- Able to perform wide set of tasks during pre-training including OCR, geo-localization, action recognition, classification

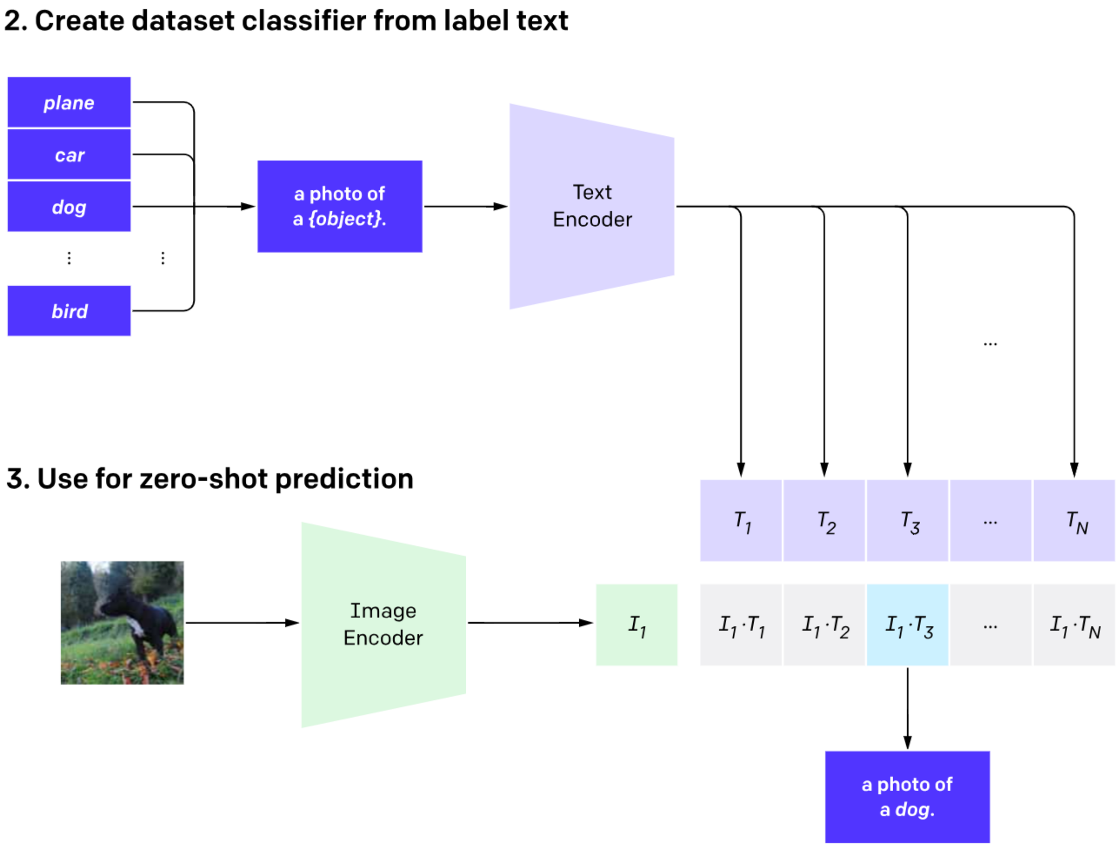
Final Architecture

- Text Encoder: BoW / GPT-2
- Image Encoder: ResNet / ViT
- Embedding dimension: 512
- **Impt: Max sequence length capped at 76 tokens
- Training Time: The largest ResNet model, RN50x64, took **18 days** to train on 592 **V100 GPUs** while the largest Vision Transformer took **12 days** on **256 V100 GPUs**

1. Contrastive pre-training



- Use cosine similarity of embeddings as gauge of similarity
- Predict only those image-text mappings given in *N* samples
- **Objective:** Maximise cosine similarity of those in blue (true pairs), and minimize the rest



- Classification by CLIP can be done by having a list of **text embeddings** corresponding to each class, and mapping it to **image embeddings**

- **Note: Prediction is done in latent/abstraction space**

- **My thoughts: Coiuld performance be better with multiple abstraction spaces?**

Details for the brave

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

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

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

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

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

Figure 3. Numpy-like pseudocode for the core of an implementation of CLIP.

- Image and text encoder can be replaced with anything that takes in image/text respectively and outputs embeddings
- Uses cross-entropy loss
  - For each image, ensure corresponding text is predicted highly
  - For each text, ensure corresponding image is predicted highly

CLIP

Train on many diverse datasets
Effective across many tasks

Good performance across 30 datasets

Food101

guacamole (90.1%) Ranked 1 out of 101 labels

- ✓ 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.

SUN397

television studio (90.2%) Ranked 1 out of 397 labels

- ✓ 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**.

Youtube-BB

airplane, person (89.0%) Ranked 1 out of 23 labels

- ✓ a photo of a **airplane**.
- × a photo of a **bird**.
- × a photo of a **bear**.
- × a photo of a **giraffe**.
- × a photo of a **car**.

EuroSAT

annual crop land (46.5%) Ranked 4 out of 10 labels

- × a centered satellite photo of **permanent crop land**.
- × a centered satellite photo of **pasture land**.
- × a centered satellite photo of **highway or road**.
- ✓ a centered satellite photo of **annual crop land**.
- × a centered satellite photo of **brushland or shrubland**.

Issues of Text Captioning for Classification

- Polysemy: Class Names may not have full context
  - E.g. ImageNet classes uses same word “Crane” for both construction cranes and cranes that fly.
  - **What this means: If you are using it for classification, try to provide more context for the classes, e.g. specify location, context of the class**

- Single word classes not common in pre-training captions:
  - To help bridge this distribution gap, we found that using the prompt template “A photo of a {label}.” to be a good default
  - **What this means: When you are using it for your tasks, try to match it to image captions in the wild**

Using more diverse and context-dependent text captioning helps

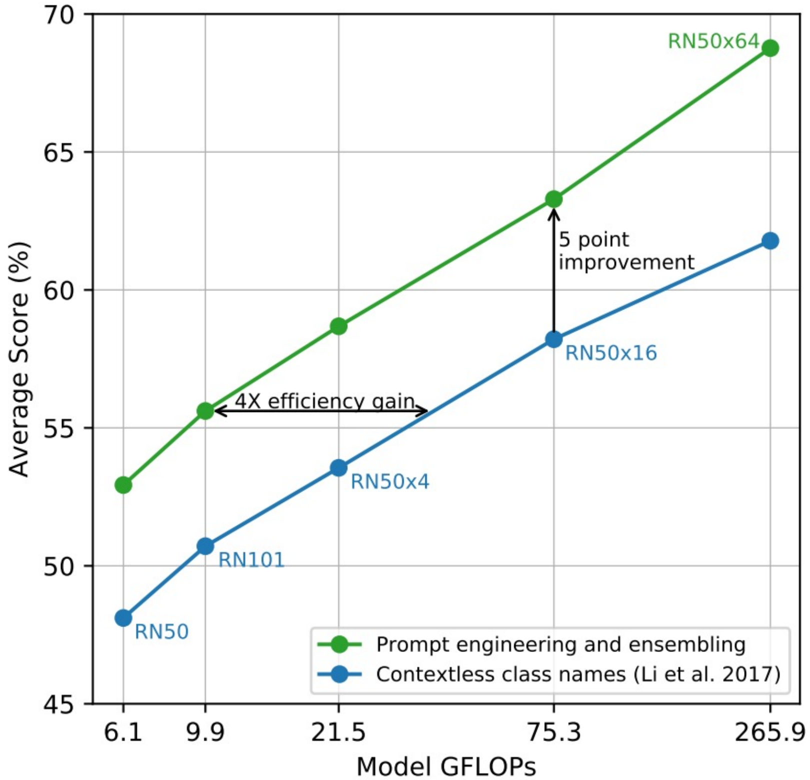


Figure 4. **Prompt engineering and ensembling improve zero-shot performance.** Compared to the baseline of using contextless class names, prompt engineering and ensembling boost zero-shot classification performance by almost 5 points on average across 36 datasets. This improvement is similar to the gain from using 4 times more compute with the baseline zero-shot method but is "free" when amortized over many predictions.

Prompt-Engineering for Datasets

- Specifying the type of dataset in text captions helps
  - Oxford-IIIT Pets: “A photo of a {label}, a type of pet.”
  - Food101: “A photo of a {label}, a type of food.”
  - FGVC Aircraft: “A photo of a {label}, a type of aircraft.”
  - OCR datasets: Put a quote around text or number to recognise
  - Satellite: “A satellite photo of a {label}”

Ensembling Text Embeddings

- Average embedding over multiple similar prompts:
  - “A photo of a big {label}”
  - ”A photo of a small {label}”
- ImageNet ensembled over 80 context prompts
- **My thoughts: Don’t use text embeddings for specific image features – it probably is lost over ensembling**

For natural image-type datasets without specialised knowledge, zero-shot CLIP is competitive

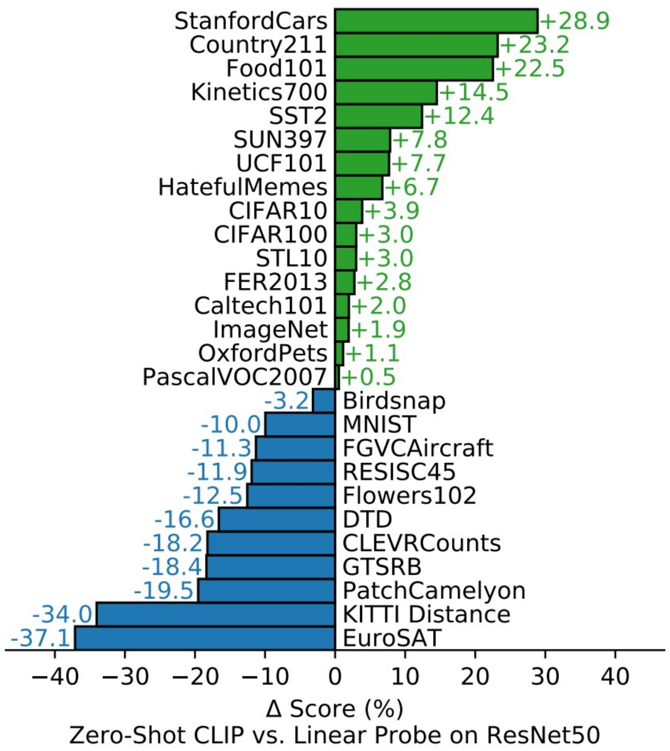


Figure 5. **Zero-shot CLIP is competitive with a fully supervised baseline.** Across a 27 dataset eval suite, a zero-shot CLIP classifier outperforms a fully supervised linear classifier fitted on ResNet-50 features on 16 datasets, including ImageNet.

My thoughts: Web-scale data may be able to augment a limited training set

Using natural text meaning for class labels can help with transfer learning

For natural image-type datasets without specialised knowledge, zero-shot CLIP is competitive

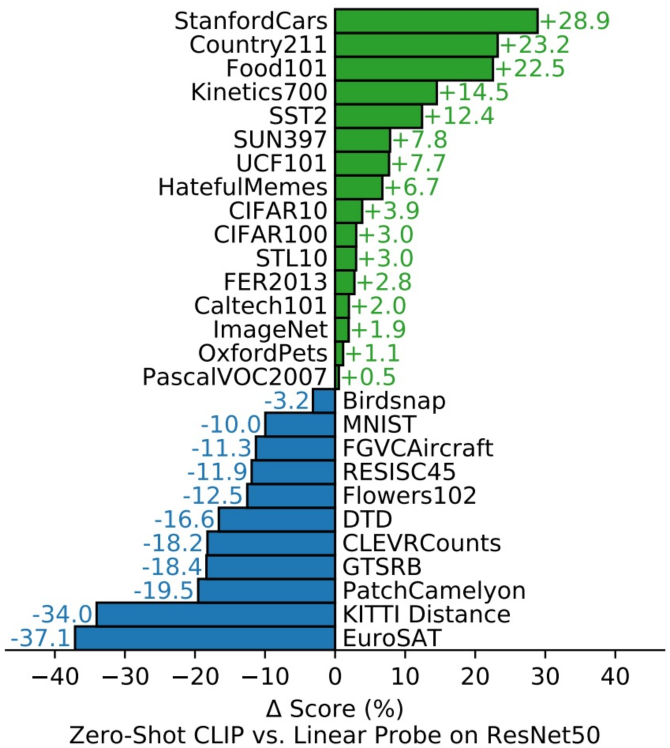


Figure 5. **Zero-shot CLIP is competitive with a fully supervised baseline.** Across a 27 dataset eval suite, a zero-shot CLIP classifier outperforms a fully supervised linear classifier fitted on ResNet-50 features on 16 datasets, including ImageNet.

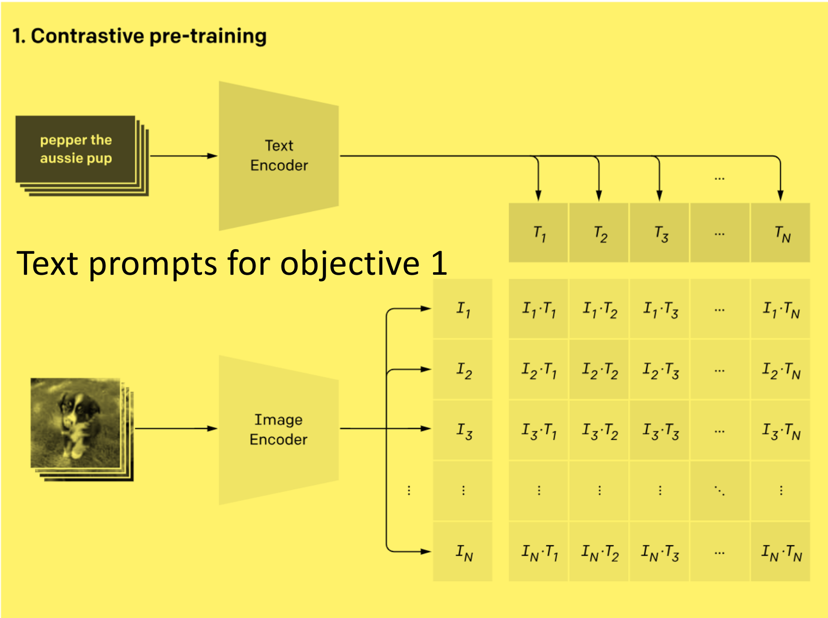
CLIP is weak at specialised tasks:

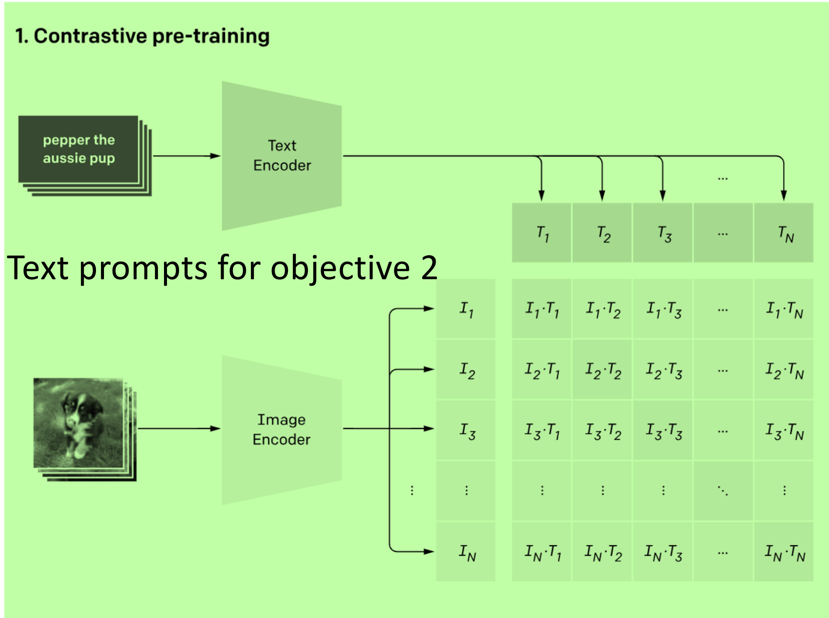
- Satellite image classification (EuroSAT and RESISC45)
- Lymph node tumor detection (PatchCamelyon)
- Counting objects in synthetic scenes (CLEVRCounts)
- German traffic sign recognition (GTSRB)
- Recognizing distance to the nearest car (KITTI Distance)

My thoughts: LLMs are not great for such tasks either – rule-based tasks or specialized tasks may perform better without interference from web-scale data

Food for thought: Multiple Abstraction Spaces?

Problem: Many potential objectives for similarity





Objective 1
e.g. background

Objective 2
e.g. objects

Choose the right objectives for your use cases

Questions to Ponder

- What does it mean to be similar in image space?
- Why would someone use image embeddings to find an image, as compared to using matching text embeddings to text abstracted from an image?
- CLIP is trained with text and image encoder from scratch. Why not start the training with pre-trained text embeddings, and then try to base the image embeddings off these?
- Will better text and image encoders help with better latent/abstraction spaces? What about dimension of latent/abstraction space?