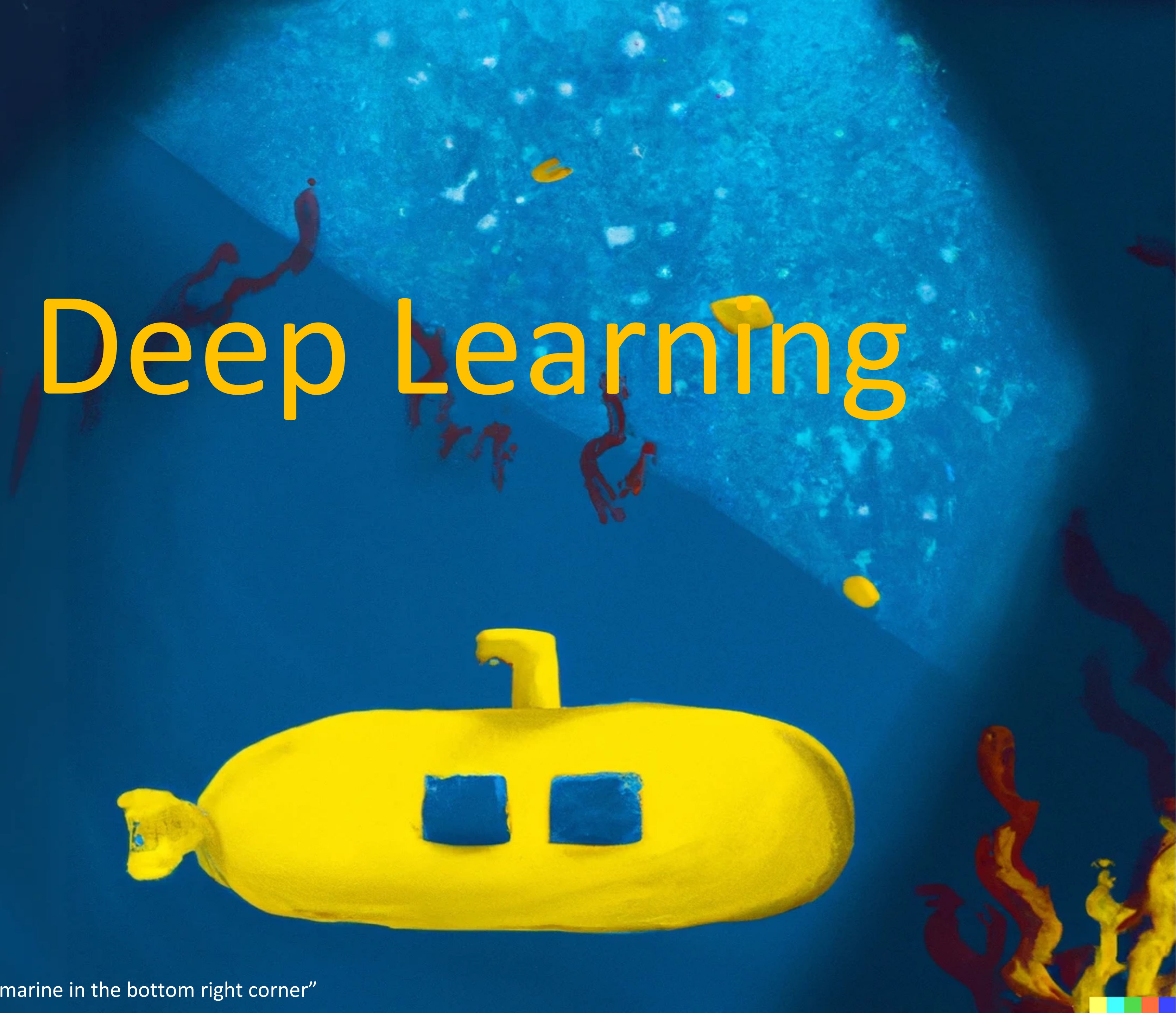
CSCI 1470/2470
Spring 2024

Guest Lecture:
Michal
Golovanevsky

April 3rd, 2024
Wednesday

Deep Learning



DALL-E 2 prompt “a painting of deep underwater with a yellow submarine in the bottom right corner”

About Me

3rd year Computer Science PhD at Brown!

Research Interests: Deep learning, multimodal learning, clinical decision support, interpretable ML

Advisors: Ritambhara Singh and Carsten Eickhoff

Website:https://michalg04.github.io/
Email: michalg@brown.edu

B.S. 2016-2020

PhD 2021 - Present

Today’s goal – understand OpenAI’s CLIP model

(1) CLIP at a high level

(2) Zero-shot Learning

(3) Contrastive Learning

(4) Walkthrough of results and CLIP’s capabilities

CLIP - Contrastive Language-Image Pre-training

- CLIP is a multi-modal (language-image) model
- Uses contrastive learning
- CLIP is a zero-shot classifier
- In 2021, CLIP beat unsupervised and supervised baselines on many datasets
- Leverages a huge amount of paired data (“web-scale”)
- While contrastive learning was not new at the time, it was never done at this multimodal scale

OpenAI

CLIP is capable of…

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.

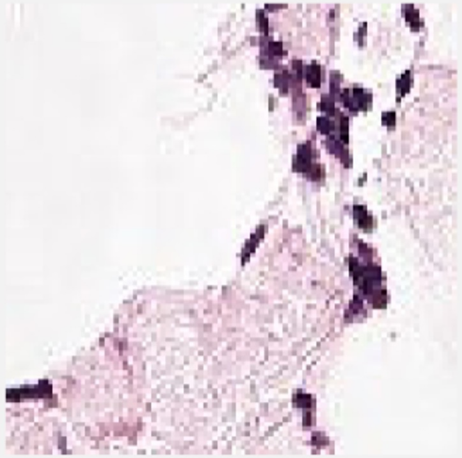
× a photo of **edamame**, a type of food.

× a photo of **tuna tartare**, a type of fo

× a photo of **hummus**, a type of food.

PatchCamelyon (PCam)

healthy lymph node tissue (77.2%) Ranked 2 out of 2 labels

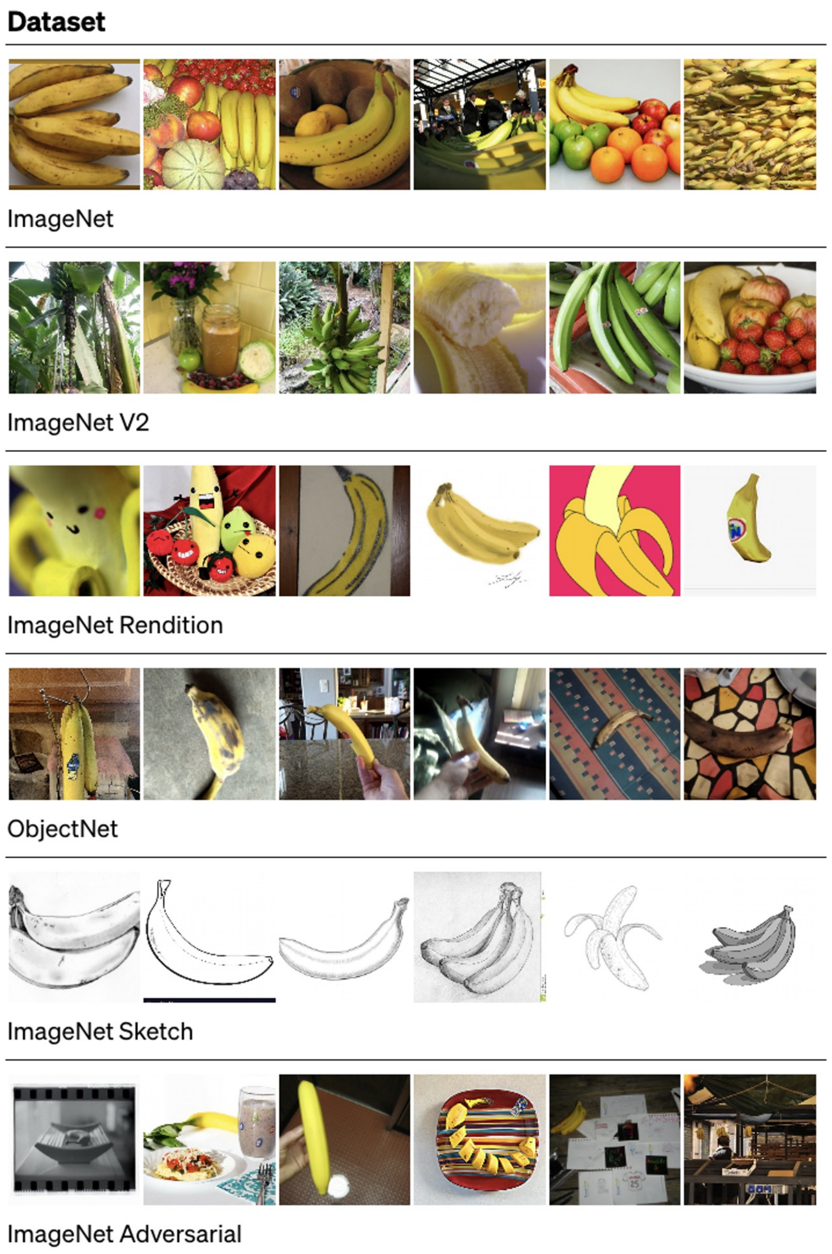


× this is a photo of **lymph node tumor tissue**

✓ this is a photo of **healthy lymph node tissue**

Motivation

- Limitations of prior image classification and captioning methods…
  - Costly datasets
  - Narrow
  - Poor real-world performance

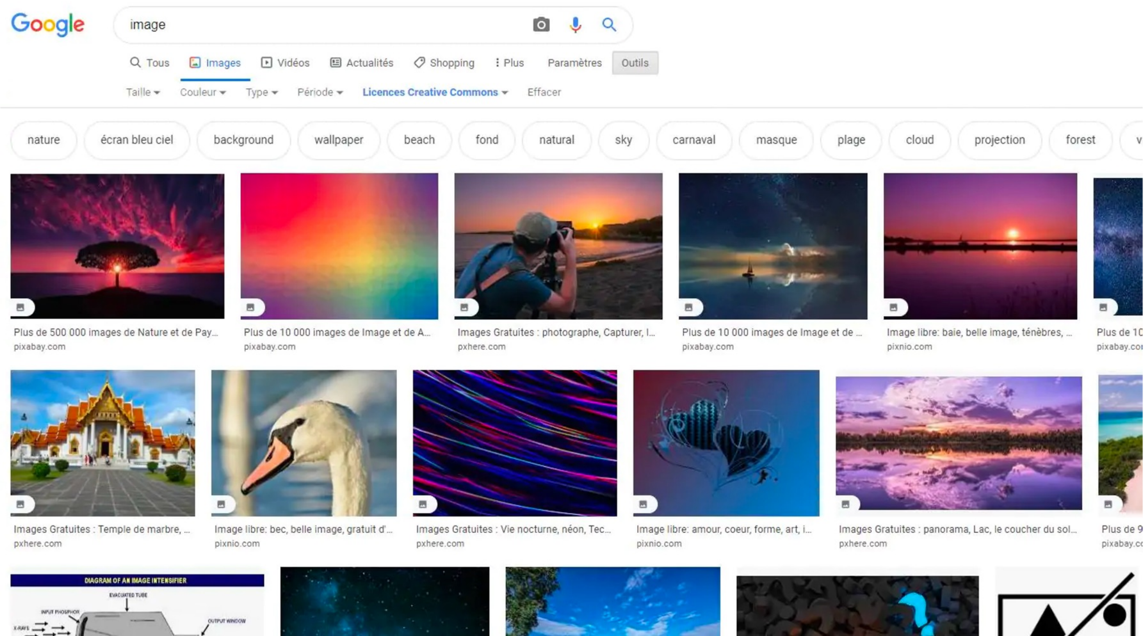


Costly Datasets

- Vision models have traditionally been trained on manually labeled datasets that are expensive to construct

- The ImageNet dataset required over 25,000 workers to annotate 14 million images

- In contrast, CLIP learns from text–image pairs that are already publicly available on the internet
    - 400 million pairs from the web



Narrow

- An ImageNet model is good at predicting the 1000 ImageNet categories, but that's all it can do "out of the box."

- If we wish to perform any other task, an ML practitioner needs to build a new dataset, add an output head, and fine-tune the model.

- In contrast, CLIP can be adapted to perform a wide variety of visual classification tasks without needing additional training examples

Poor Real-World Performance

- Deep learning has surpassed human abilities in a variety of benchmarks (tasks)

- Yet when deployed in the wild, their performance can be far below the expectation set by the benchmark

- In other words, there is a gap between “benchmark performance” and “real performance.”

Poor Real-World Performance

- Hypothesis: models “cheat” by only optimizing for performance on the benchmark
  - much like a student who passed an exam by studying only the questions on past years’ exams

- In contrast, the CLIP model can be evaluated on benchmarks without having to train on their data, so it can’t “cheat” in this manner
  - CLIP is a zero-shot learner!

Zero-shot Learning

- Zero-shot learning refers to the ability of a model to correctly make predictions for tasks it has not explicitly been trained for

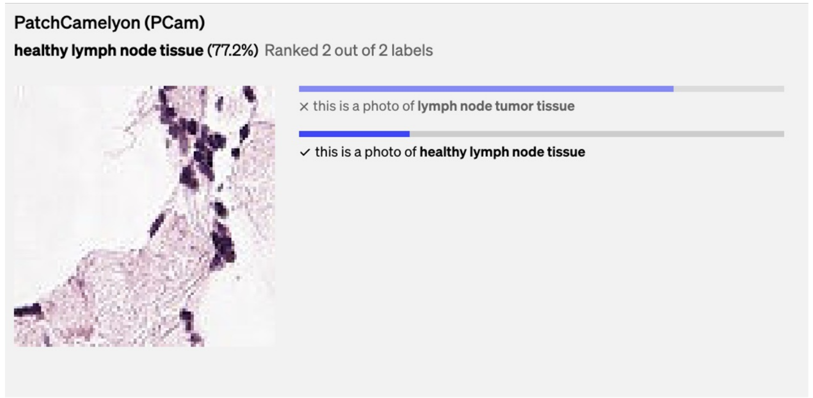
- It's called "zero-shot" because the model sees zero examples of the specific task during training

- Instead, it relies on a generalized understanding and representation of the data it was trained on, allowing it to make inferences about new, unseen tasks

Zero-shot Learning

- In the context of CLIP, zero-shot learning allows the model to understand and relate textual descriptions to images in ways it was not explicitly trained for

- This is possible because CLIP is trained on a vast amount of image-text pairs, learning a rich, multimodal space that generalizes well beyond its training data



Zero-shot Learning vs. Unsupervised Learning

- Unlike unsupervised learning, where models attempt to learn patterns from data without any labeled examples, zero-shot learning models are typically trained on large, labeled (ish) datasets

- The key difference is in application:
    - Unsupervised learning seeks to understand the structure of data without explicit labels
    - Zero-shot learning uses its pre-existing knowledge and understanding to make inferences about completely new and unseen tasks or data categories

Any questions on the motivation for CLIP or zero-shot learning?

CLIP - Road Map

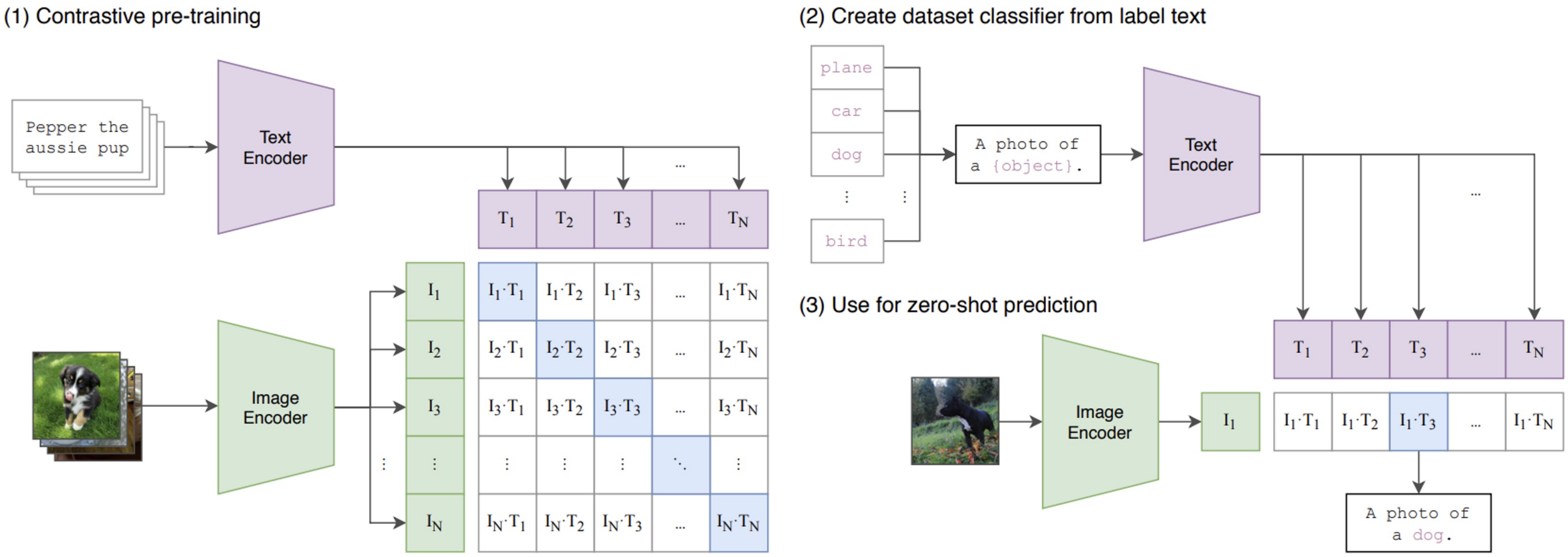
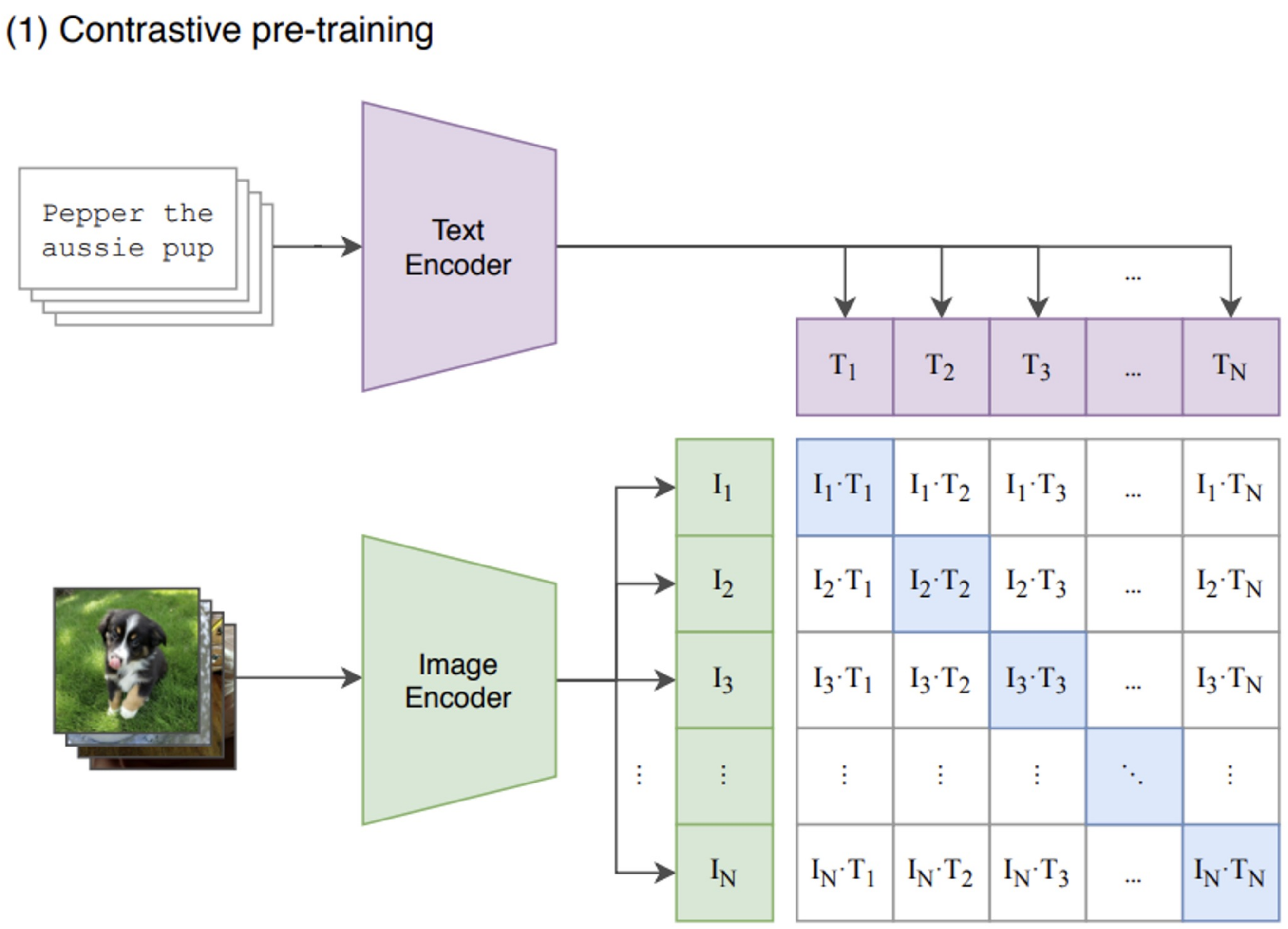


Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

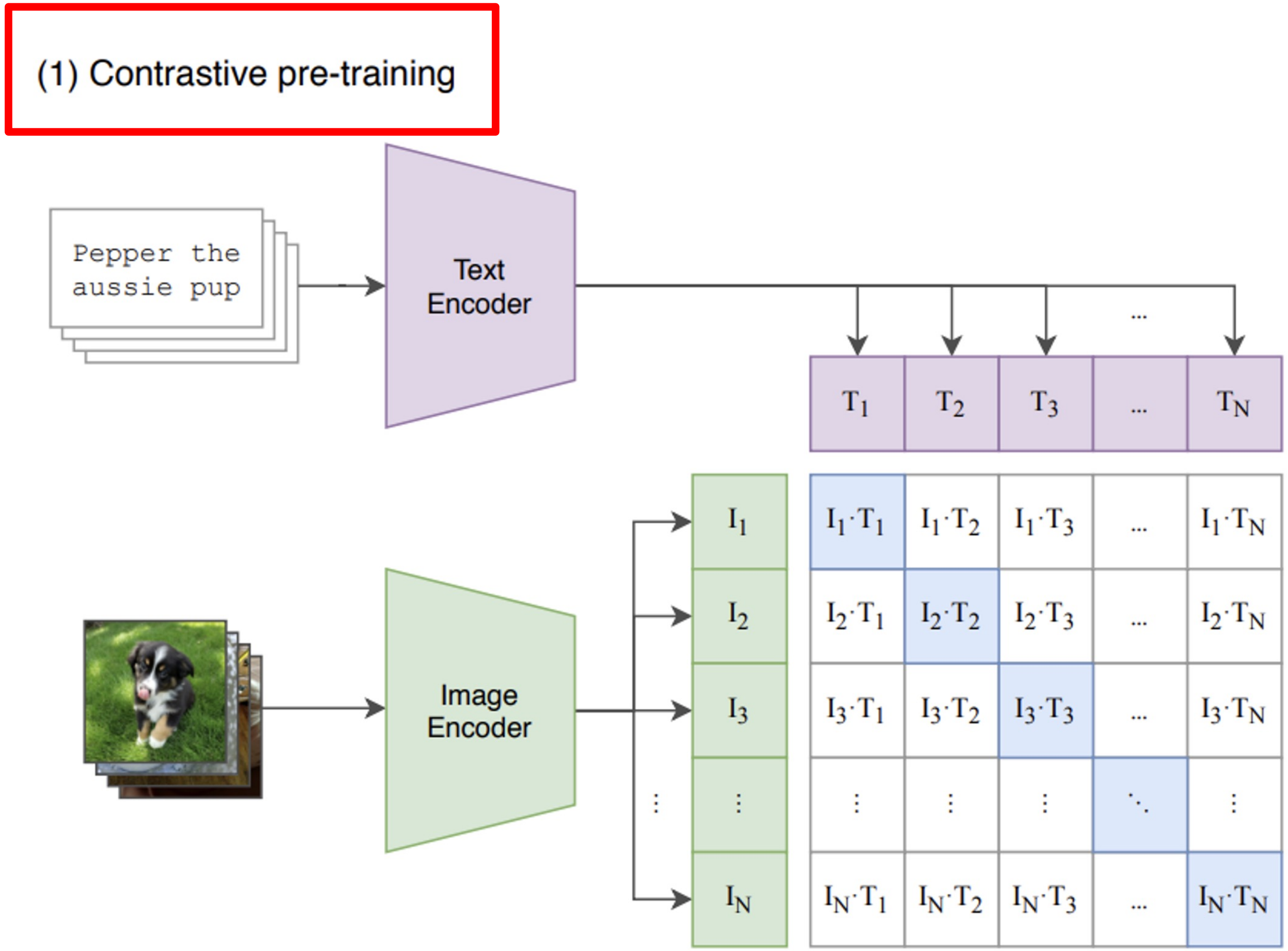
Let’s dive into CLIP step 1!



Important:

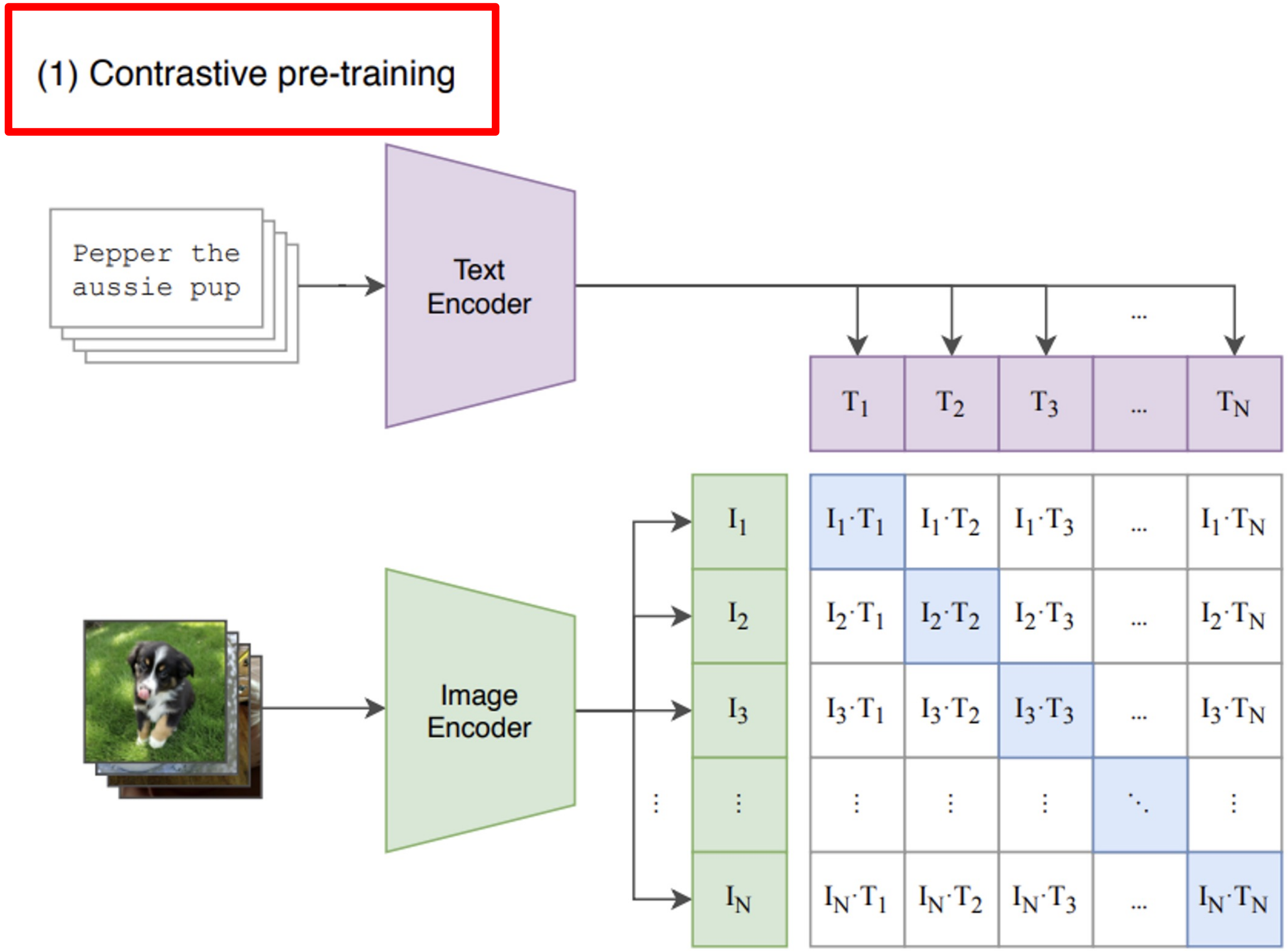
- CLIP uses data-data pairs for training, not data-label pairs!
- This does not require manual annotation

So.. what is “contrastive pre-training”?



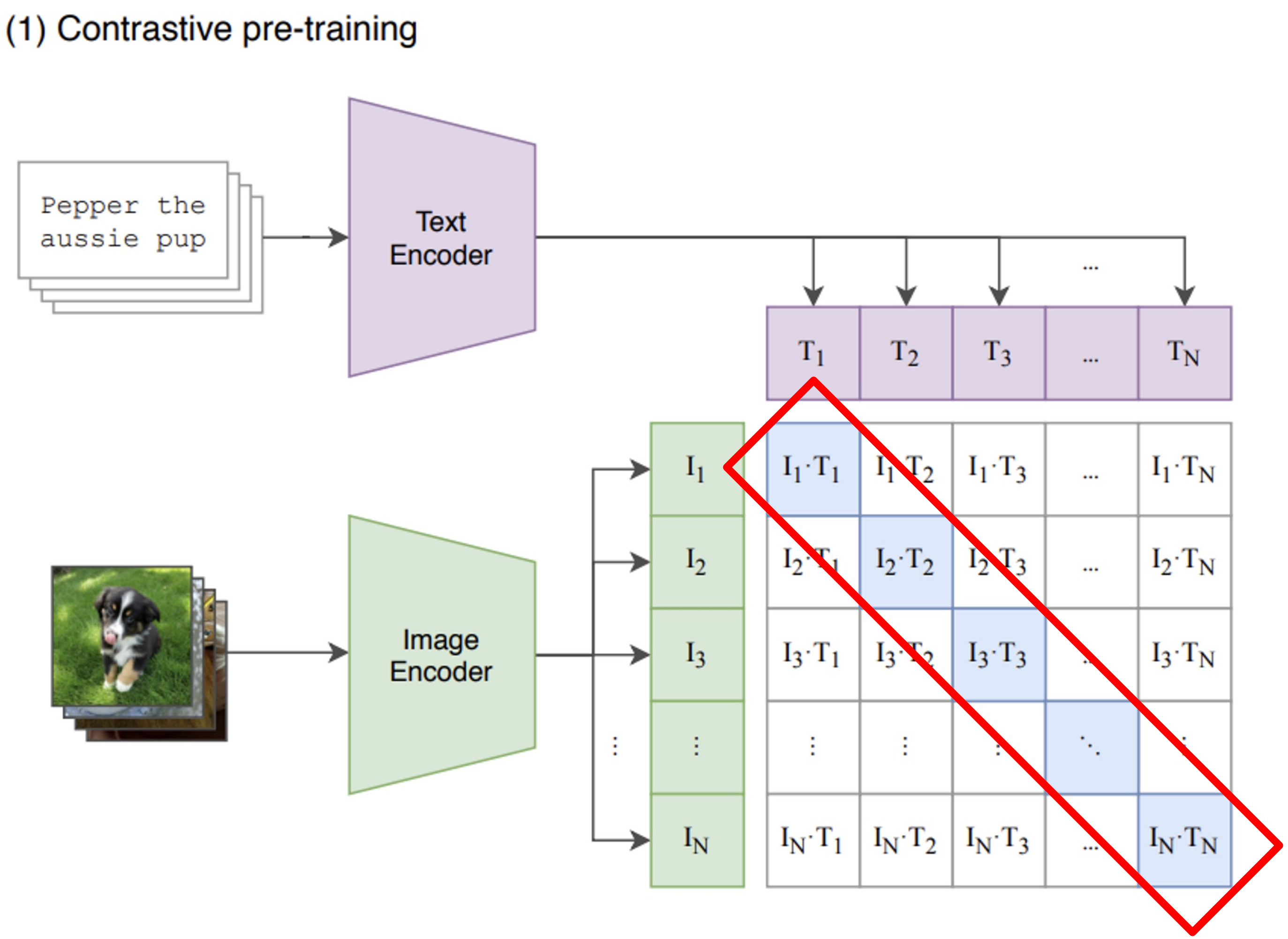
Contrastive pre-training is where a model learns to distinguish between similar and dissimilar pairs of data points during its training phase.

So.. what is "contrastive pre-training"?



It's called "contrastive" because it focuses on contrasting or comparing features within pairs to learn discriminative representations, effectively teaching the model what makes each data point unique or similar to others

How does CLIP learn which image belongs to which caption?

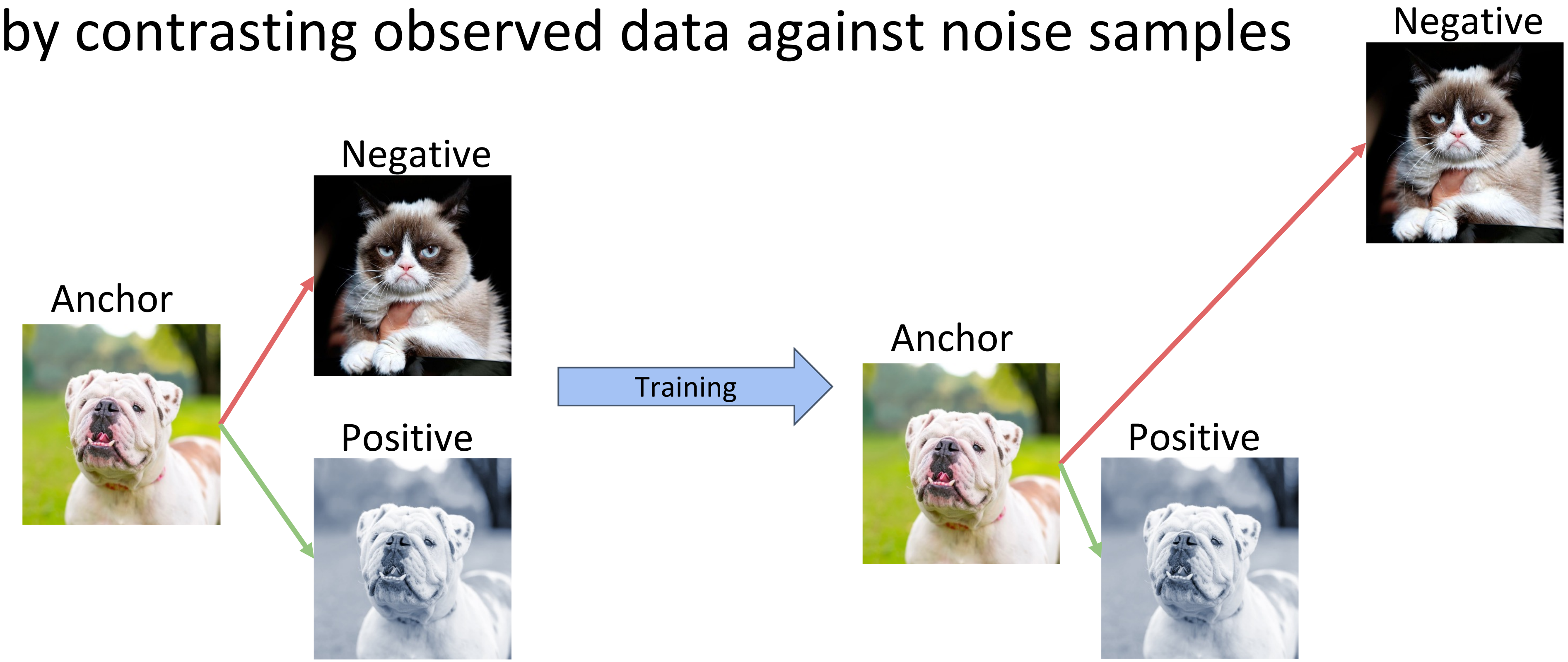


InfoNCE loss:

- maximizes the similarity between correct pairs and minimizes the similarity between incorrect pairs

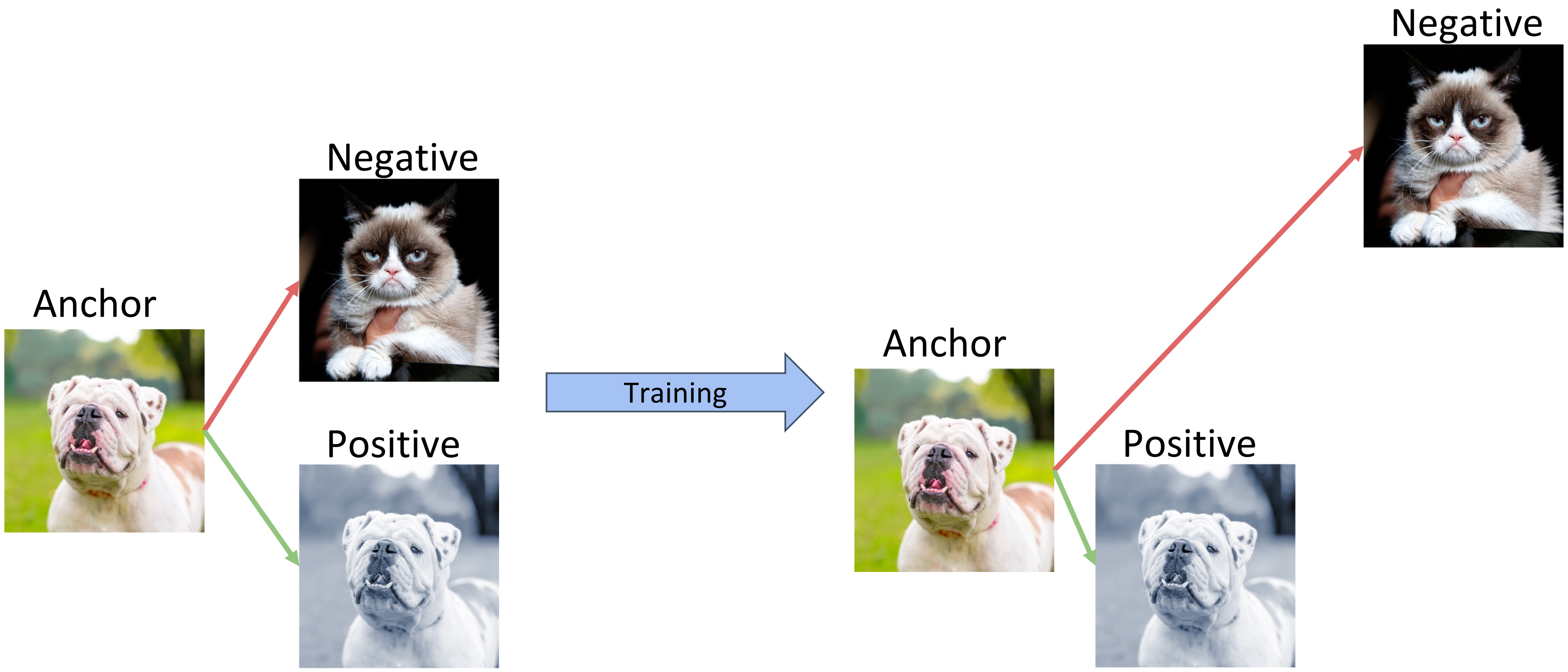
InfoNCE Loss

- “InfoNCE" stands for Information Noise-Contrastive Estimation
- Traditionally, it is a method that estimates mutual information between variables by contrasting observed data against noise samples



InfoNCE Loss

- Goal: pull positive samples closer together and push negative samples farther apart



InfoNCE Loss

$$L_{\text{InfoNCE}} = -\log\left(\frac{\exp(s_{\text{positive}}/\tau)}{\exp(s_{\text{positive}}/\tau) + \sum_{i=1}^{K}\exp(s_{\text{negative}_i}/\tau)}\right)$$

InfoNCE Loss

$$L_{\text{InfoNCE}} = -\log\left(\frac{\exp(s_{\text{positive}}\square)}{\exp(s_{\text{positive}}\square) + \sum_{i=1}^{K}\exp(s_{\text{negative}_i}\square)}\right)$$

For ease of understanding, we can ignore the temperature *tau,* which is scalar that controls the smoothness of the softmax distribution

InfoNCE Loss, similarity calculation

$$L_{\text{InfoNCE}} = -\log\left(\frac{\exp(\boxed{s_{\text{positive}}}\square)}{\exp(\boxed{s_{\text{positive}}}\square) + \sum_{i=1}^{K}\exp(\boxed{s_{\text{negative},}}\square)}\right)$$

$$s(x_i, y_j) = \frac{f(x_i)^T g(y_j)}{\|f(x_i)\|\|g(y_j)\|}$$

Look familiar?

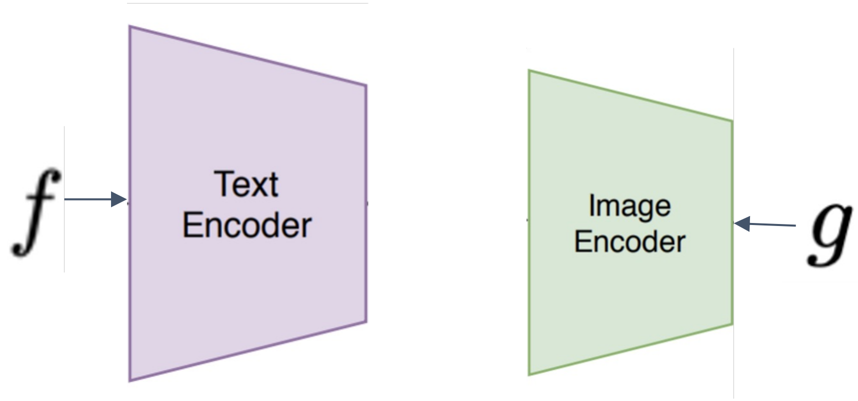
InfoNCE Loss, similarity calculation

$$L_{\text{InfoNCE}} = -\log\left(\frac{\exp(\boxed{s_{\text{positive}}}\square)}{\exp(\boxed{s_{\text{positive}}}\square) + \sum_{i=1}^{K}\exp(\boxed{s_{\text{negative},}}\square)}\right)$$

$$s(x_i, y_j) = \frac{f(x_i)^T g(y_j)}{\|f(x_i)\|\|g(y_j)\|}$$

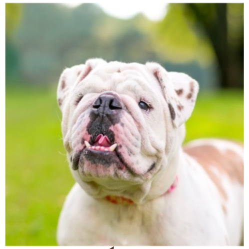
Look familiar? This is just cosine similarity!

InfoNCE Loss, similarity calculation

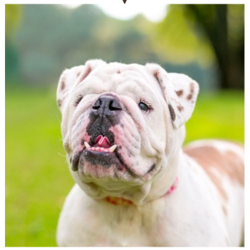


$$s(x_i, y_j) = \frac{f(x_i)^T g(y_j)}{\|f(x_i)\|\|g(y_j)\|}$$

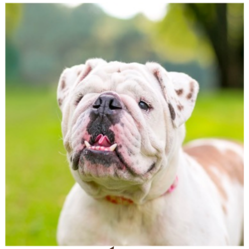
Meaning of “Positive” vs “Negative” pairs



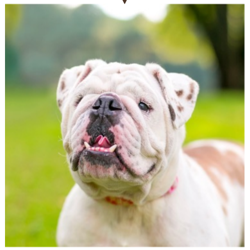
$$L_{\text{InfoNCE}} = -\log\left(\frac{\exp(\boxed{s_{\text{positive}}})}{\exp(\boxed{s_{\text{positive}}}) + \sum_{i=1}^{K}\exp(\boxed{s_{\text{negative}}})}\right)$$

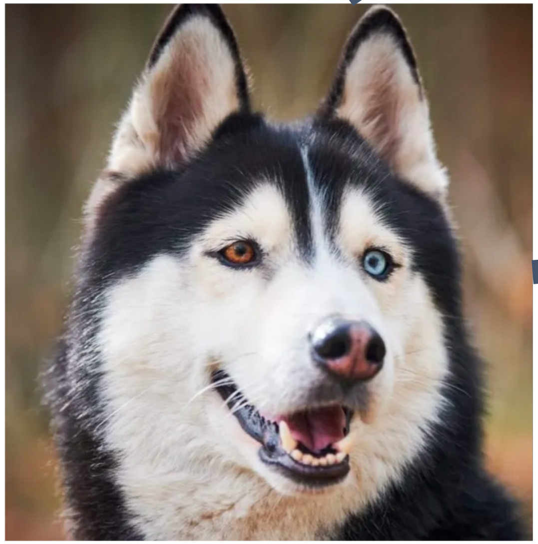


Question: Do you think the picture of a husky would be considered a positive or negative example?



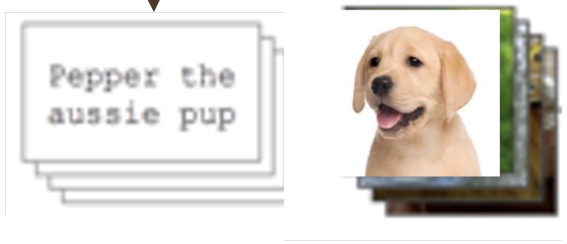
$$L_{\text{InfoNCE}} = \log\left(\frac{\exp(\boxed{s_{\text{positive}}})}{\exp(\boxed{s_{\text{positive}}}) + \sum_{i=1}^{K}\exp(\boxed{s_{\text{negative}_i}})}\right)$$



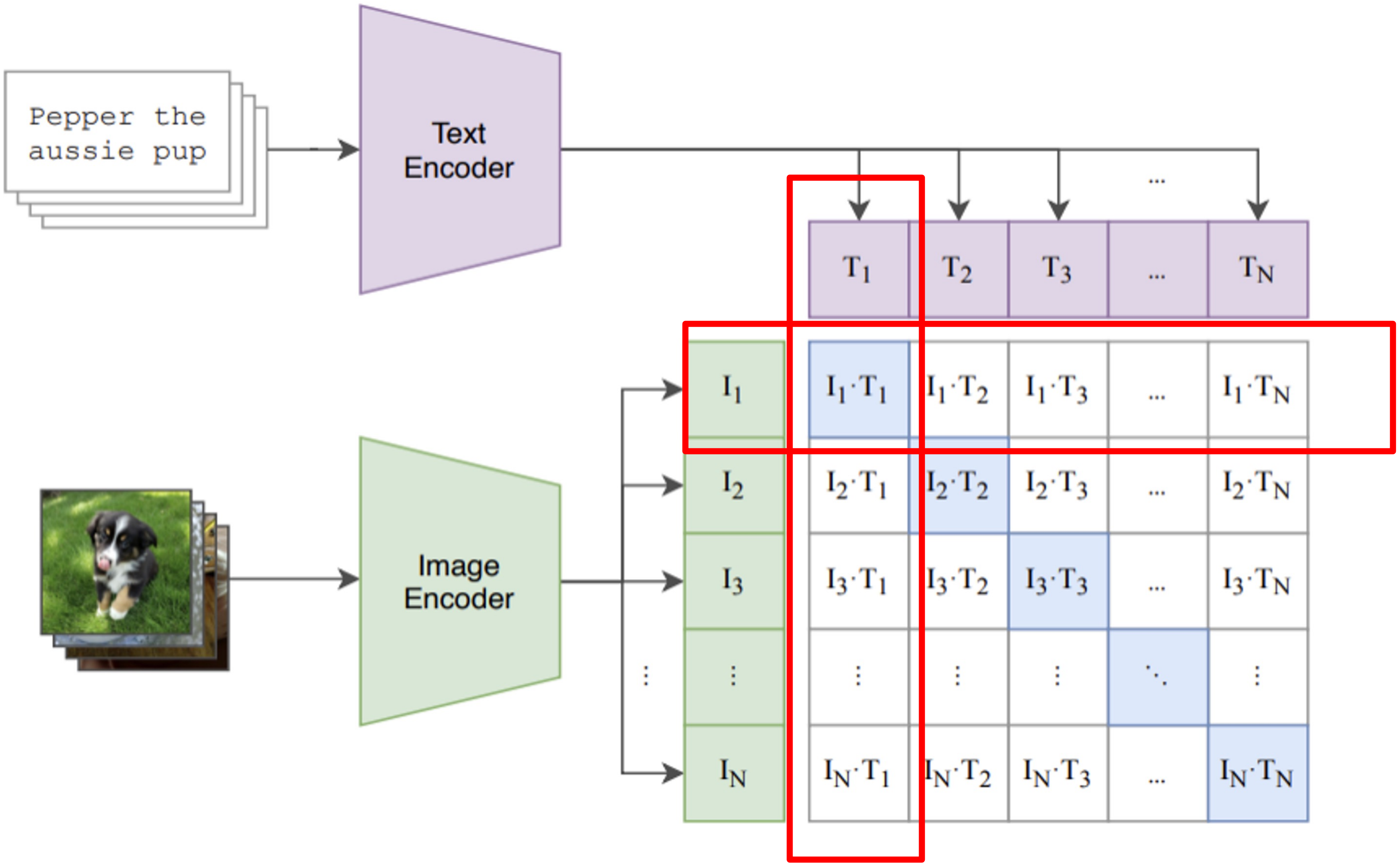


Meaning of “Positive” vs “Negative” pairs, CLIP

$$L_{\text{InfoNCE}} = -\log\left(\frac{\exp\left(\boxed{s_{\text{positive}}}\right)}{\exp\left(\boxed{s_{\text{positive}}}\right) + \sum_{i=1}^{K}\exp\left(\boxed{s_{\text{negative},i}}\right)}\right)$$

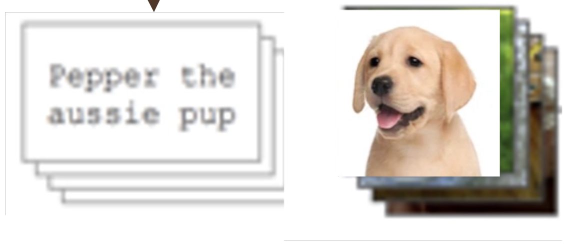


Summation in the denominator comes from…



Meaning of “Positive” vs “Negative” pairs, CLIP

$$L_{\text{InfoNCE}} = -\log\left(\frac{\exp(s_{\text{positive}})}{\exp(s_{\text{positive}}) + \sum_{i=1}^{K}\exp(s_{\text{negative}_i})}\right)$$



Goal: pull positive samples closer together and push negative samples farther apart

Is step (1) really that simple?

```
# 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.

How do we use the learned information? Steps (2) and (3)

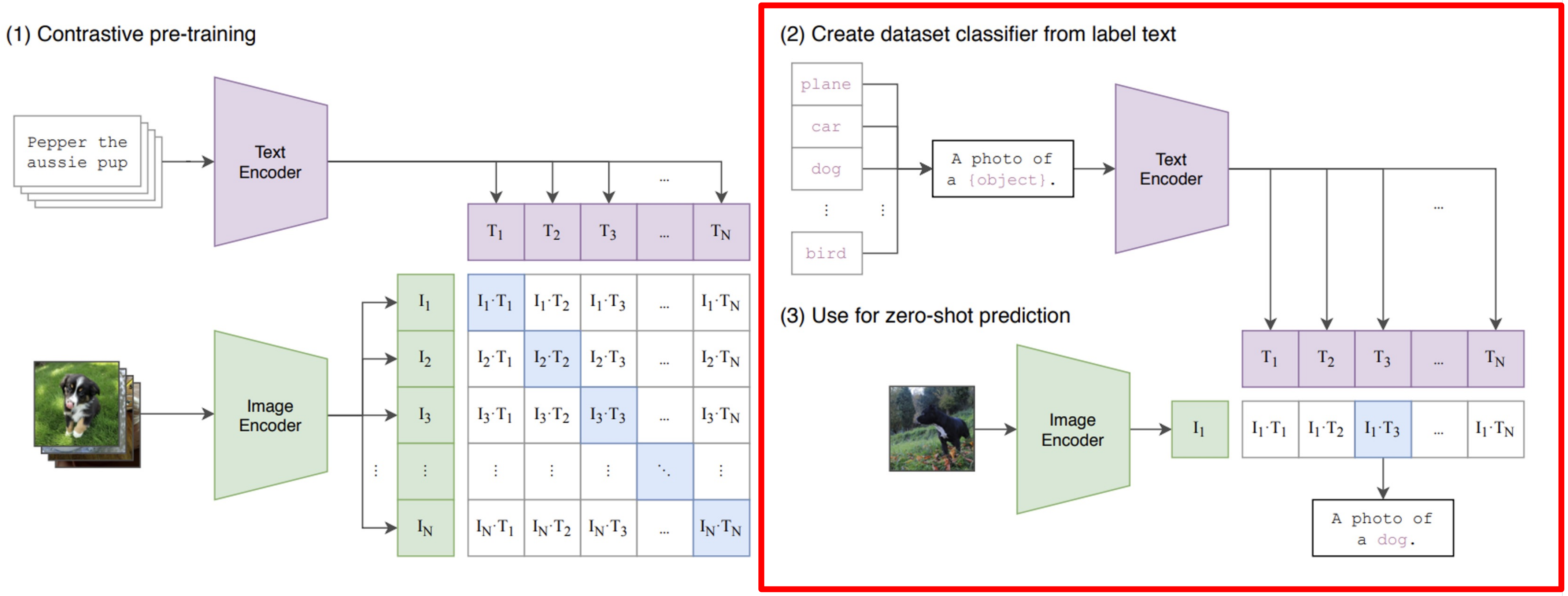
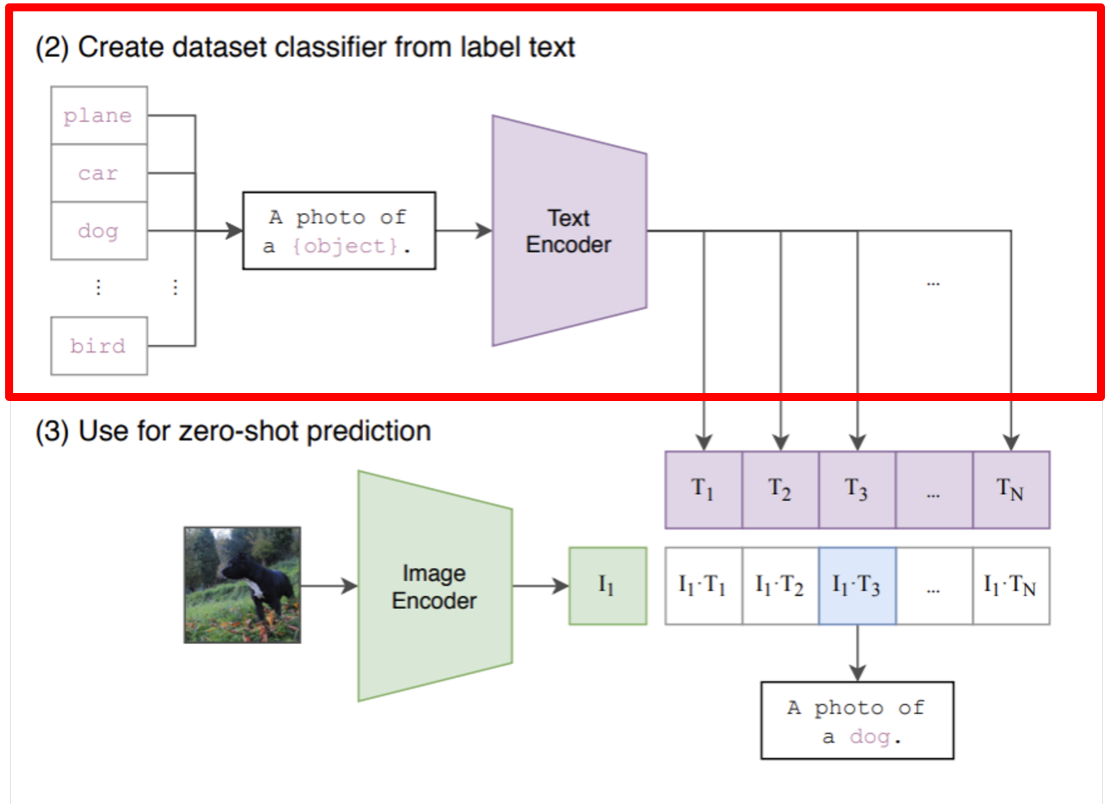


Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

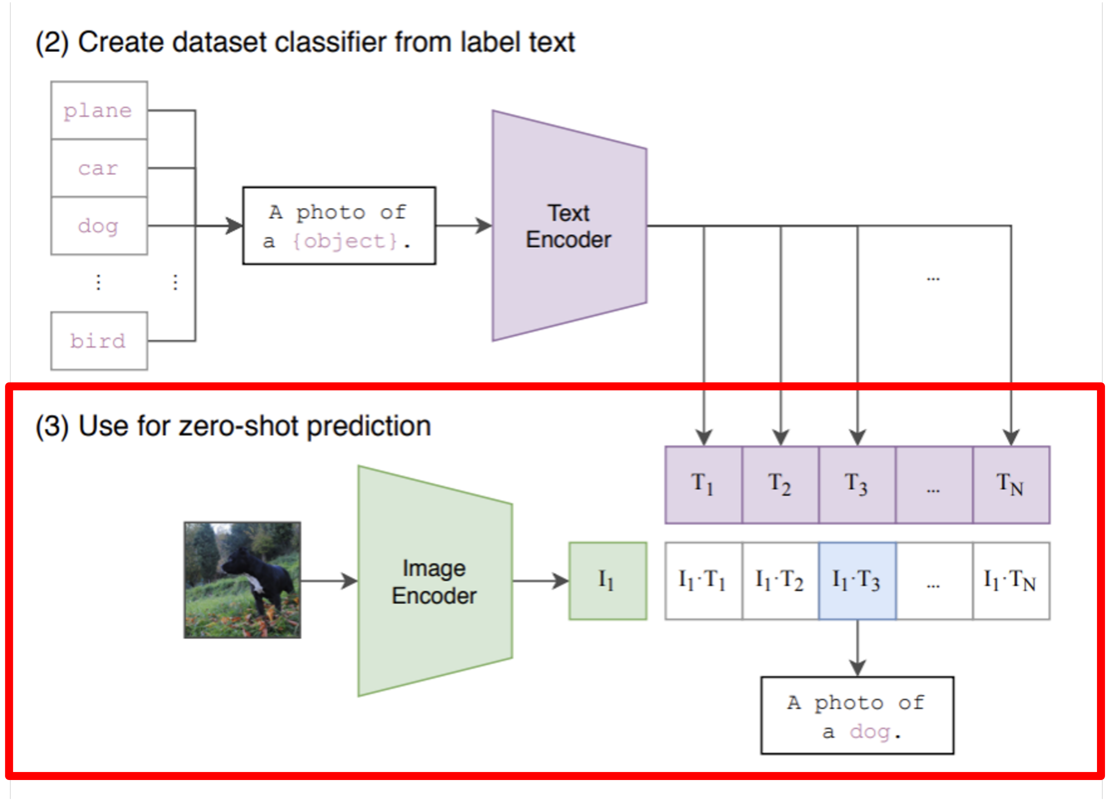
CLIP - Steps (2) and (3)

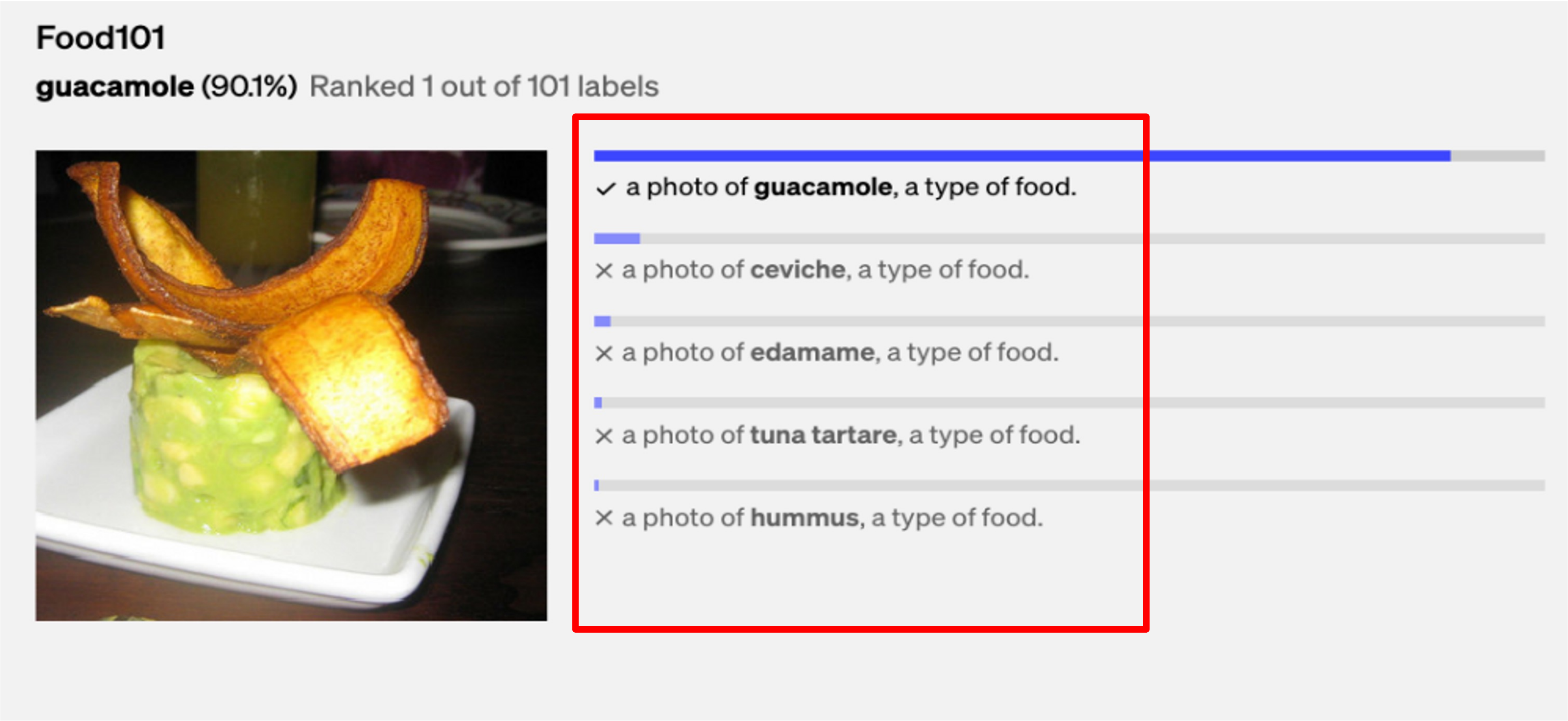
- Classes come from a target dataset (e.g. ImageNet)
- Text prompts are created
- The learned Text Encoder (from Step 1) embeds the prompts



CLIP - Steps (2) and (3)

- Test images are embedded by the Image Encoder (step 1)
- Cosine similarity is calculated between all the prompts and the current image
- The image-caption pair with the highest similarity becomes the “predicted label”

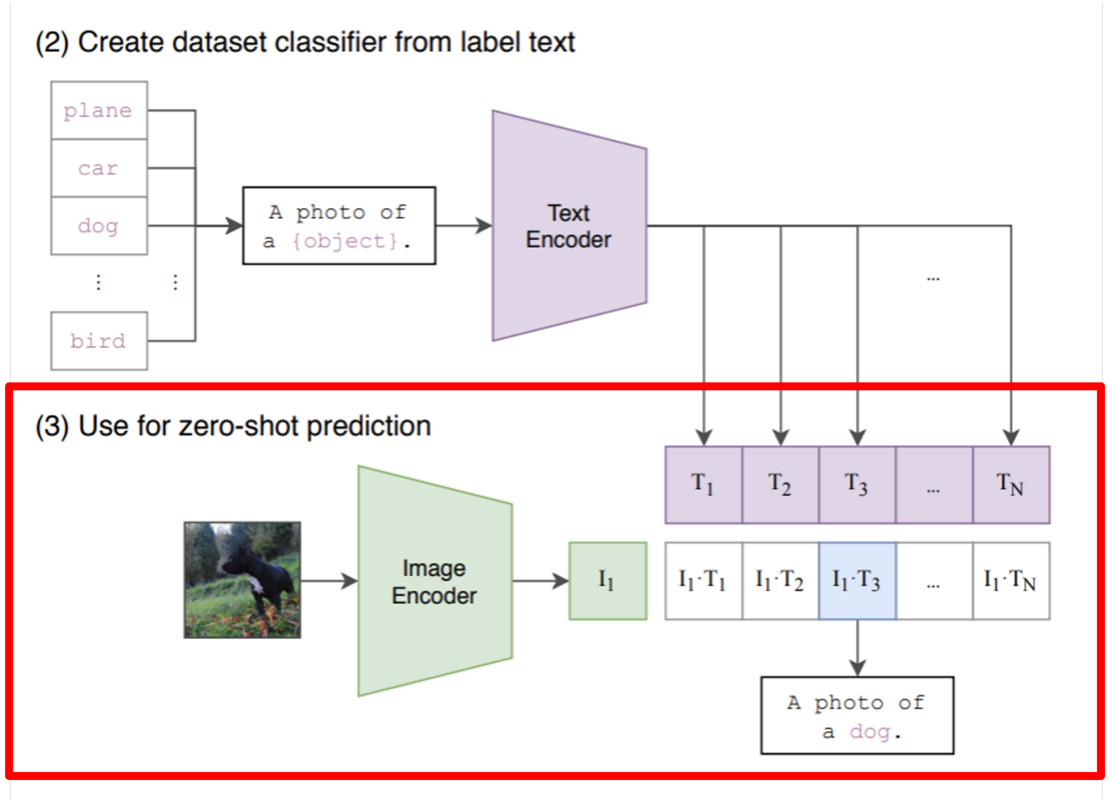




Example of how prompts are used

CLIP - Steps (2) and (3)

- TLDR: Using contrastively trained image and text encoders that understand which image-text pairs “belong” together in the wild, enables the model to make a prediction over which new *unseen* image-text pairs belong together



CLIP - Overview of Training and Model Details

- Existing datasets are... "small"
  - Visual Genome and MSCOCO ~ 100,000
  - "High quality" images from YFCC100M ~ 15M

- CLIP is trained on 400M image-text dataset from a "variety of publicly available sources"

- Very little data augmentation needed - only used a random square crop

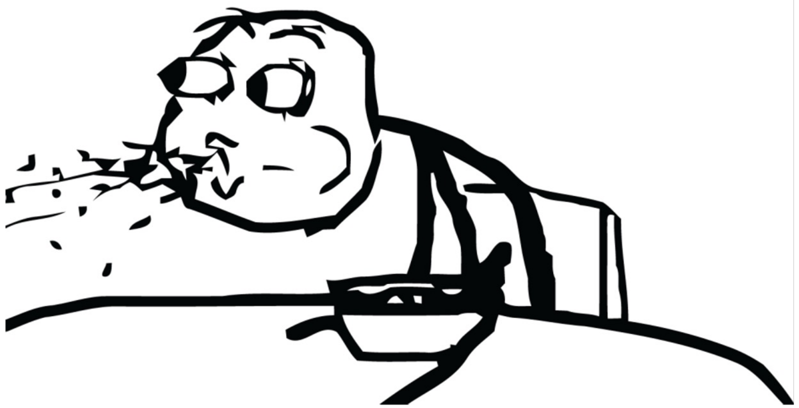
- Temperature parameter (scalar) is learned - not tuned

OpenAI

CLIP - Overview of Training and Model Details

- Text encoder → Transformer
  - a 63M-parameter 12- layer 512-wide model with 8 attention heads

- Image encoder → Modified ResNet or Vision Transformer
  - 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



Experiments and Results

- ResNet (pre-trained on ImageNet) and linearly probed for each dataset
    - i.e. ResNet frozen + fine-tuning linear layer
- Zero-shot CLIP beats ResNet on 16/27
    - **Including ImageNet!!!**
- New SoTA for STL10! (99.3%)
- General trend is that 'specialized' datasets perform worse with CLIP

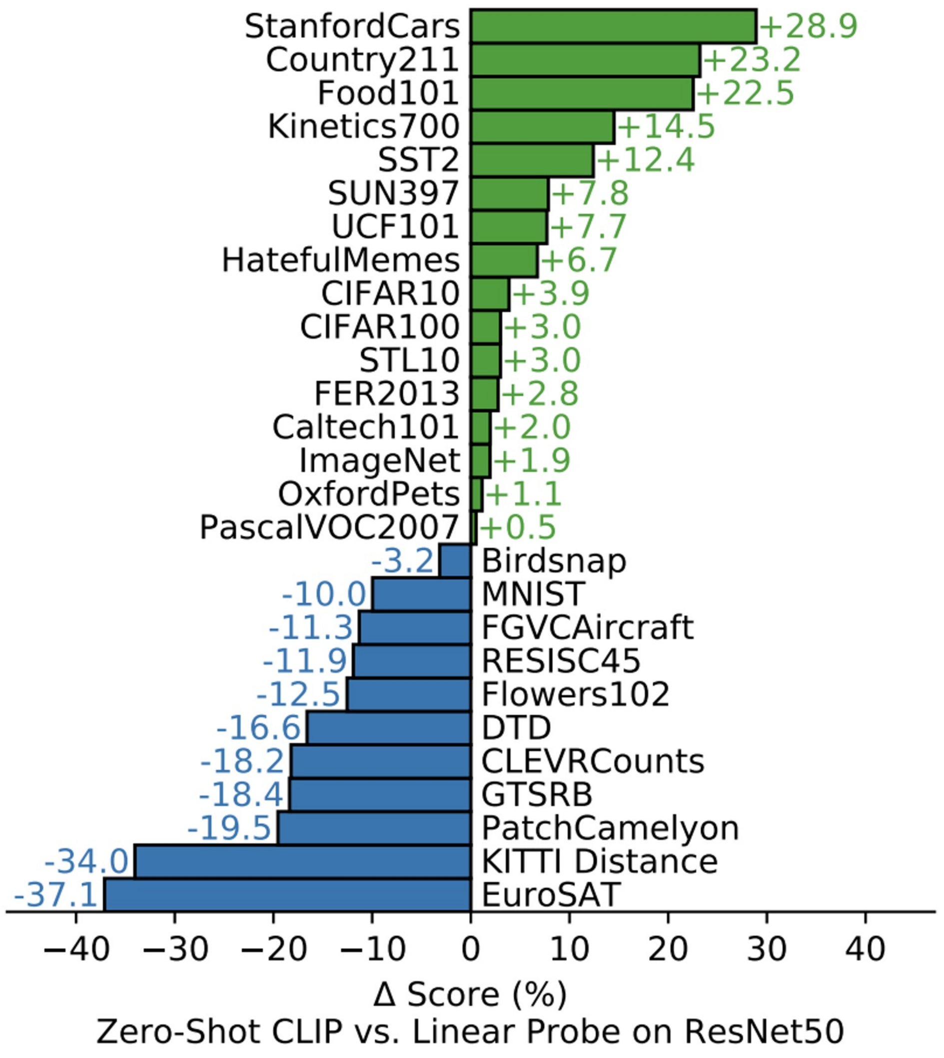


Figure 5. **Zero-shot CLIP is competitive with a fully supervised baseline.** Across a 27 dataset eval suite, a zero-shot CLIP classifier outperforms a fully supervised linear classifier fitted on ResNet-50 features on 16 datasets, including ImageNet.

Experiments and Results

- Why doesn't CLIP do well on simple datasets like MNIST?

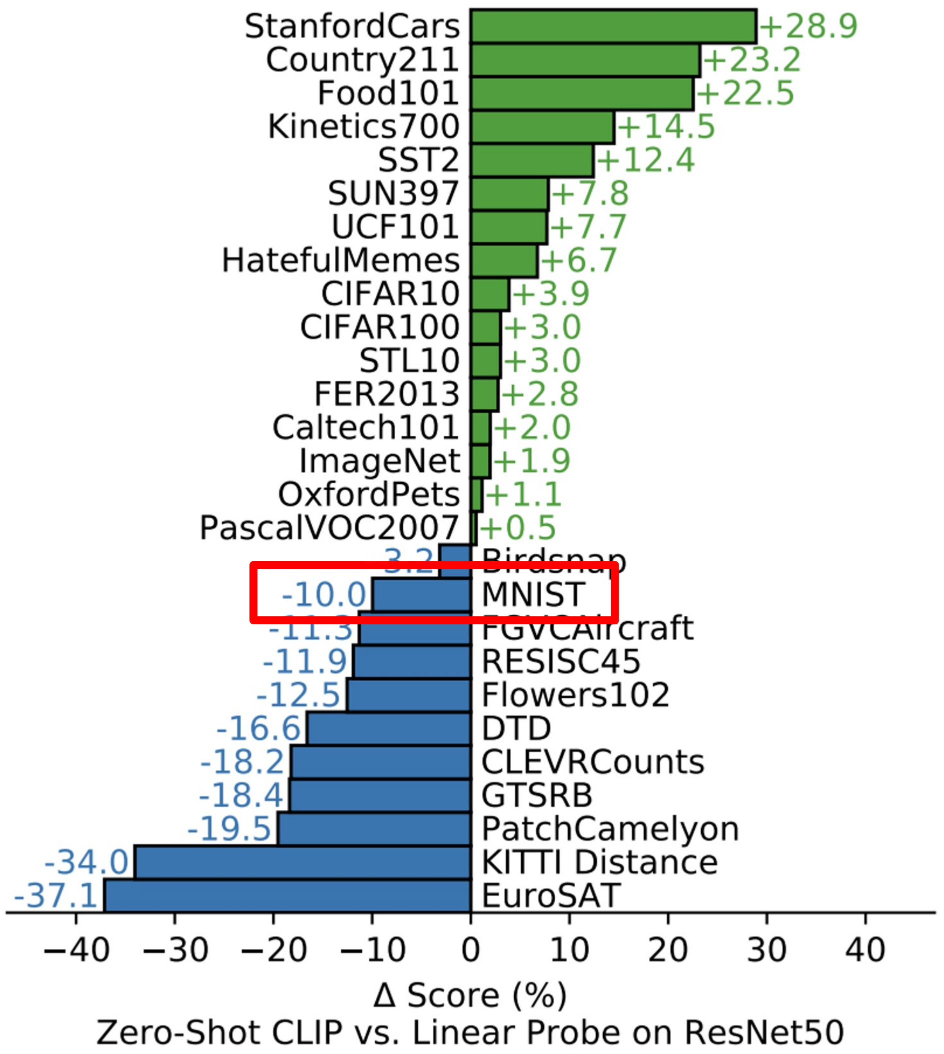
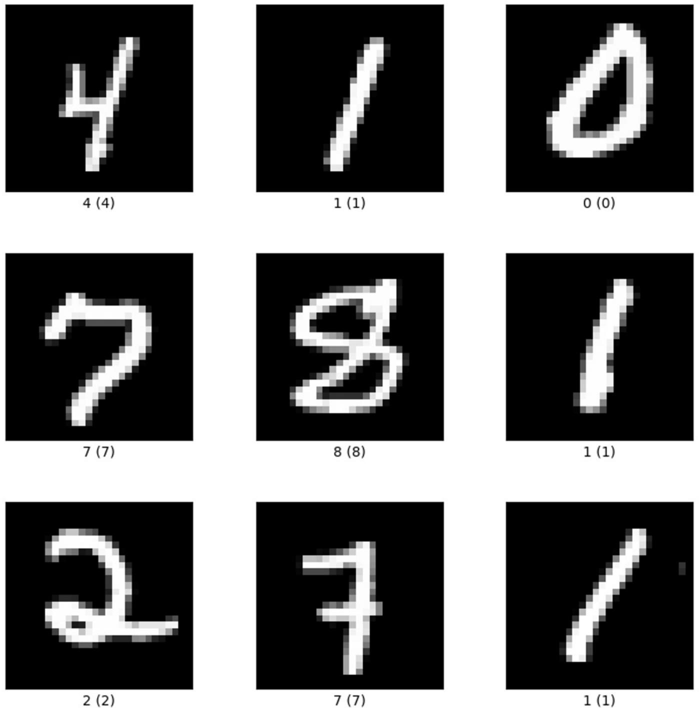


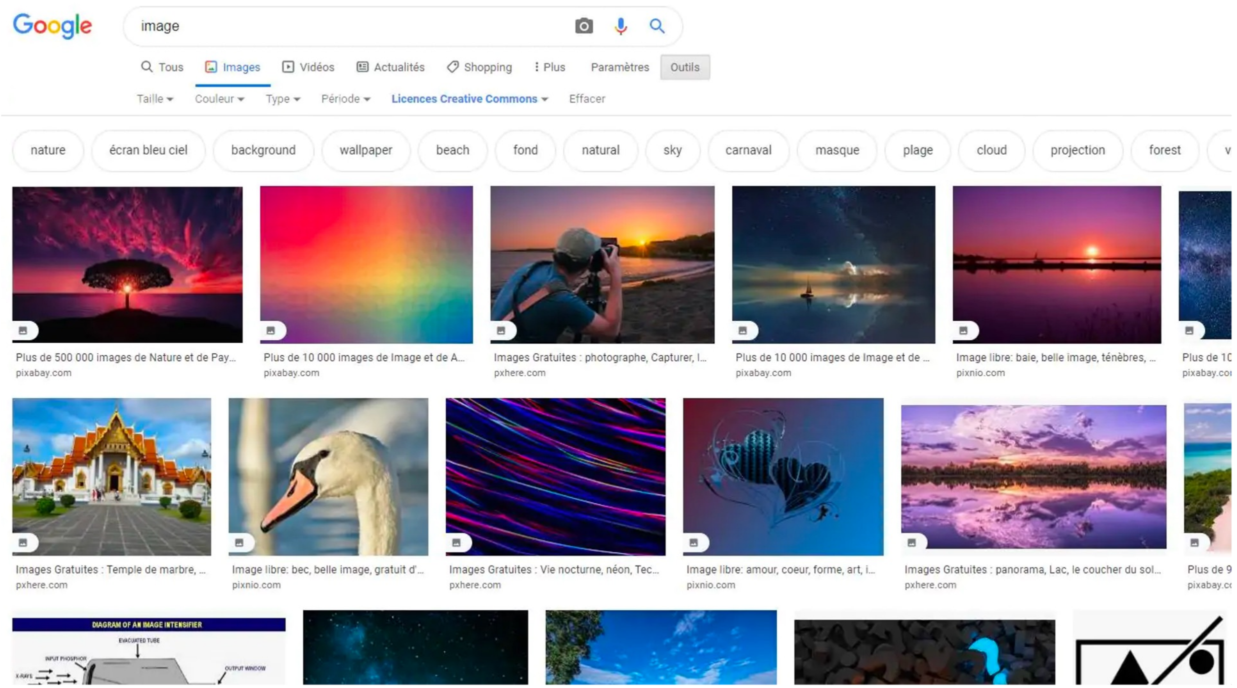
Figure 5. **Zero-shot CLIP is competitive with a fully supervised baseline.** Across a 27 dataset eval suite, a zero-shot CLIP classifier outperforms a fully supervised linear classifier fitted on ResNet-50 features on 16 datasets, including ImageNet.

Experiments and Results

- Why doesn’t CLIP do well on simple datasets like MNIST?

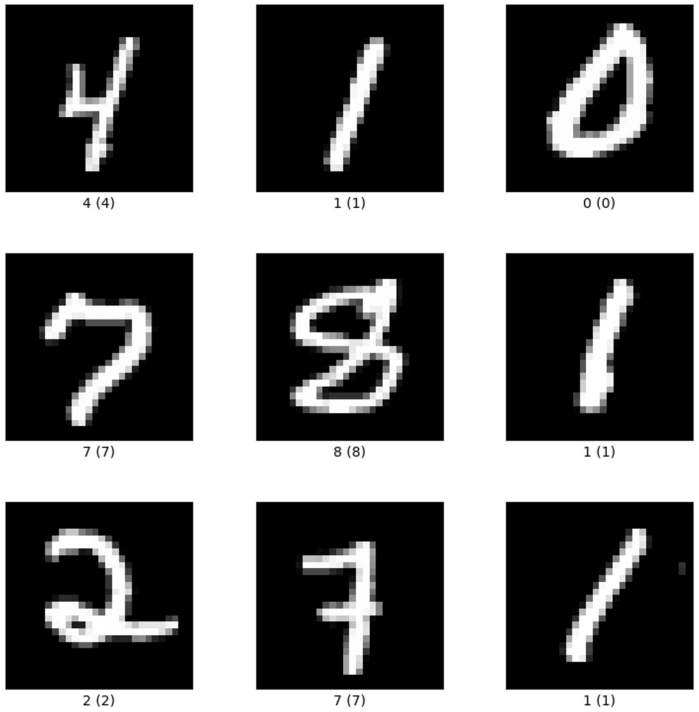


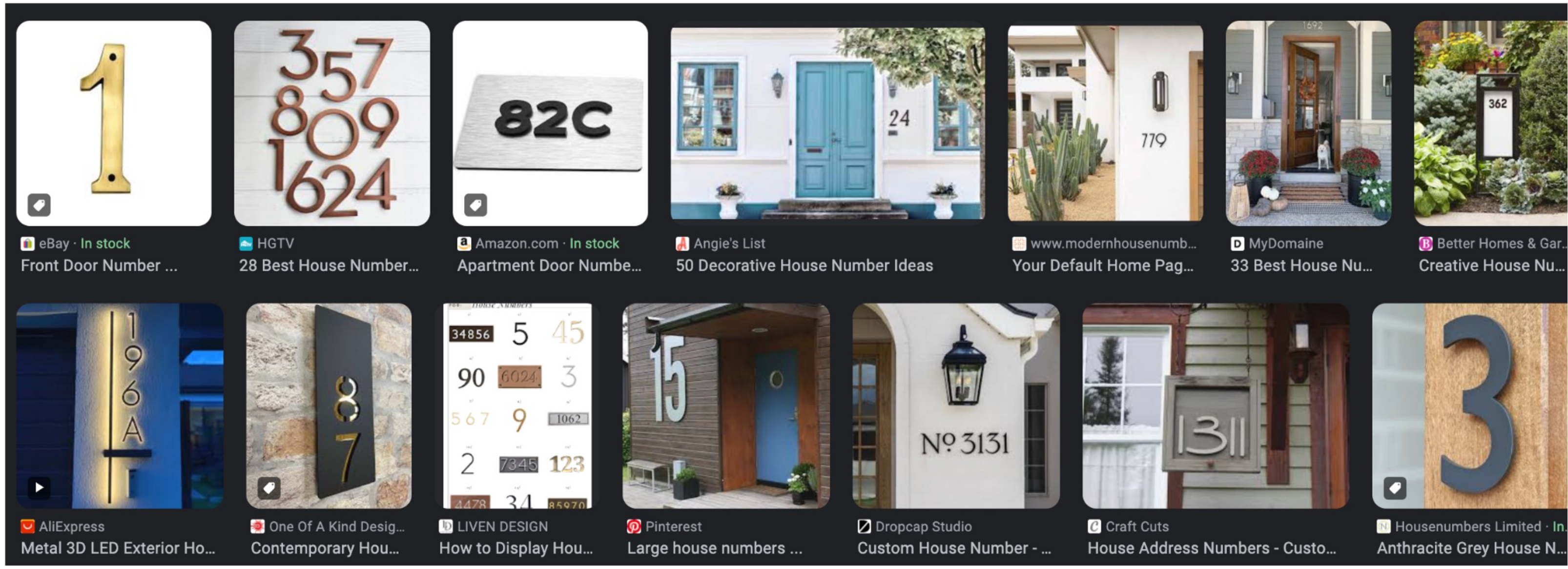
vs



Experiments and Results

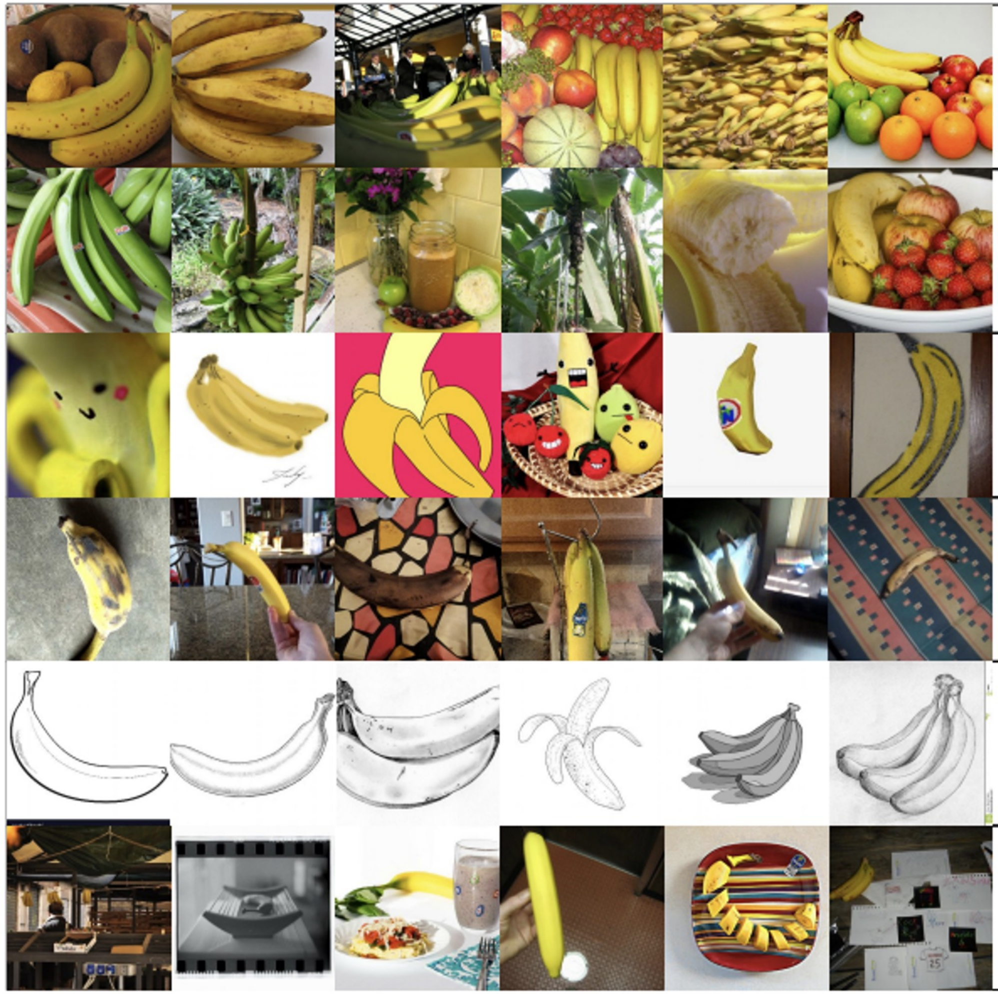
- Why doesn’t CLIP do well on simple datasets like MNIST?





CLIP is a robust vision model

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%

Limitations of CLIP

- Zero-shot CLIP is competitive with ResNet. But ResNet is far from SOTA
  - Authors estimate a 1000x increase in compute is required for zero-shot CLIP to match SOTA

- Poor generalization to (true) out of domain tasks (e.g. MNIST)
  - CLIP does little to address brittle generalization of DL models - rather attempts to circumvent generalization by training on a huge amount of data

- CLIP is **expensive**
  - while it does not require manual data collection and annotation, running a Transformer and a ResNet/ViT on 400 million image-text pairs is a significant effort

- “Web-scale” also means biased

The end!

We learned:

- Contrastive learning can be an effective way to learn image-text representations

- Zero-shot models like CLIP show promise in diverse tasks, limiting the need for more manually annotated datasets

- These models are expensive to train, but the underlying idea can be applied to other modalities and domains!

- Read more about CLIP here :)

Non Zero-shot Performance compared to other models