INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

IIIT Hyderabad

How to train your model?

Makarand Tapaswi

CS7.505 Spring 2024

10th April 2024

Standard one-sample losses

- Regression losses?
  - MSE
  - L1
- Classification losses?
  - Cross-entropy
  - Focal loss

Mean Squared Error Loss

$$L(x, y) = \| \, y \, - f(x; \theta) \, \|^2$$

- Relates input x through some model f with parameters θ to output y
- Loss sums over multiple samples

Cross-entropy Loss

$$L(x, y) = -y \log p(x; \theta)$$

- Relates input x through some model p with parameters θ to output y
- Loss sums over multiple samples

Contrastive Learning

- Relate multiple inputs x with each other (in the limit, at least 2)
- Think about representations instead of outputs
- What kind of losses can come here?

Reference material

- Very good blog posts
- https://lilianweng.github.io/posts/2021-05-31-contrastive/
- https://gombru.github.io/2019/04/03/ranking_loss/

Contrastive loss

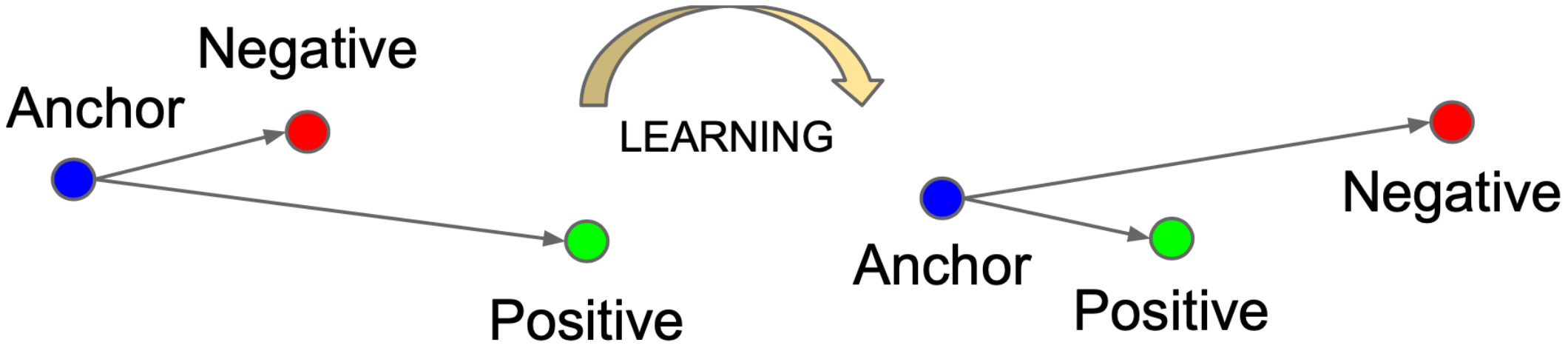
Contrastive loss (Chopra et al. 2005) is one of the earliest training objectives used for deep metric learning in a contrastive fashion.

Given a list of input samples $\{\mathbf{x}_i\}$, each has a corresponding label $y_i \in \{1, \dots, L\}$ among L classes. We would like to learn a function $f_\theta(.) : \mathcal{X} \to \mathbb{R}^d$ that encodes x_i into an embedding vector such that examples from the same class have similar embeddings and samples from different classes have very different ones. Thus, contrastive loss takes a pair of inputs (x_i, x_j) and minimizes the embedding distance when they are from the same class but maximizes the distance otherwise.

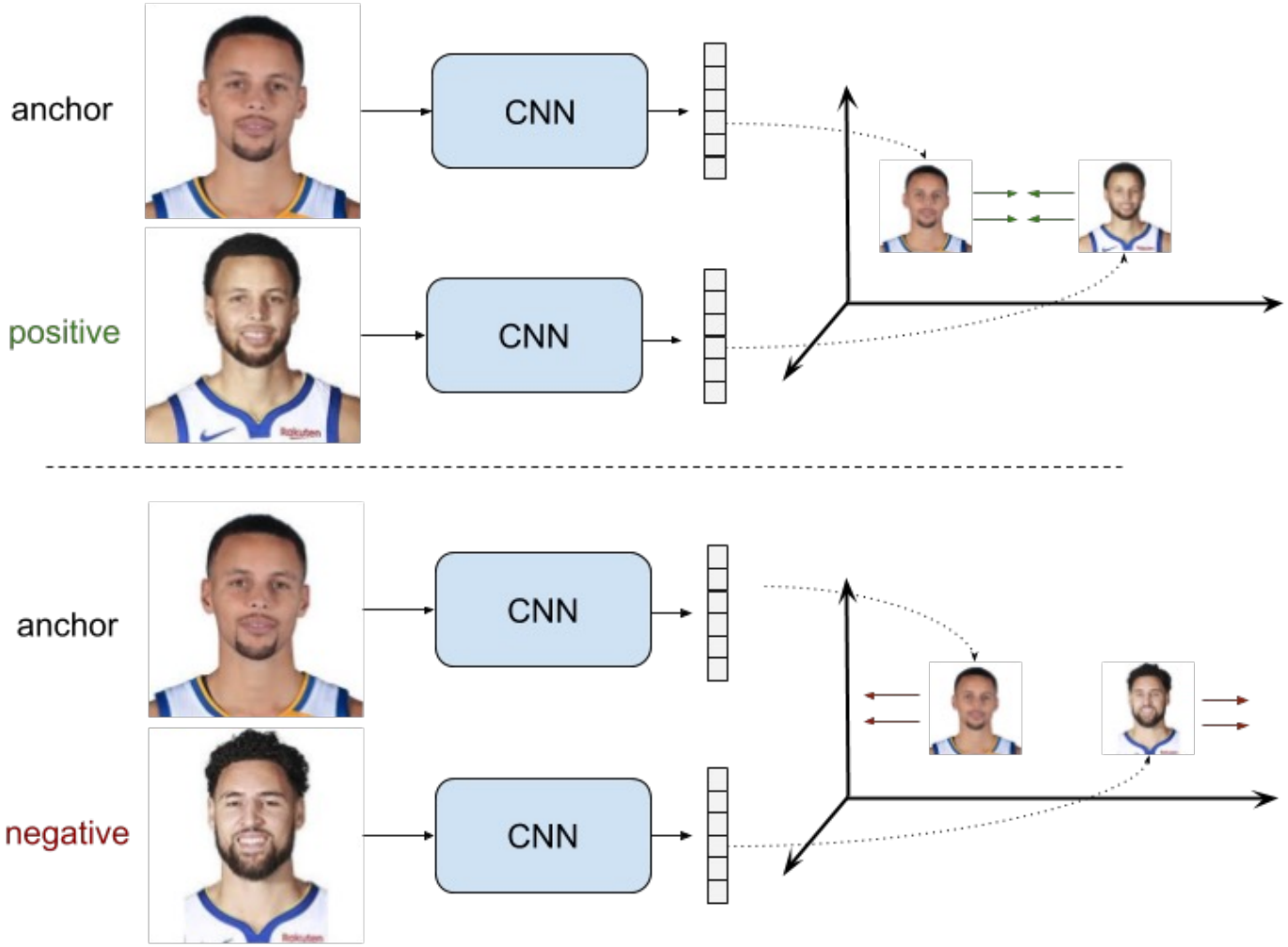
$$\mathcal{L}_{\text{cont}}(\mathbf{x}_i, \mathbf{x}_j, \theta) = \mathbb{1}[y_i = y_j] \|f_\theta(\mathbf{x}_i) - f_\theta(\mathbf{x}_j)\|_2^2 + \mathbb{1}[y_i \neq y_j] \max(0, \epsilon - \|f_\theta(\mathbf{x}_i) - f_\theta(\mathbf{x}_j)\|_2)^2$$

where ϵ is a hyperparameter, defining the lower bound distance between samples of different classes.

Triplet loss



Margin loss



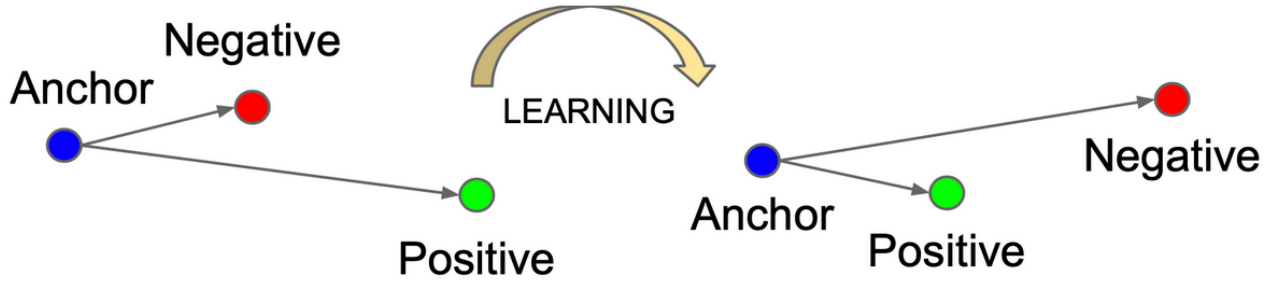


Fig. 1. Illustration of triplet loss given one positive and one negative per anchor. (Image source: Schroff et al. 2015)

Given one anchor input $\mathbf{x}$, we select one positive sample $\mathbf{x}^+$ and one negative $\mathbf{x}^-$, meaning that $\mathbf{x}^+$ and $\mathbf{x}$ belong to the same class and $\mathbf{x}^-$ is sampled from another different class. Triplet loss learns to minimize the distance between the anchor $\mathbf{x}$ and positive $\mathbf{x}^+$ and maximize the distance between the anchor $\mathbf{x}$ and negative $\mathbf{x}^-$ at the same time with the following equation:

$$\mathcal{L}_{\text{triplet}}(\mathbf{x}, \mathbf{x}^+, \mathbf{x}^-) = \sum_{\mathbf{x} \in \mathcal{X}} \max\left(0, \|f(\mathbf{x}) - f(\mathbf{x}^+)\|_2^2 - \|f(\mathbf{x}) - f(\mathbf{x}^-)\|_2^2 + \epsilon\right)$$

where the margin parameter ϵ is configured as the minimum offset between distances of similar vs dissimilar pairs.

It is crucial to select challenging $\mathbf{x}^-$ to truly improve the model.

N-pair loss

Multi-Class N-pair loss (Sohn 2016) generalizes triplet loss to include comparison with multiple negative samples.

Given a $(N+1)$-tuplet of training samples, $\{\mathbf{x}, \mathbf{x}^+, \mathbf{x}_1^-, \dots, \mathbf{x}_{N-1}^-\}$, including one positive and $N-1$ negative ones, N-pair loss is defined as:

$$\begin{aligned}\mathcal{L}_{\text{N-pair}}(\mathbf{x}, \mathbf{x}^+, \{\mathbf{x}_i^-\}_{i=1}^{N-1}) &= \log\left(1 + \sum_{i=1}^{N-1} \exp(f(\mathbf{x})^\top f(\mathbf{x}_i^-) - f(\mathbf{x})^\top f(\mathbf{x}^+))\right) \\ &= -\log \frac{\exp(f(\mathbf{x})^\top f(\mathbf{x}^+))}{\exp(f(\mathbf{x})^\top f(\mathbf{x}^+)) + \sum_{i=1}^{N-1} \exp(f(\mathbf{x})^\top f(\mathbf{x}_i^-))}\end{aligned}$$

If we only sample one negative sample per class, it is equivalent to the softmax loss for multi-class classification.

N-pair vs. Cross-entropy loss

- Derive the similarity between them

Questions?

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

IIIT Hyderabad

Self-supervised Learning (SSL)

Makarand Tapaswi

CS7.505 Spring 2024

10th April 2024

Learning without labels

- Supervision is derived from within the data: “self” supervised
- Use the structure in the data as labels

Examples of structure in language

- Grammar
  - Fill-in-the-blanks
  - Next token prediction
- …

Examples of Structure in images

- Colorization
- Order (jigsaw)
- Neighborhood proximity
- …

Noise Contrastive Estimation (NCE) loss

- Negative samples of N-pair loss treated as “noise”
- “InfoNCE” used in Contrastive Predictive Coding

The **InfoNCE loss** in CPC (Contrastive Predictive Coding; van den Oord, et al. 2018), inspired by NCE, uses categorical cross-entropy loss to identify the positive sample amongst a set of unrelated noise samples.

Given a context vector $\mathbf{c}$, the positive sample should be drawn from the conditional distribution $p(\mathbf{x}|\mathbf{c})$, while $N-1$ negative samples are drawn from the proposal distribution $p(\mathbf{x})$, independent from the context $\mathbf{c}$. For brevity, let us label all the samples as $X = \{\mathbf{x}_i\}_{i=1}^N$ among which only one of them $\mathbf{x}_\mathbf{pos}$ is a positive sample. The probability of we detecting the positive sample correctly is:

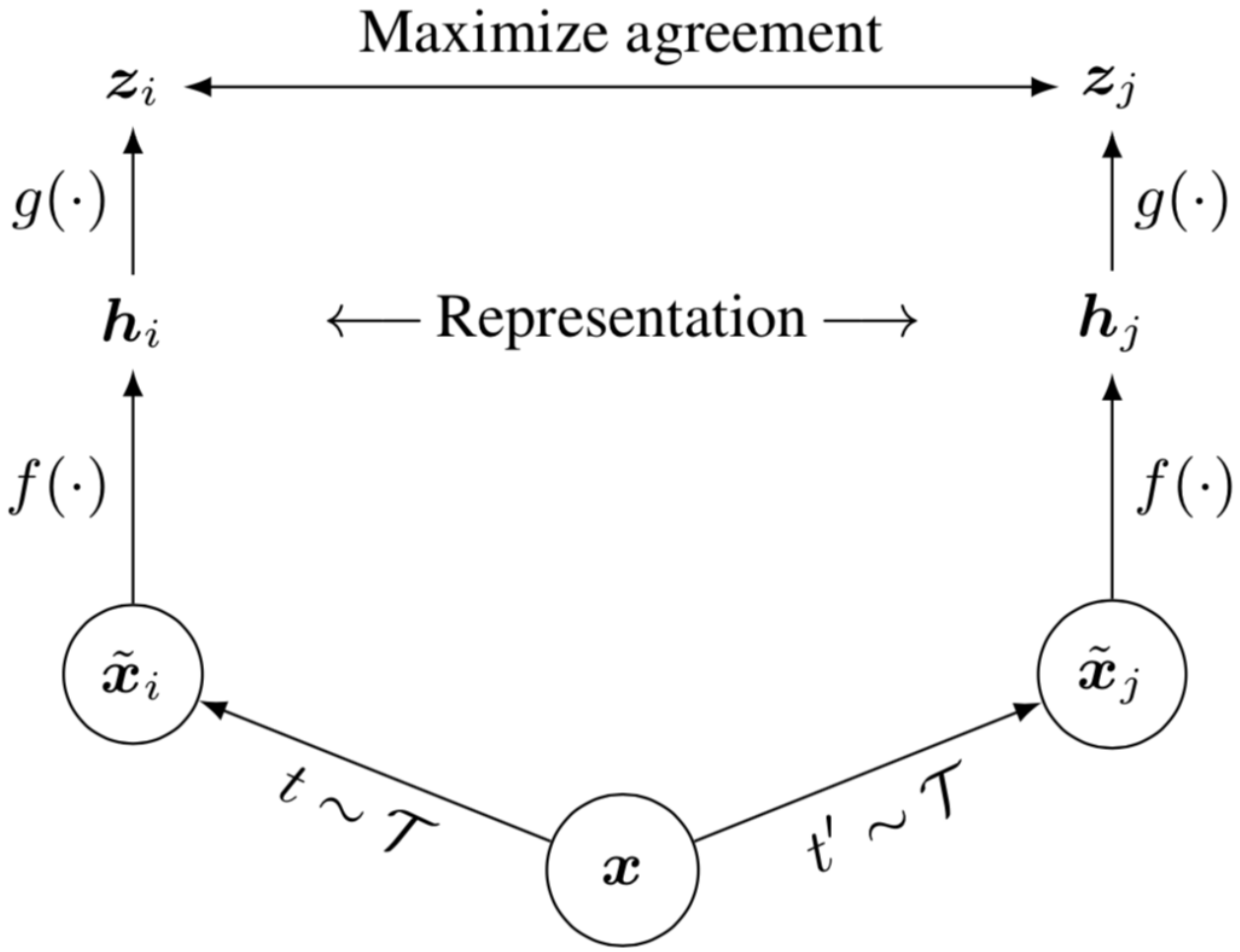
$$p(C = \mathbf{pos}|X, \mathbf{c}) = \frac{p(x_\mathbf{pos}|\mathbf{c}) \prod_{i=1,\dots,N; i\neq\mathbf{pos}} p(\mathbf{x}_i)}{\sum_{j=1}^N \left[p(\mathbf{x}_j|\mathbf{c}) \prod_{i=1,\dots,N; i\neq j} p(\mathbf{x}_i) \right]} = \frac{\frac{p(\mathbf{x}_\mathbf{pos}|c)}{p(\mathbf{x}_\mathbf{pos})}}{\sum_{j=1}^N \frac{p(\mathbf{x}_j|\mathbf{c})}{p(\mathbf{x}_j)}} = \frac{f(\mathbf{x}_\mathbf{pos}, \mathbf{c})}{\sum_{j=1}^N f(\mathbf{x}_j, \mathbf{c})}$$

where the scoring function is $f(\mathbf{x}, \mathbf{c}) \propto \frac{p(\mathbf{x}|\mathbf{c})}{p(\mathbf{x})}$.

The InfoNCE loss optimizes the negative log probability of classifying the positive sample correctly:

$$\mathcal{L}_\text{InfoNCE} = -\mathbb{E}\Big[\log \frac{f(\mathbf{x}, \mathbf{c})}{\sum_{\mathbf{x}' \in X} f(\mathbf{x}', \mathbf{c})} \Big]$$

SimCLR



Chen, et al. A Simple Framework for Contrastive Learning of Visual Representations. ICML 2020

1. Randomly sample a minibatch of N samples and each sample is applied with two different data augmentation operations, resulting in $2N$ augmented samples in total.

$$\tilde{\mathbf{x}}_i = t(\mathbf{x}), \quad \tilde{\mathbf{x}}_j = t'(\mathbf{x}), \quad t, t' \sim \mathcal{T}$$

where two separate data augmentation operators, t and t', are sampled from the same family of augmentations $\mathcal{T}$. Data augmentation includes random crop, resize with random flip, color distortions, and Gaussian blur.

2. Given one positive pair, other $2(N-1)$ data points are treated as negative samples. The representation is produced by a base encoder $f(.)$:

$$\mathbf{h}_i = f(\tilde{\mathbf{x}}_i), \quad \mathbf{h}_j = f(\tilde{\mathbf{x}}_j)$$

3. The contrastive learning loss is defined using cosine similarity $\text{sim}(.,.)$. Note that the loss operates on an extra projection layer of the representation $g(.)$ rather than on the representation space directly. But only the representation $\mathbf{h}$ is used for downstream tasks.

$$\mathbf{z}_i = g(\mathbf{h}_i), \quad \mathbf{z}_j = g(\mathbf{h}_j)$$

$$\mathcal{L}_{\text{SimCLR}}^{(i,j)} = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)}$$

where $\mathbb{1}_{[k \neq i]}$ is an indicator function: 1 if $k \neq i$ 0 otherwise.

Questions?

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

IIIT Hyderabad

Contrastive Language-Image Pretraining

Makarand Tapaswi

CS7.505 Spring 2024

10th April 2024

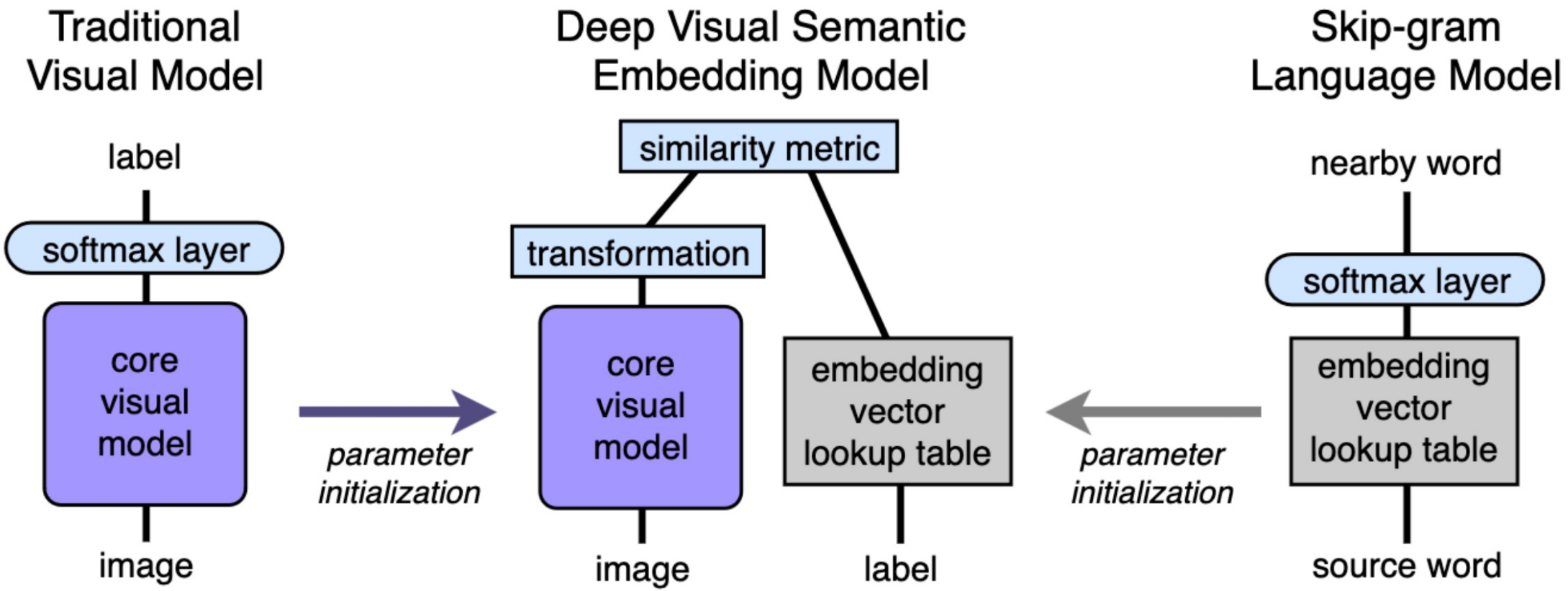
Defining terminology: what is ___

- Pretraining
- Post-pretraining (new! well, relatively)
- Fine-tuning
- Zero-shot learning
- Few-shot learning

Category labels are “language”, but …

- Are hard to scale
  - Very difficult to label objects beyond some 10k! (Full ImageNet has 22k!)
- With limited descriptive potential
  - White shirt vs. Blue shirt
- Are not compositional
  - Laptop **on top of** table
- …

A brief walk into history …

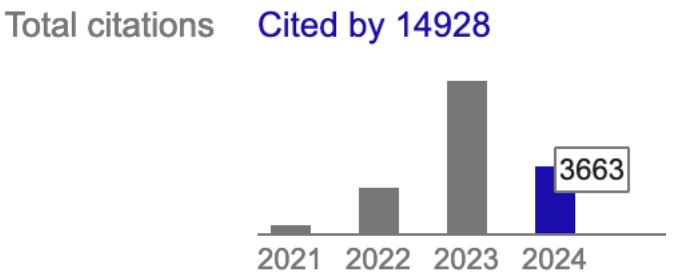


A. Frome, et al. DeViSE: A Deep Visual-Semantic Embedding Model. NeurIPS 2013

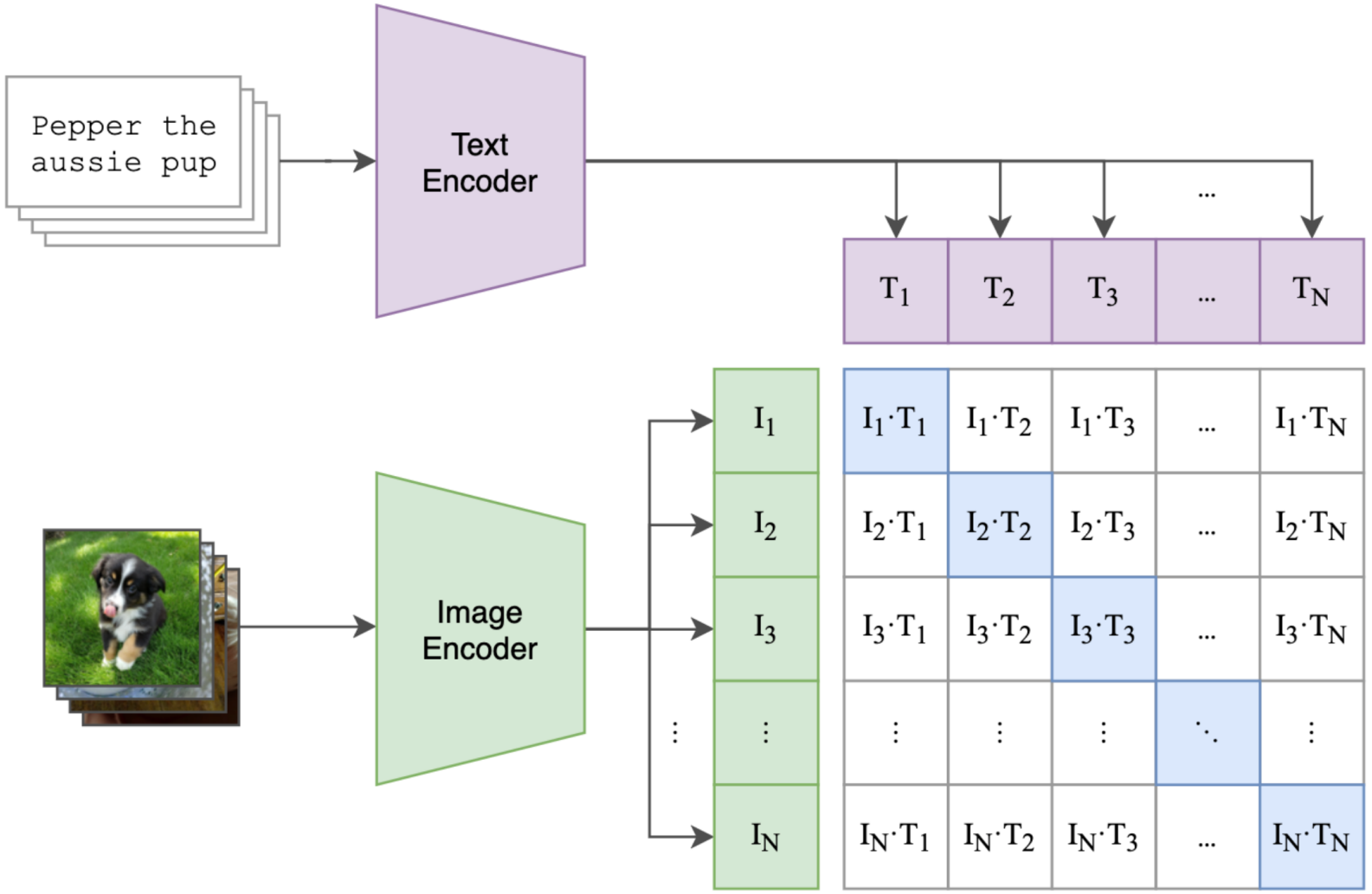
Radford, et al. arXiv: 2103.00020

Learning Transferable Visual Models from Natural Language Supervision

other visual concept. Learning directly from raw text about images is a promising alternative which leverages a much broader source of supervision. We demonstrate that the simple pre-training task of predicting which caption goes with which image is an efficient and scalable way to learn SOTA image representations from scratch on a dataset of 400 million (image, text) pairs collected from the internet. After pre-training, natural language is used to reference learned visual concepts (or describe new ones) enabling zero-shot transfer of the model to downstream tasks. We study



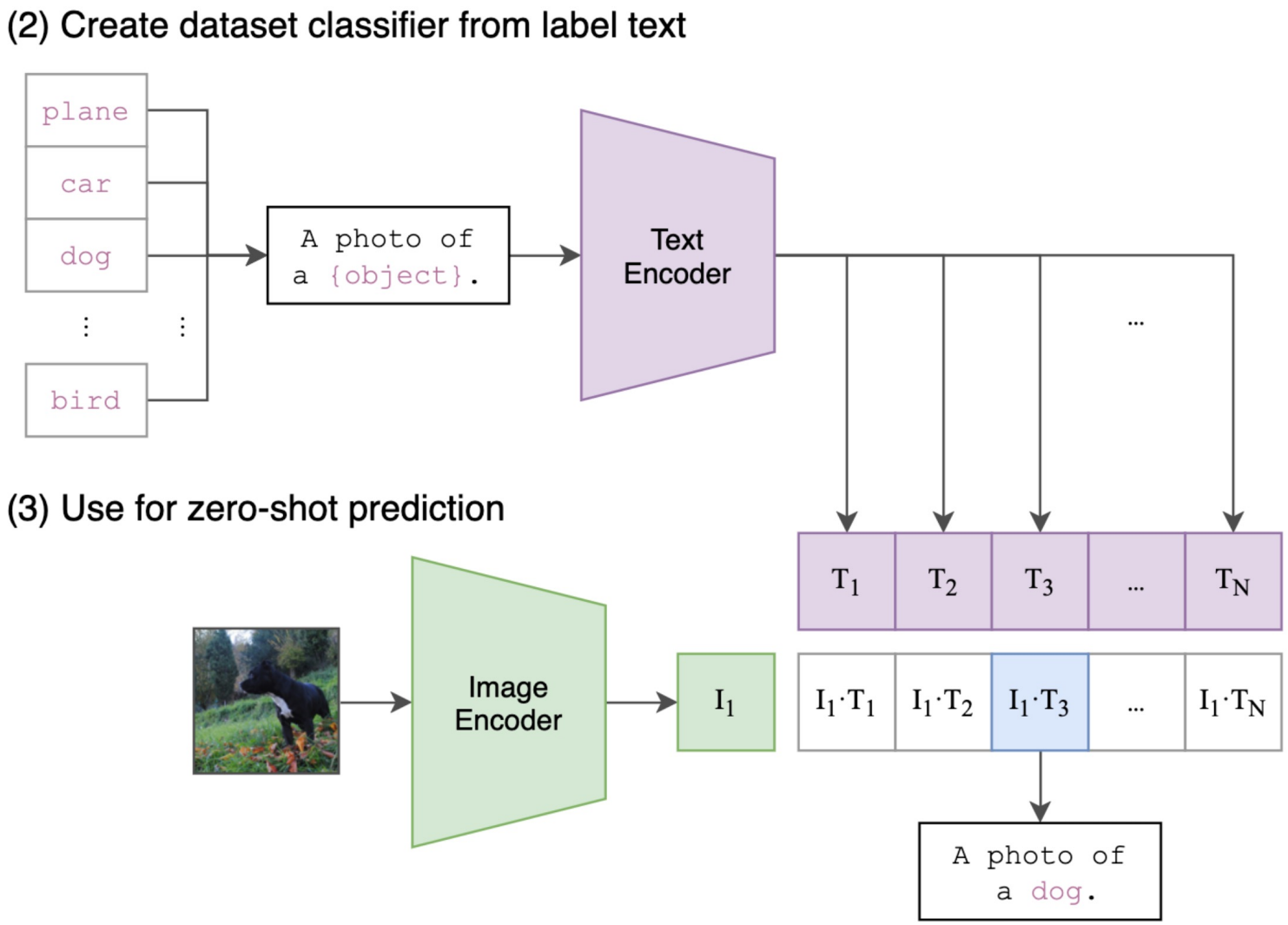
Contrastive Pretraining



Practice math

- Write the equation for the InfoNCE loss between the image and text pairs of CLIP

Zero-shot classification



What's the secret sauce?

A major motivation for natural language supervision is the large quantities of data of this form available publicly on the internet. Since existing datasets do not adequately reflect this possibility, considering results only on them would underestimate the potential of this line of research. To address this, we constructed a new dataset of 400 million (image, text) pairs collected form a variety of publicly available sources on the Internet. To attempt to cover as broad a set of visual concepts as possible, we search for (image, text) pairs as part of the construction process whose text includes one of a set of 500,000 queries. We approximately class balance the results by including up to 20,000 (image, text) pairs per query. The resulting dataset has a similar total word count as the WebText dataset used to train GPT-2. We refer to this dataset as WIT for WebImageText

CLIP pseudocode

```
# 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.

Zero-shot
CLIP is good!

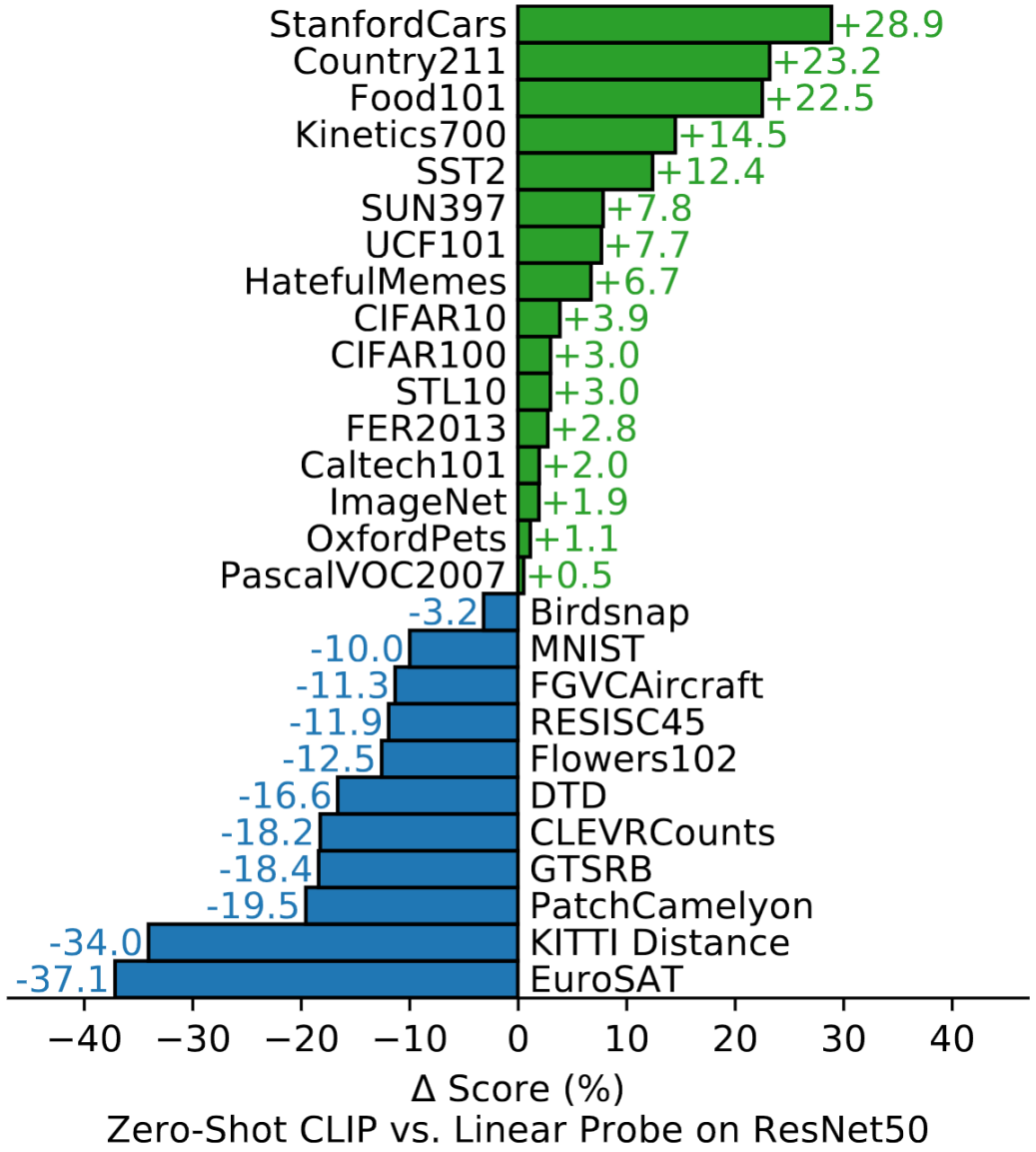
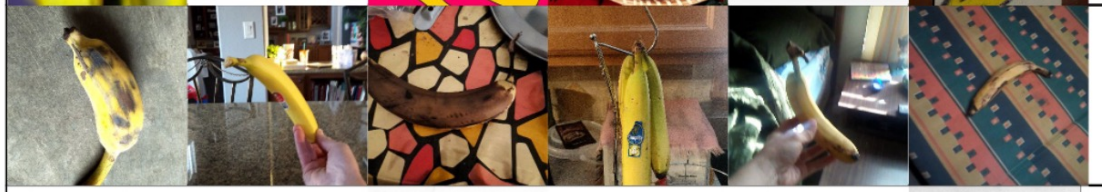
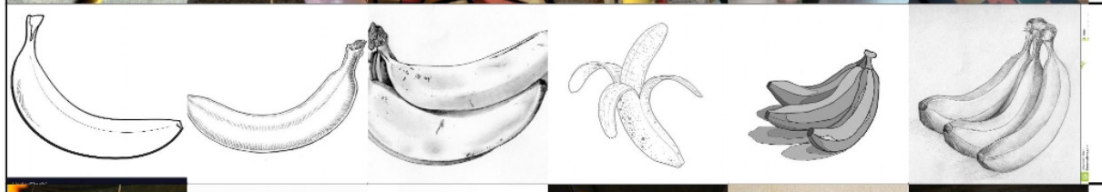
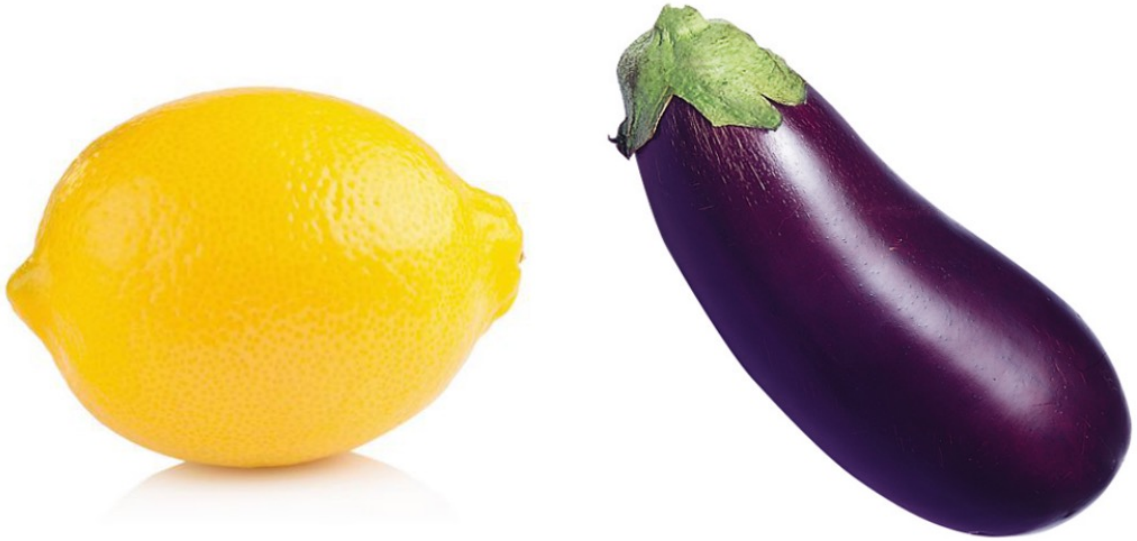


Figure 5. **Zero-shot CLIP is competitive with a fully supervised baseline.** Across a 27 dataset eval suite, a zero-shot CLIP classifier outperforms a fully supervised linear classifier fitted on ResNet-50 features on 16 datasets, including ImageNet.

Robustness

	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%

CLIP Association Bias



CLIP: "In this picture, the color of the lemon is purple."

Figure 1. When we ask CLIP the color of the lemon in the above image, CLIP answers “purple”. The text prompt we use is “In this picture, the color of the lemon is [mask]”, where CLIP picks one from [red, green, yellow, orange, purple].

Yamada, et al. When are Lemons Purple? The Concept Association Bias of CLIP. EMNLP 2023

Winoground failures

(a) some plants surrounding a lightbulb

(b) a lightbulb surrounding some plants

Figure 1. An example from Winoground. The two sentences contain the same words but in a different order. The task of understanding which image and caption match is trivial for humans but much harder for vision and language models. Every model that we tested (UNITER, ViLLA, VinVL, VisualBERT, ViLT, LXMERT, ViLBERT, UniT, FLAVA, CLIP, VSE++, and VSRN) fails to correctly pair the images and captions, except the large checkpoint of ViLLA by a very thin margin (0.00013 confidence).

Thrush, et al. Winoground: Probing Vision and Language Models for Visio-Linguistic Compositionality. CVPR 2022

Why?

- Not aware about fine-grained details
- Is not compositional
- Matching global representations is insufficient
- Needs hard negatives!

Impact

- Among commonly used visual encoder!
- LAION-5B: open-source data alternatives to WIT
- BLIP: inspired by CLIP ideas
- InstructBLIP: instruction tuning (CLIP visual encoder + Llama LLM)
- Stable Diffusion: uses CLIP text encoder

Questions?