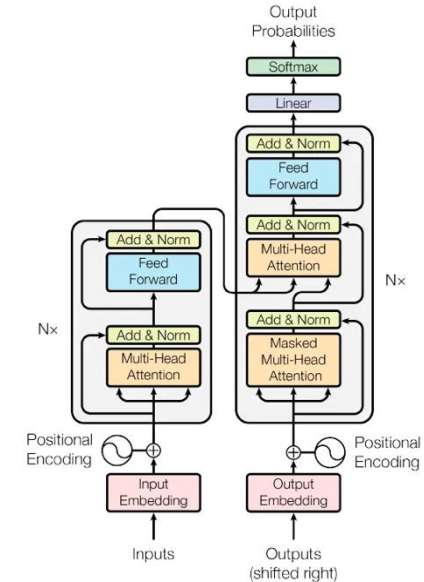


Contrastive Language-Image Pretraining (CLIP)

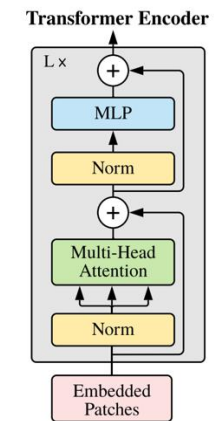
Transformers for Robotics, Lecture 3, Nikolaus Correll

So far...

- Self-attention has replaced recurrent models as it is easier to train (parallel) and numerically stabler (vanishing gradients)
- Transformers use self-attention to process tokens of
 - Text (Lecture 2)
 - Images (Lecture 3)
- Today: mixing images and text



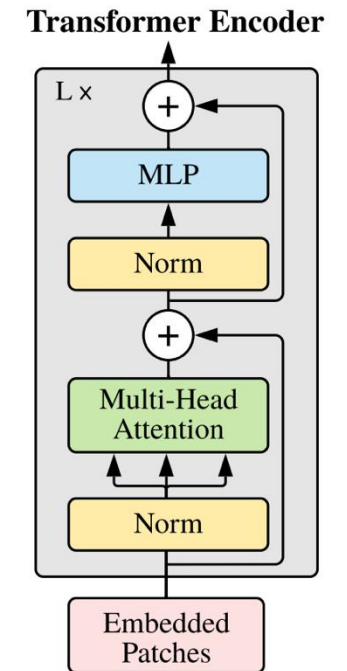
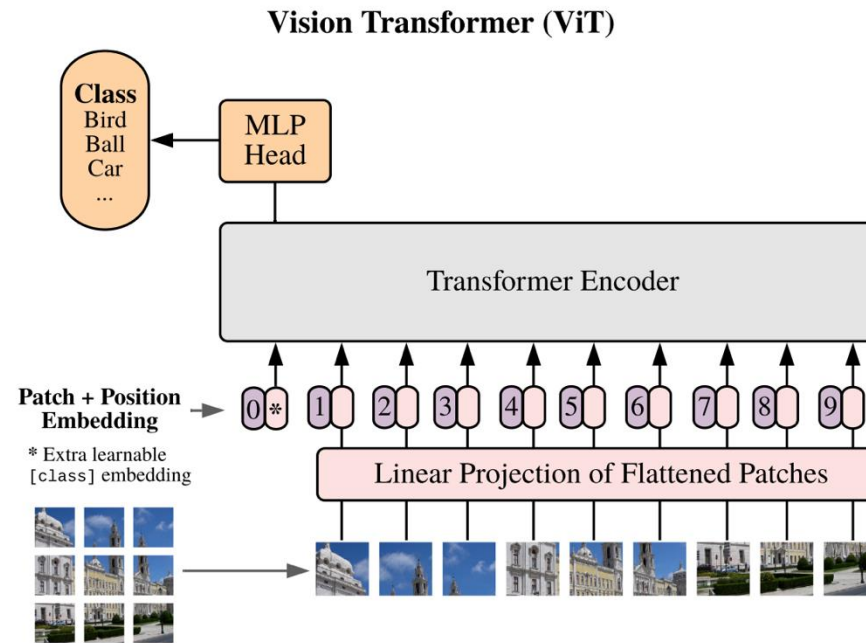
Encoder-Decoder for ChatGPT



Encoder for image classification

Last week: The Vision Transformer

- Break images into patches
- Perform linear projection
- Add position encoding and class embedding
- Encoder just like for text
- Final MLP head for classification on x_{class} token



$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1 \mathbf{E}; \mathbf{x}_p^2 \mathbf{E}; \dots; \mathbf{x}_p^N \mathbf{E}] + \mathbf{E}_{\text{pos}}, \quad \mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}, \mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+1) \times D}$$

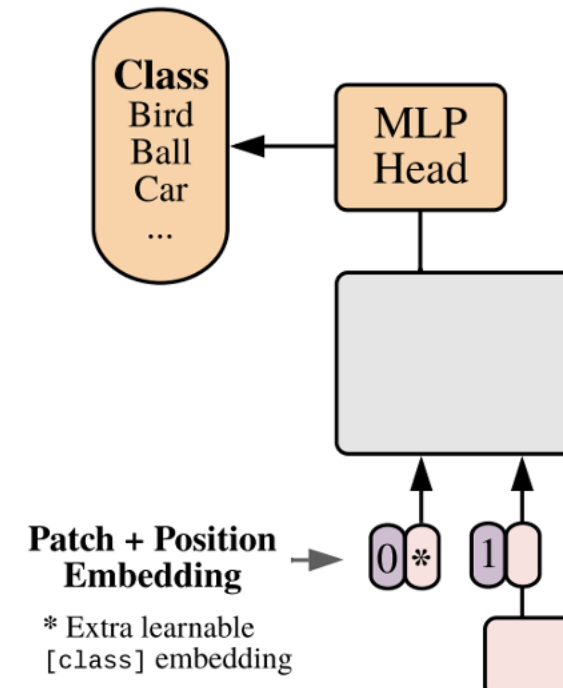
$$\mathbf{z}'_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \quad \ell = 1 \dots L$$

$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell, \quad \ell = 1 \dots L$$

$$\mathbf{y} = \text{LN}(\mathbf{z}_L^0)$$

The Class Token

- The class token represents the *entire* image
- **The class token is a summary of the image based on the relationships between image patches**
- It does so using the self-attention mechanism
 - It is "in touch" with every other token/patch
 - It gets transformed every single layer
- The last MLP head only looks at the class token
- The same can be done for text: summarize a sentence into a single token



Today

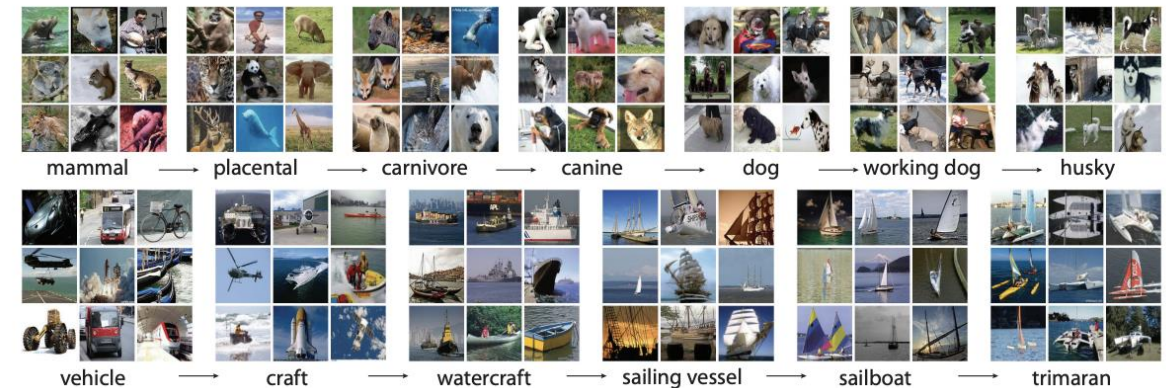
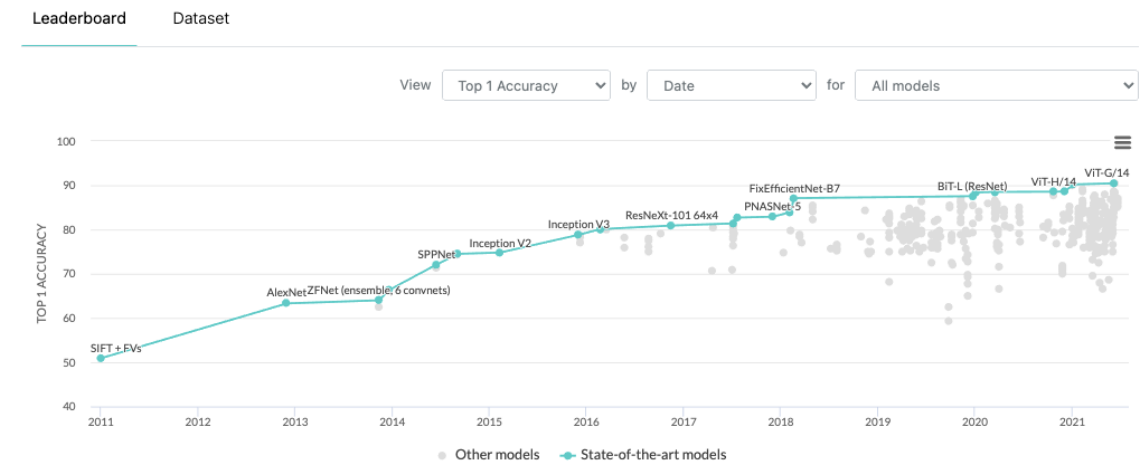
- Combining vision and text embeddings
- Paradigm shift in vision due to self-supervised learning
- Application: Open World Language Vision Transformers

State-of-the-Art before CLIP

- Vision Transformers (last week) outperform convolutional methods both in accuracy and training time
- Both are a supervised approach – “zero-shot”^{*} performance on unseen objects is very low
- Training requires additional labeled examples

^{*} Lampert, C. H., Nickisch, H., and Harmeling, S. Learning to detect unseen object classes by between-class attribute transfer. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 951–958. IEEE, 2009.

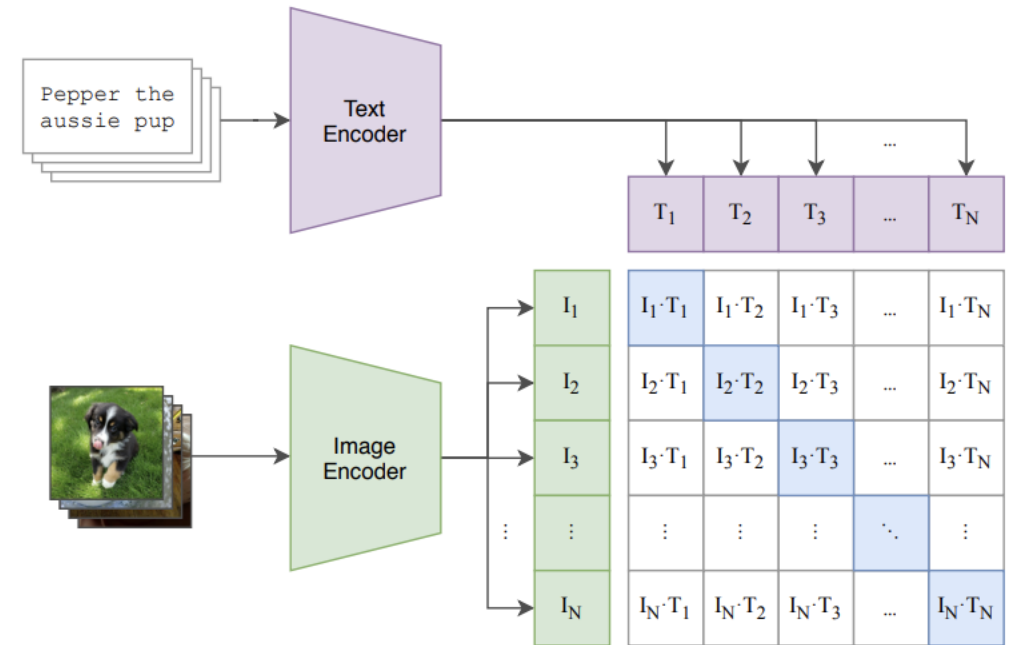
Image Classification on ImageNet



Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. and Fei-Fei, L., 2009, June. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). IEEE.

New idea: Contrastive Learning

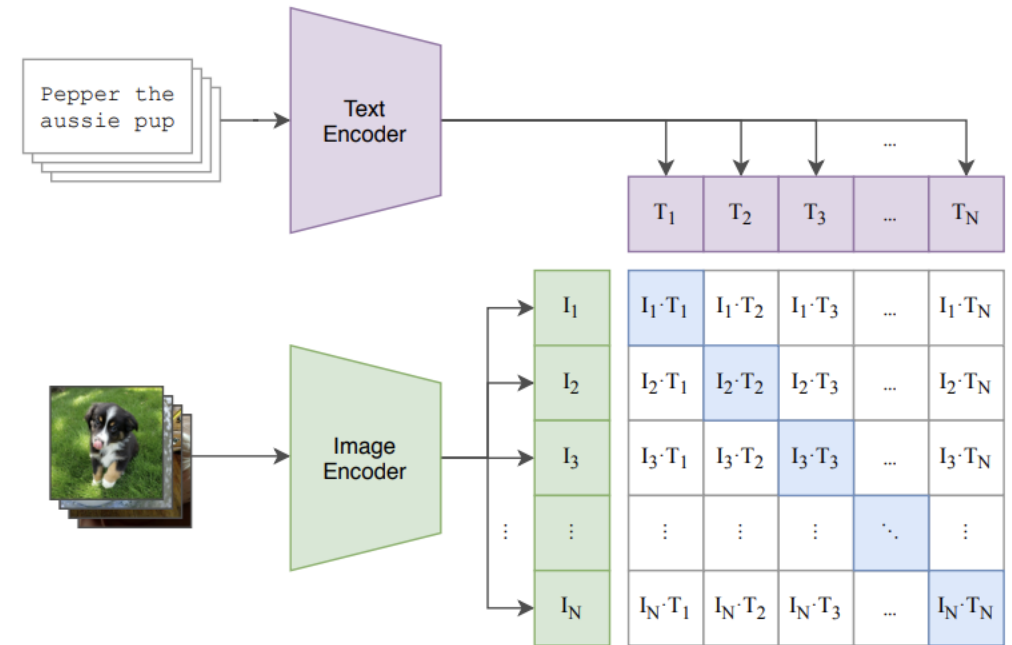
- Instead of hand-curated image-text pairs (300M for training ViT), use 400M text-image pairings from the internet
- Operate on batches of N
- **Maximize cosine similarity between correct pairs, minimize between all others**
- Train Text and Image Encoder together
- Symmetric cross entropy loss
- “Temperature” controls the range of logits in the SoftMax



Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Aspell, A., Mishkin, P., Clark, J. and Krueger, G., 2021, July. Learning transferable visual models from natural language supervision. In *International conference on machine learning* (pp. 8748-8763). PMLR.

Encoders

- Image
 - Various versions of ViT and ResNet
 - ViT-B/32, a ViT-B/16, and a ViT-L/14.
- Text
 - 63M-parameter 12- layer 512-wide model with 8 attention heads
 - Limit to 76 tokens
 - Bracketed with [SOS] and [EOS] tokens
 - [SOS]A picture of a dog[EOS]
 - [EOS] token is used as class token
 - Masking is used to ignore padding



Algorithm

- Perform learning in minibatches of size n of image/text pairs
- Embedding dimension of images d_i and d_t can be different
- Weight matrices W_i and W_t are used to project into d_e dimensional space
- Temperature sharpens/broadens the distribution
- Compute loss along the image-text and text-image directions

```
# 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
```

Algorithm

- I_e and T_e are $n \times d_e$ matrices
- Multiplying them results in a $n \times n$ matrix
- Labels are indicating which entry in each row is the correct one -> diagonal matrix
- Losses are computed both horizontally and vertically
- Losses are simply averaged

	T_1	T_2	T_3	...	T_N
I_1	$I_1 \cdot T_1$	$I_1 \cdot T_2$	$I_1 \cdot T_3$...	$I_1 \cdot T_N$
I_2	$I_2 \cdot T_1$	$I_2 \cdot T_2$	$I_2 \cdot T_3$...	$I_2 \cdot T_N$
I_3	$I_3 \cdot T_1$	$I_3 \cdot T_2$	$I_3 \cdot T_3$...	$I_3 \cdot T_N$
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots
I_N	$I_N \cdot T_1$	$I_N \cdot T_2$	$I_N \cdot T_3$...	$I_N \cdot T_N$

```
# 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
```

Temperature

- High temperature (>1) makes the model more relaxed, distribution is more uniform
- Low temperature (<1) sharpens the differences, increases confidence
- Example: image of a cat
 - Candidate 1: “a cat” \rightarrow 3.0
 - Candidate 2: “a dog” \rightarrow 1.5
- Low temperature: 90% cat
- High temperature: 60% cat

Optimizations

- **Mixed-precision**

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G. and Wu, H., 2017. Mixed precision training. *arXiv preprint arXiv:1710.03740*.

- **Gradient checkpointing**

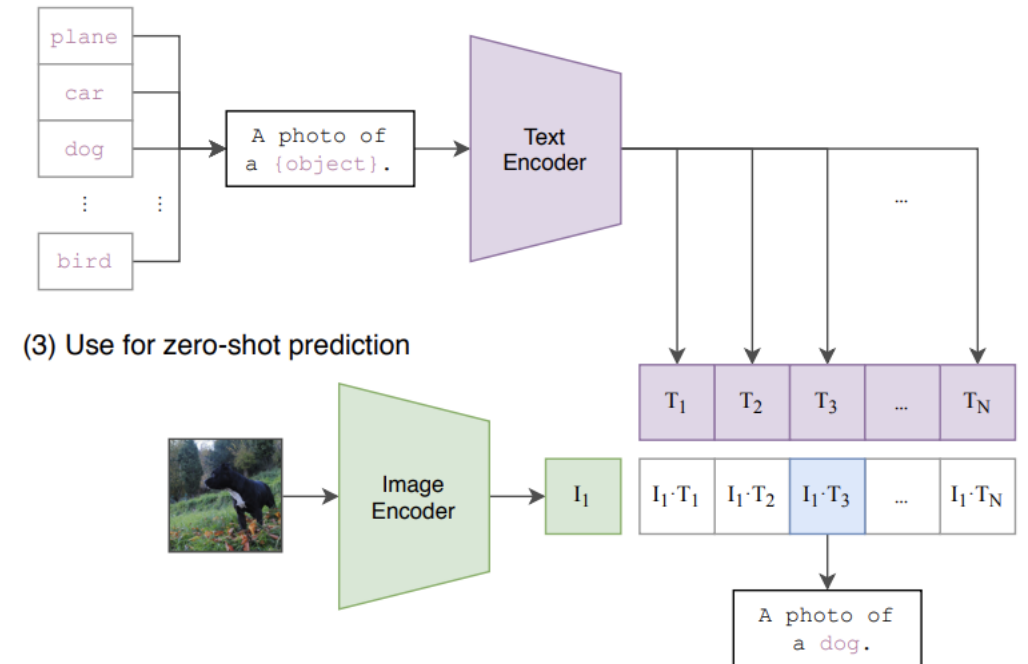
Griewank, A. and Walther, A. Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Transactions on Mathematical Software (TOMS)*, 26(1):19–45, 2000.

- **Half-precision Adam statistics**

Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A., and Sutskever, I. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

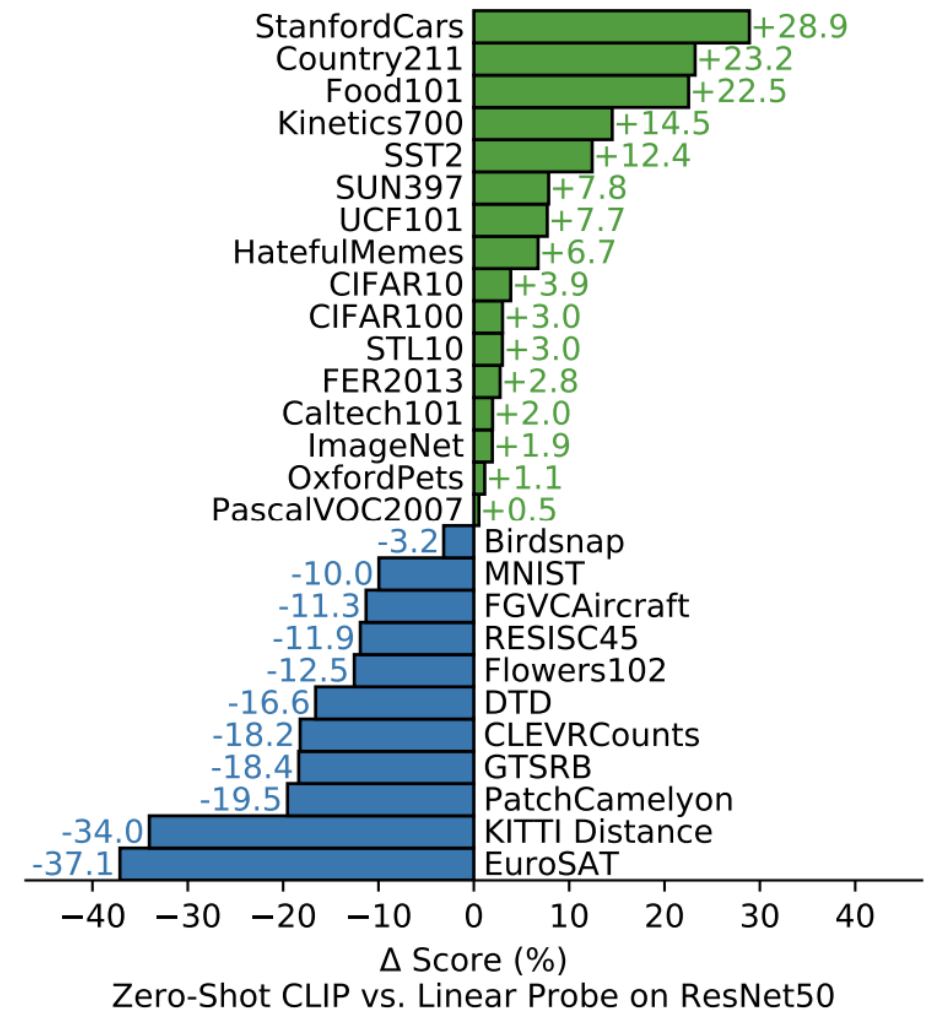
Inference

- Use trained image and text encoders to find most likely combination of text and image embedding
- Here: Find the most likely match for “A photo of a [object]”
- Text embeddings for all possible classes can be precomputed/cached
- Other combinations are possible, searching over all images in a database



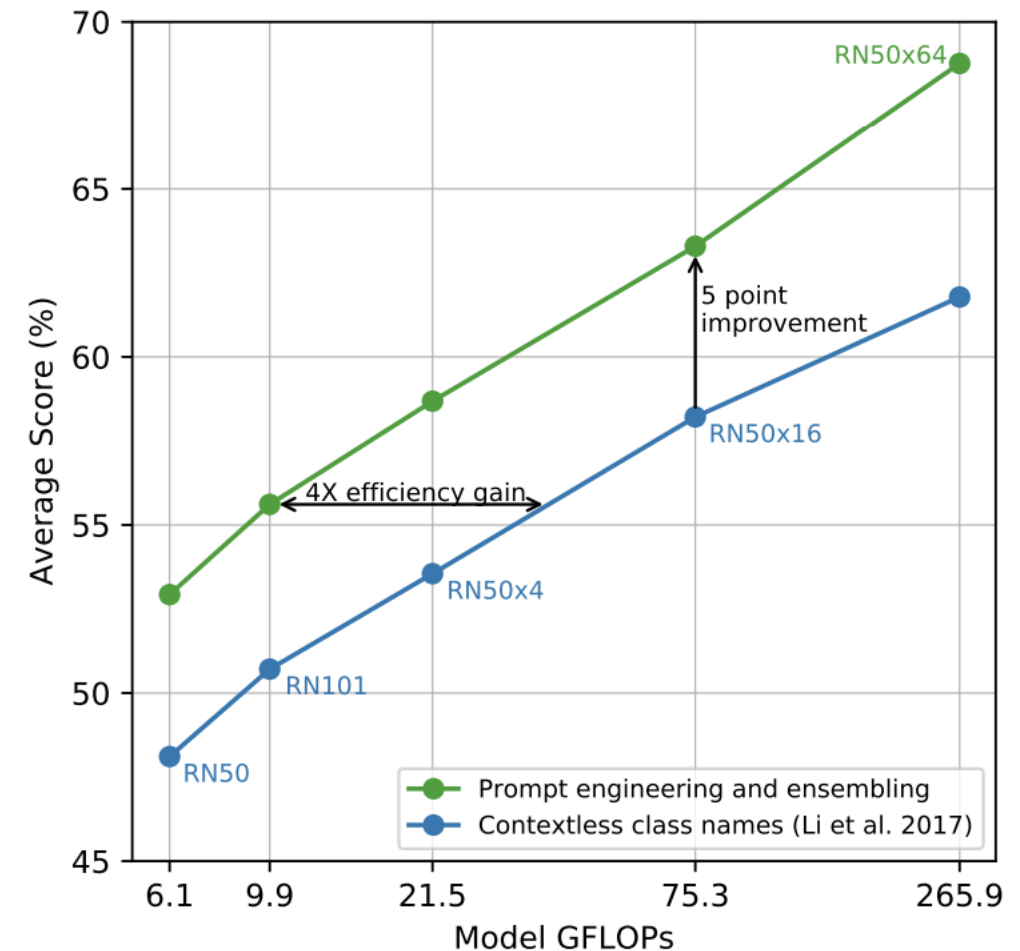
Zero-shot Performance vs. ResNet

- As good or better than ResNet on standard datasets
- Very good at activity recognition, possibly due to qualifying verbs
- Poor zero-shot capability on “expert” domains
 - German Traffic Signs (GTSRB)
 - Lymphnode tumor detection (PatchCamelyon)
 - Synthetic scences (CLEVRCounts)
 - Satellite imagery



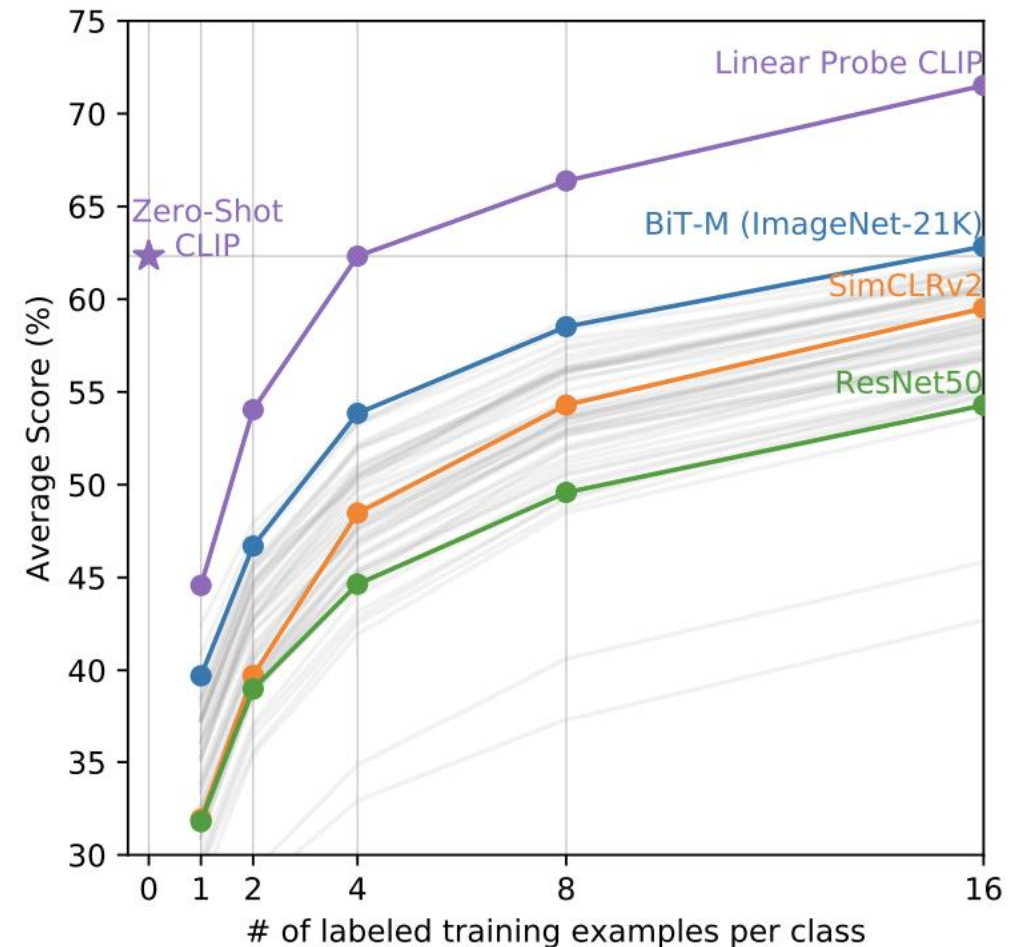
Issues with evaluating zero-shot capabilities

- Many datasets only contain class numbers, no class names
- Polysemy (e.g. bank, bat, boxer, crane) becomes problematic when training on text, not class number
- Images are rarely labeled with a single noun
- Prompt engineering helps
 - “A photo of a [object]”
 - “A photo of a [object], a type of pet/food/aircraft.”
 - “a satellite photo of a {label}.”
- Ensembling also helps
 - “A photo of a small [object]”
 - “A photo of a big [object]”
 - Averaging over the text embeddings, not the probabilities, resulting into a single query



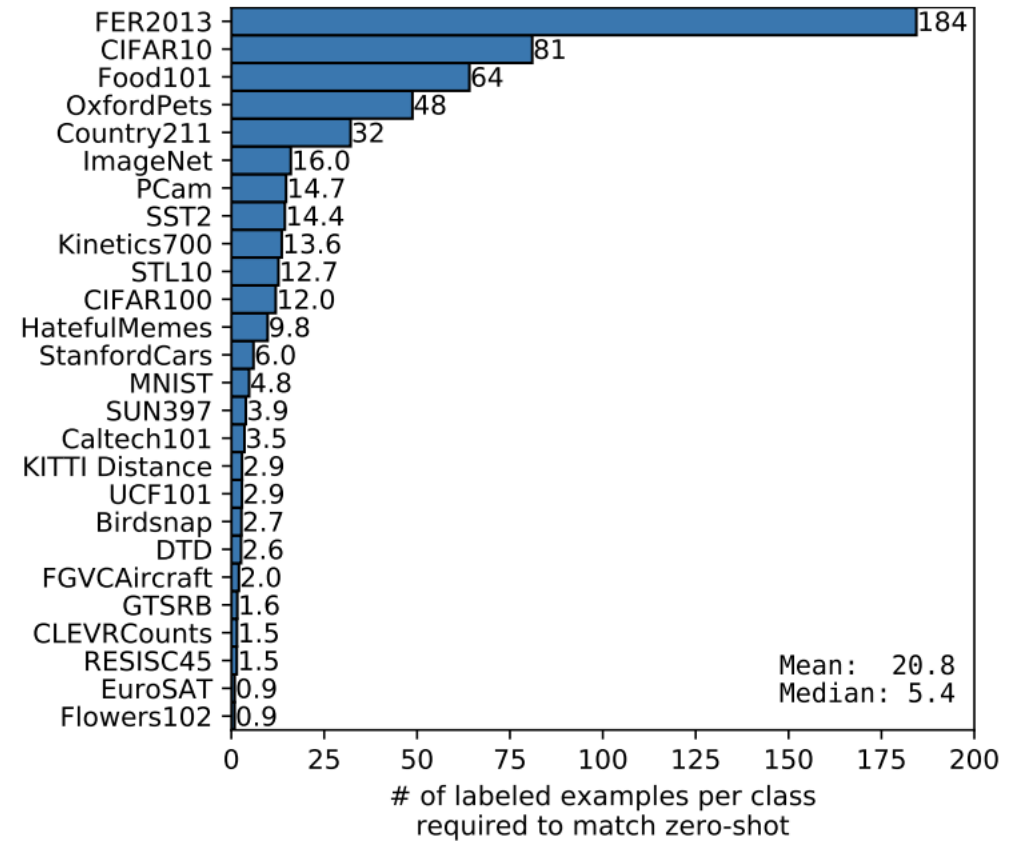
Multi-shot performance

- Multi-shot approach
 - Provide N sample images that contain the desired category
 - Compute N embeddings, average them
 - Compare sample with target image to compute likelihood (cosine similarity)
- Zero-Shot CLIP is as good as 4-shot CLIP, illustrating the power of the text prompt
- CLIP outperforms previous methods on multi-shot



Multi-shot via logistic regression

- Compute embeddings of N sample images
- Use logistic regression to determine class
- Due to contrastive pre-training with text embeddings images with similar content are closer in embedding space
- This works better for some images than others



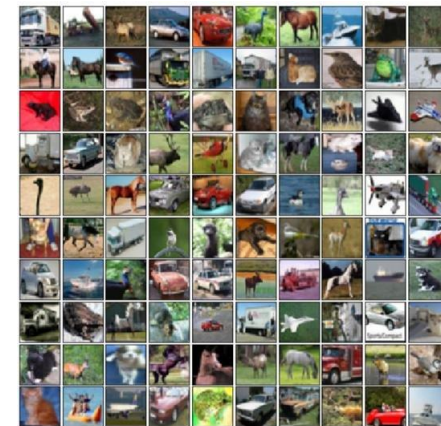
$$z = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b = \mathbf{w}^\top \mathbf{x} + b$$

Where CLIP embeddings reach their limitations

- FER2013 (worst) contains emotional expressions that might not be well represented in the 400M training images
- Flowers102 (best) depict flower species for which ample examples exist
- CIFAR10 (second worst) is very low resolution, possibly not matching well what the CLIP-trained image encoder learned to distinguish



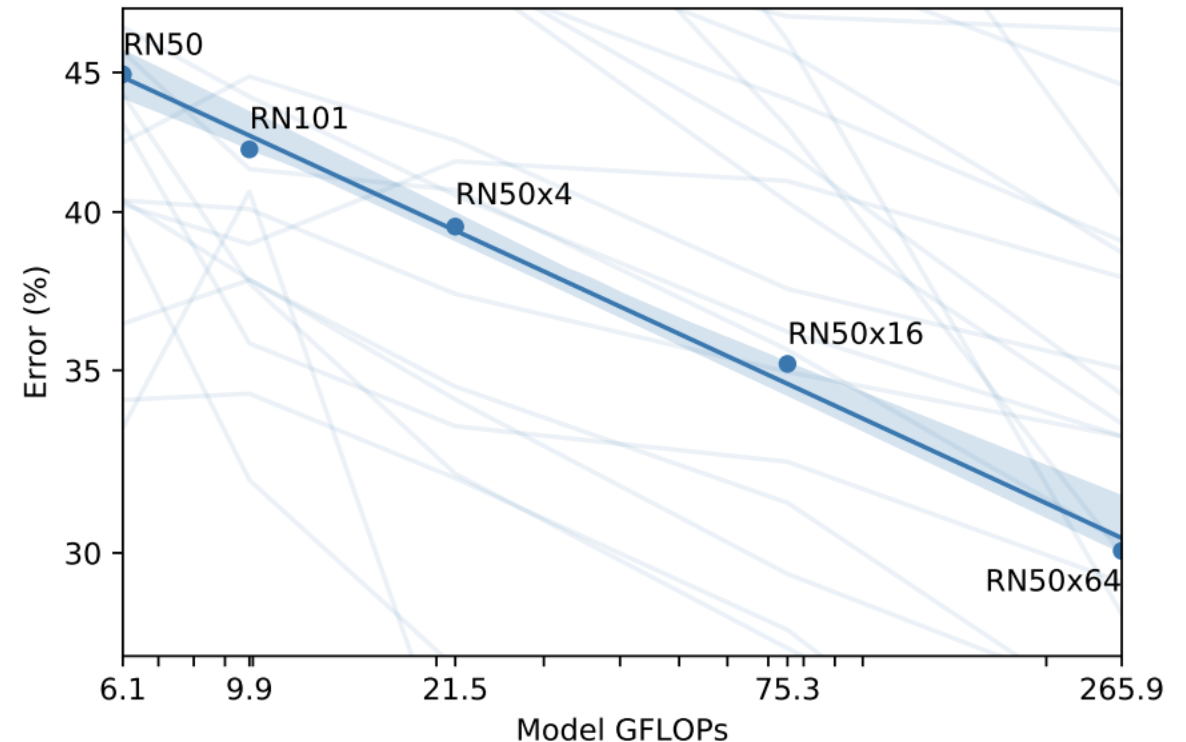
FER2013



CIFAR10

Scaling

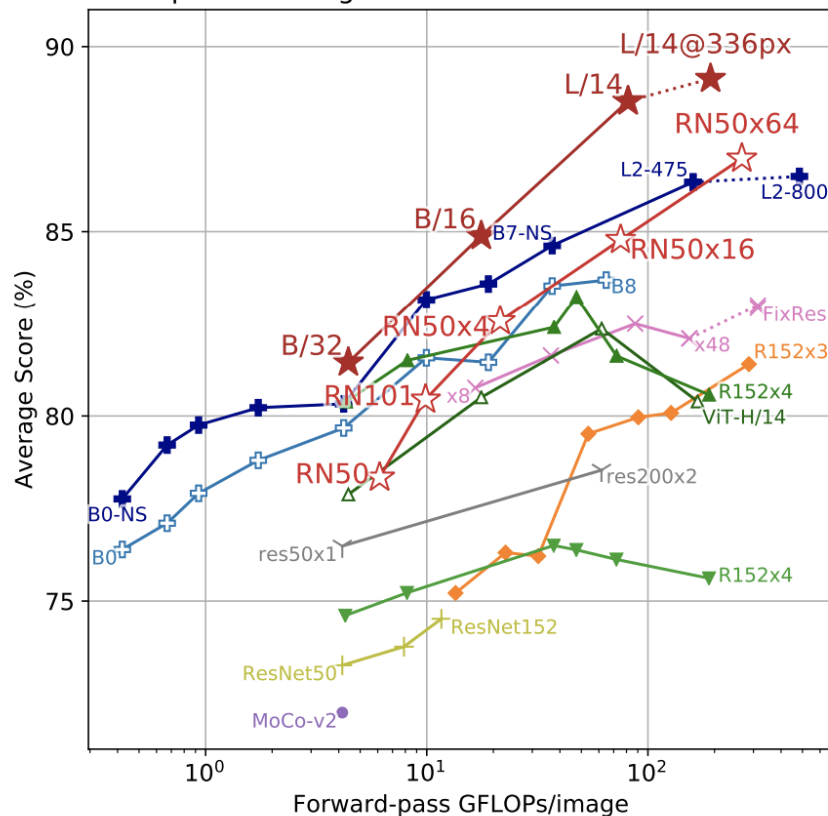
- ChatGPT has shown linear improvement with growing dataset size / compute (up to 10000x)
- CLIP shows the same trend on average across 44x compute improvements
- Individual trials vary, however



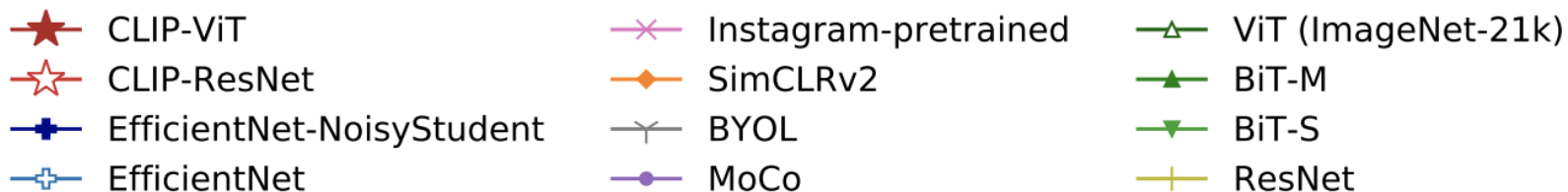
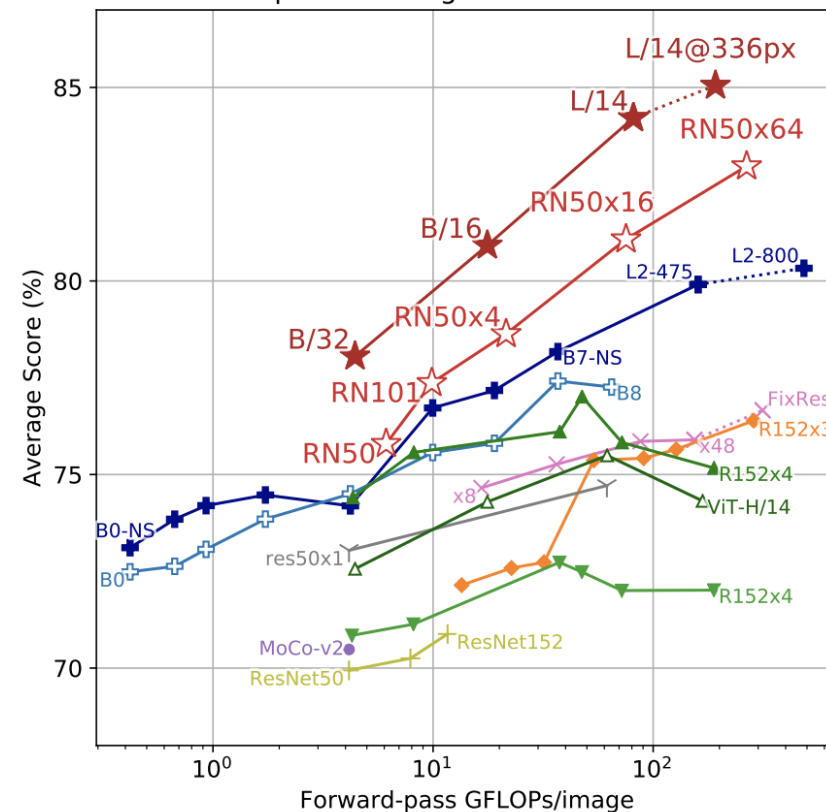
Efficiency

- CLIP provides highest accuracy for the same amount of computation than other models
- ViT outperforms ResNet
- Task diversity further emphasizes CLIP's representational power

Linear probe average over Kornblith et al.'s 12 datasets





















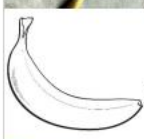











Linear probe average over all 27 datasets



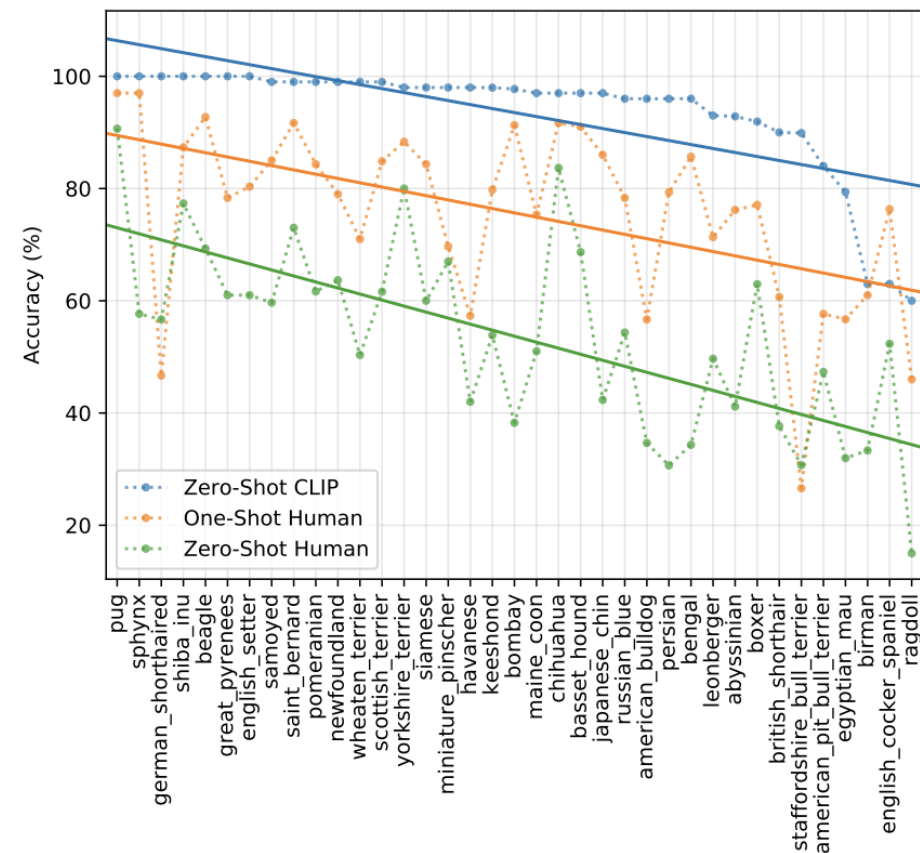
Resilience to Distribution Shift

- Various datasets explore “distribution shifts”
- Examples here are ordered by decreased performance

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%

Few-shot learning in CLIP vs. human performance

- Human performance quickly increases when provided only one (1-shot) example when determining dog breeds
- This is not the case for CLIP – 1-shot helps very little
- Further improvement by adding training data are likely

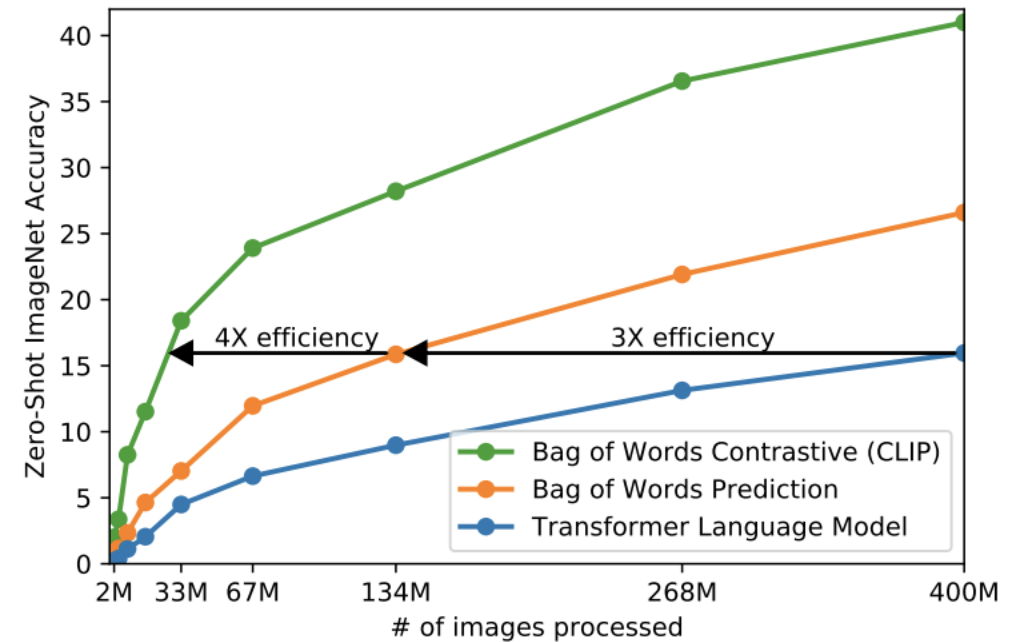


Limitations

- CLIP is not good at distinguishing car, airplane or flower types and other task where fine details matter (that are usually not in the label)
- CLIP is not good at abstract tasks such as counting instances or measuring distance to an object

Notes

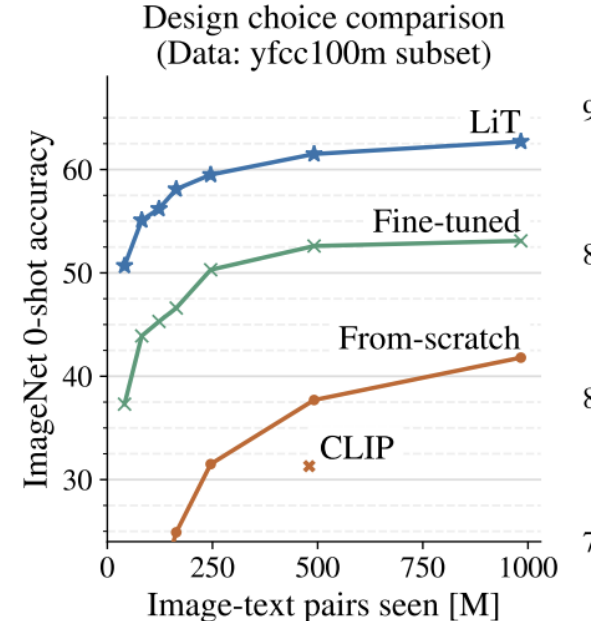
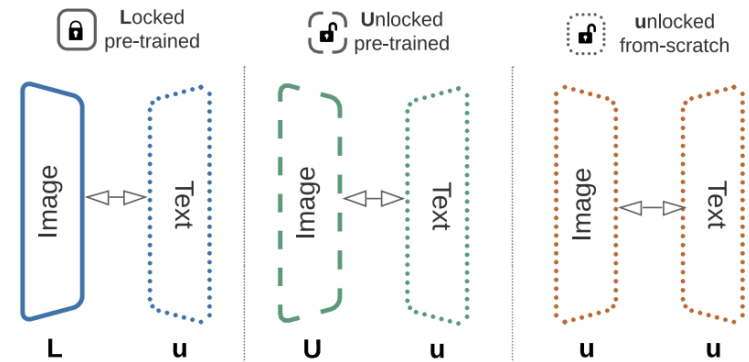
- The CLIP paper also provides an interesting summary on *how* they arrived at this model
 - Initial attempt: predicting labels word-for-word based on image embedding
 - Simplification: predicting whether a label is the correct one
 - Finally: contrastive learning
- Generally not a good way to write a paper



LiT: Zero-Shot Transfer With Locked-Image Text Tuning

- Instead of pre-training image and text together, lock the image encoder (e.g. expensive ViT model), and train only text encoder
- Builds up on the insight that Vision Transformers develop fundamental, general vision architectures

Zhai, X., Wang, X., Mustafa, B., Steiner, A., Keysers, D., Kolesnikov, A. and Beyer, L., 2022. Lit: Zero-shot transfer with locked-image text tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 18123-18133).



Lab

- Colab notebook needs two integrations
 - Your google drive – to save model
 - Your Huggingface account – to access ClothingMNIST
 - *Both are important capabilities/tools for your project*
- Tasks
 1. Make sure, every line of code is clear!
 2. Increase training accuracy
 3. Identify pre-trained embeddings
 4. Identify pre-trained CLIP models

Mini-project

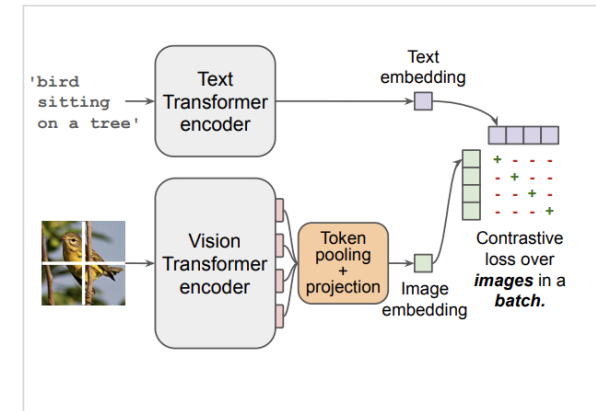
- Train your own CLIP model
 - Train CLIP from scratch on your own data
 - Use a pre-trained image and/or text embedding
 - Improve the ClothingMNIST performance by generating more descriptive labels
 - “A simple idea worth trying is joint training of a contrastive and generative objective with the hope of combining the efficiency of CLIP with the flexibility of a caption model.”
 - ...
- Document your work on a Blog article
 - Build up on Matthew Nguyen’s article
 - Explain at least one new aspect, e.g. ablation study, change in model structure, compare pre-training with from-scratch training, *show that you learned something new*
- Submit to Canvas
- *This is not the final project, don’t overdo it*

Open World Language – Vision Transformer

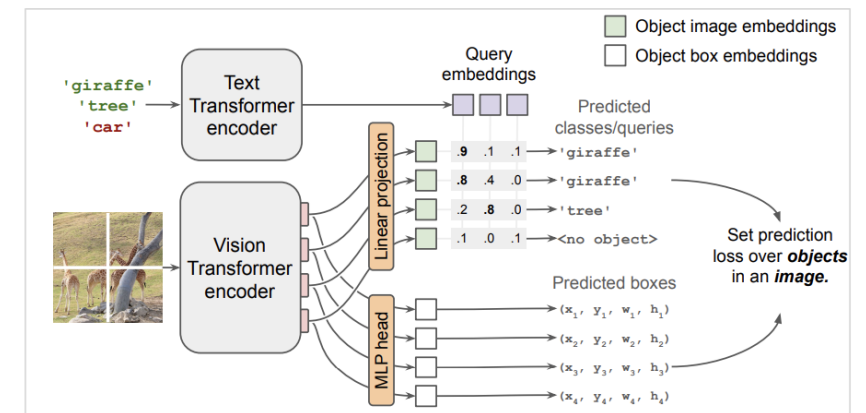
- Trains like CLIP
- Instead of CLS token, all patches are used to check against queries
- Each patch predicts a bounding box
- Bounding box are merged using Non-Maximum Suppression (NMS)
- One object per patch
- Two step training:
 - CLIP-like Image and text-encoder pretraining
 - Training of bounding box detector using COCO dataset

Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., Mahendran, A., Arnab, A., Dehghani, M., Shen, Z. and Wang, X., 2022, October. Simple open-vocabulary object detection. In *European Conference on Computer Vision* (pp. 728-755).

Image-level contrastive pre-training

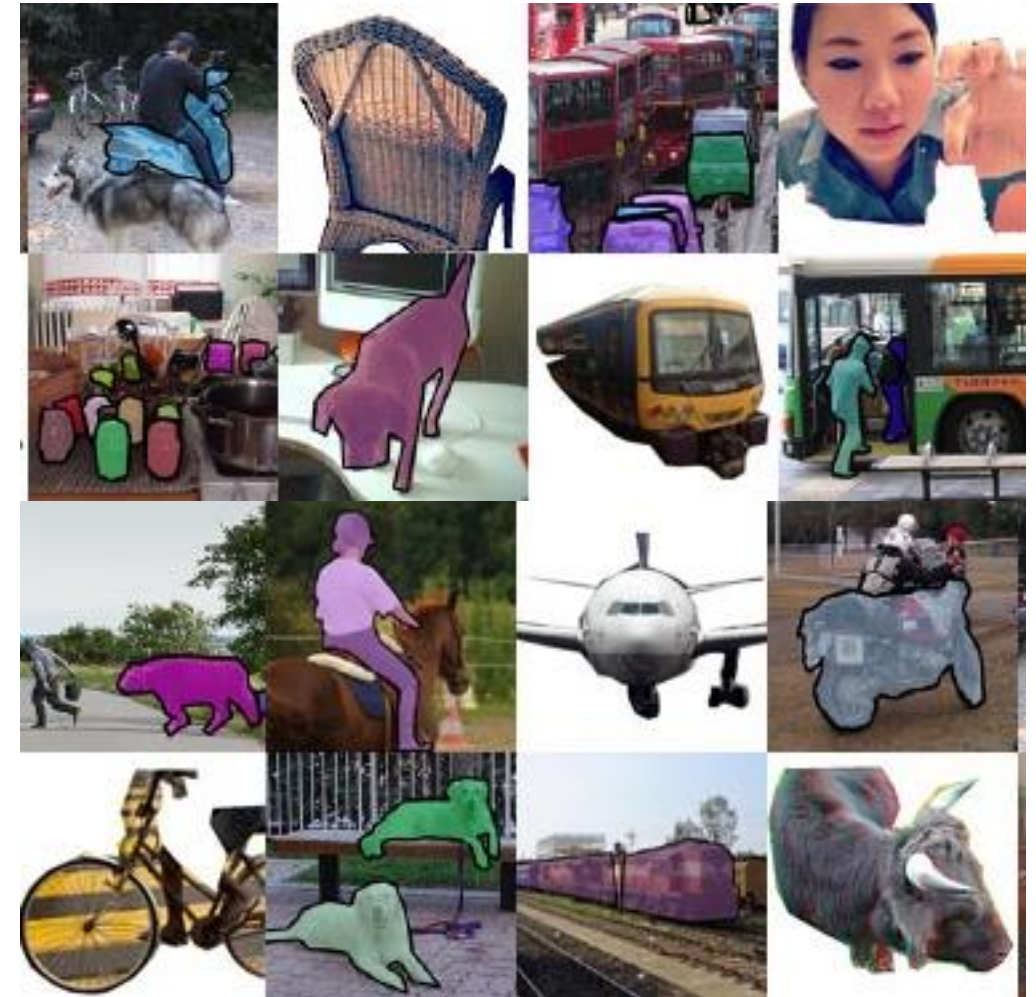


Transfer to open-vocabulary detection



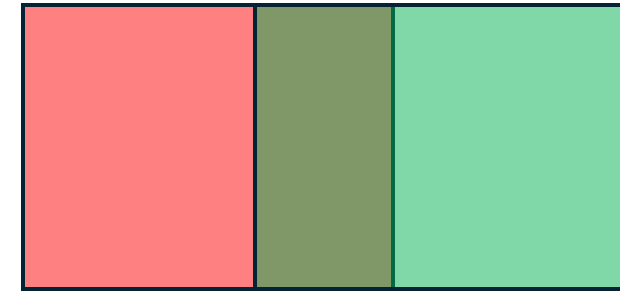
COCO Dataset

- Properties
 - 330k images
 - 80 object categories
 - 91 stuff categories
 - 250,000 people
- Even though the classes in COCO are limited, OWL-ViT understands all the CLIP embeddings
- Opportunity (?): generate training data via SAM from robots in the wild



Non-Maximum Supression

- Intersection over Union (IoU)
 - IoU of 0 means no overlap
 - IoU of 1 means complete overlap
- Sort all bounding boxes for an object by confidence
- Pick the most likely one, remove all overlapping ones
- Historically discrete algorithm, can also be learned, e.g.



$\text{IoU} =$

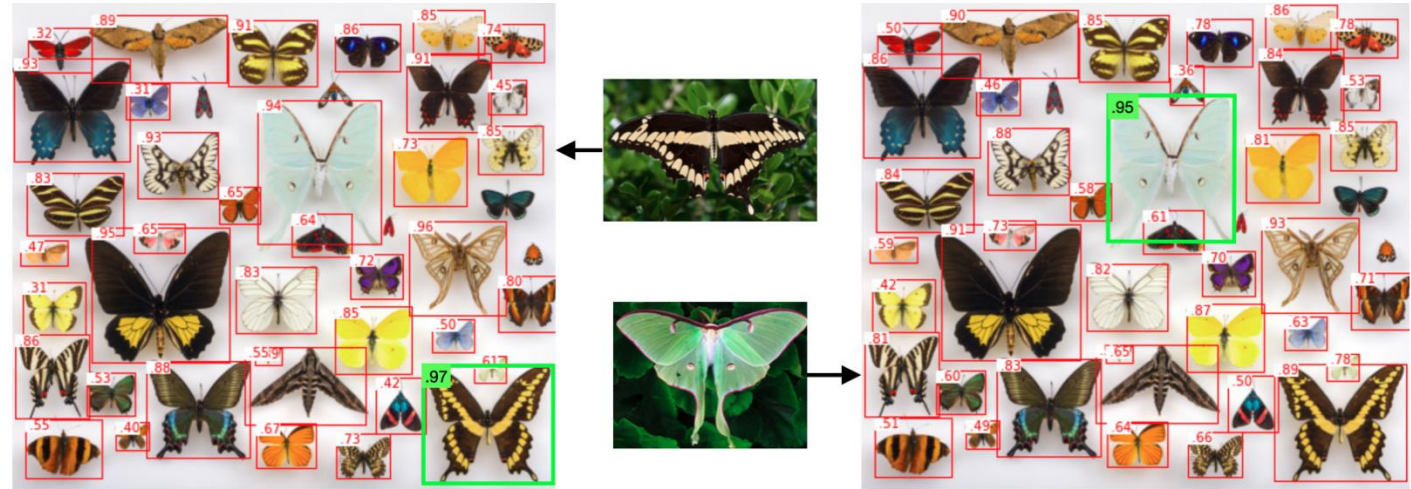
Area of
Overlap

Area of Union

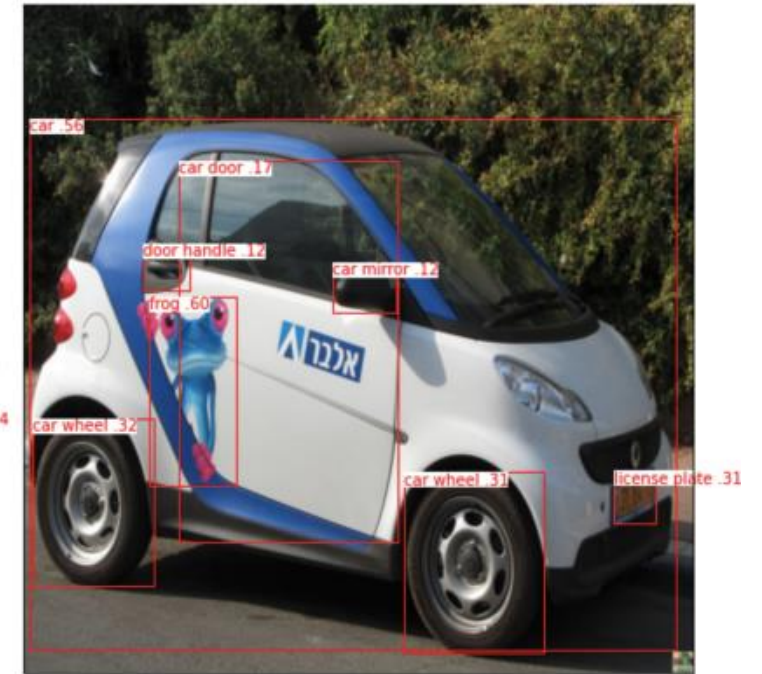
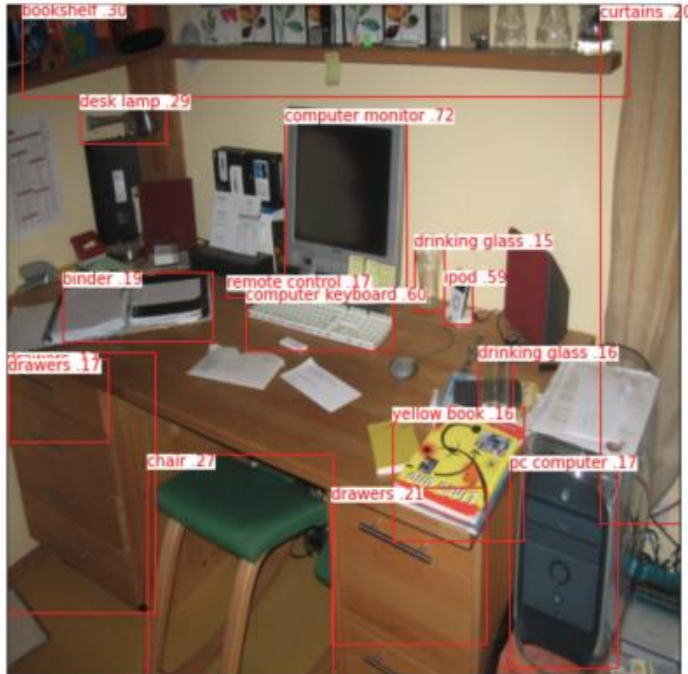
Hosang, J., Benenson, R. and Schiele, B., 2017. Learning non-maximum suppression. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4507-4515).

Querying OWL-VIT

- Based on CLIP, queries can be either text or image embeddings
- This is also called “one-shot” classification, because the query image represents a training image



More examples



Query: An image of a {object}

What's next?

- Adding LLMs to CLIP to generate labels/explanations: VLMs
- Finetune VLMs to generate robot trajectories: Open VLA
- Using CLIP to control diffusion: Dall-E