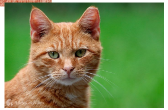
Learning Transferable Visual Models From Natural Language Supervision

Presentation by:
Muhammad Ferjad Naeem

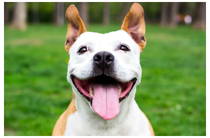
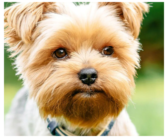
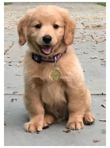
“Classic” Image Classification

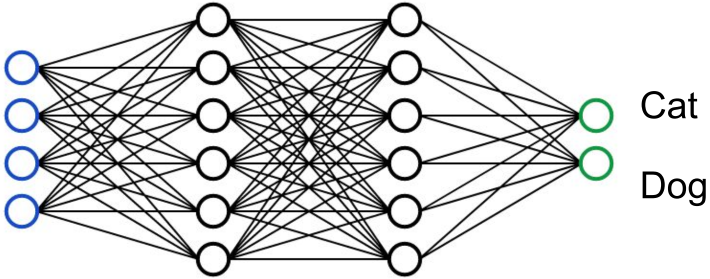




Cats

Dogs

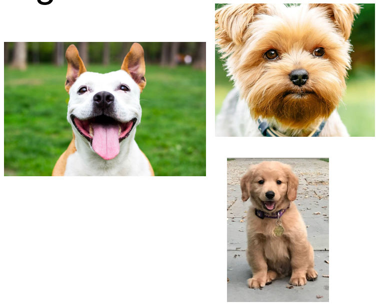






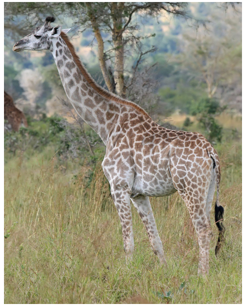
“Classic” Image Classification

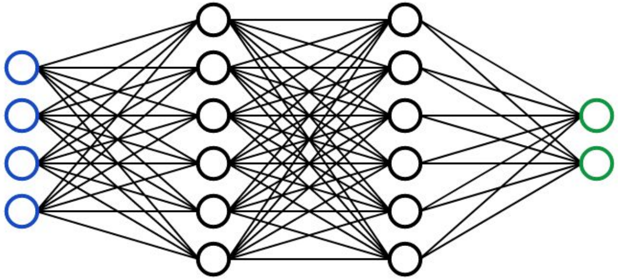
Cats

Dogs



What if I want to add a Giraffe class now?





Cat

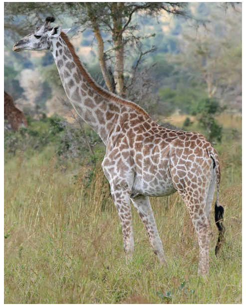
Dog

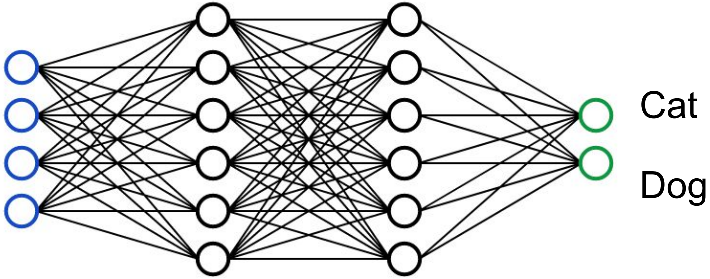
"Classic" Image Classification

Cats

Dogs

What if I want to add a Giraffe class now?





- Collect enough labelled images for Giraffe
- Retrain the model with cross entropy loss

One label classification setting is not flexible

- What is in this image?
  - Zebra
  - Goat
  - Grazing plane

Humans don’t describe images to single words often.

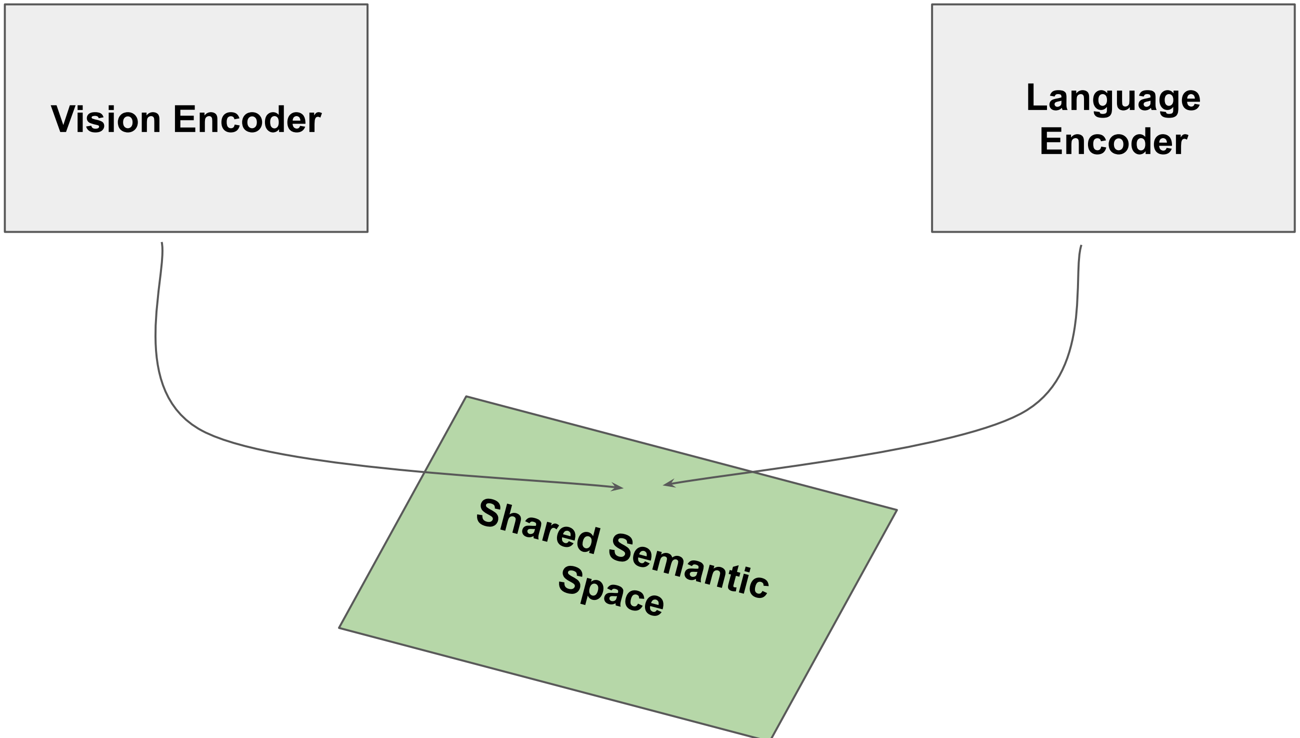
A zebra and three goats are grazing on a grass plane

Enjoying a nice sunset on a beach. The day is clear with some clouds.

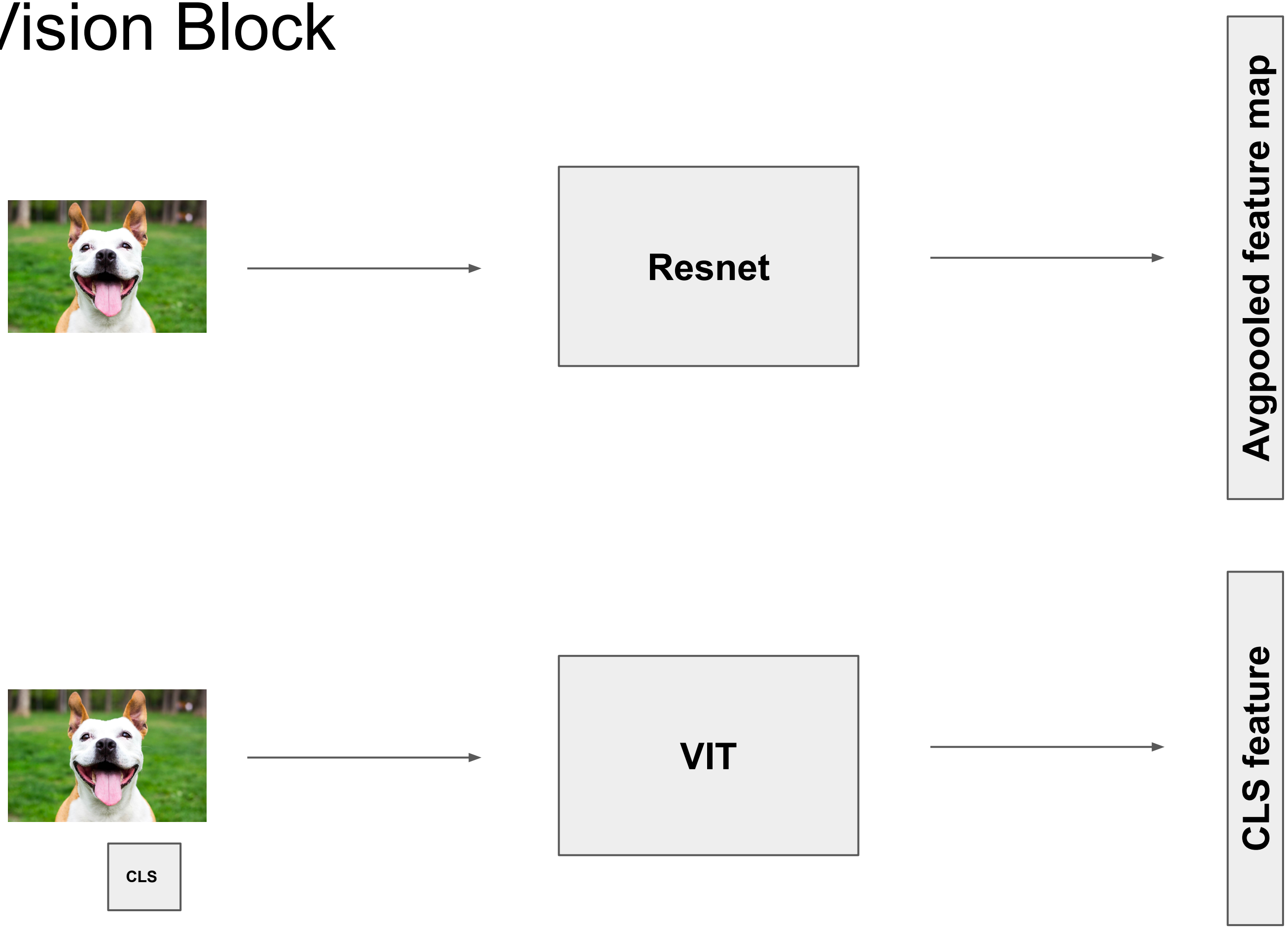
The man in the red shirt tackles the man in the white shirt during a football game.

Can we introduce the nuance of language in our Vision Models?

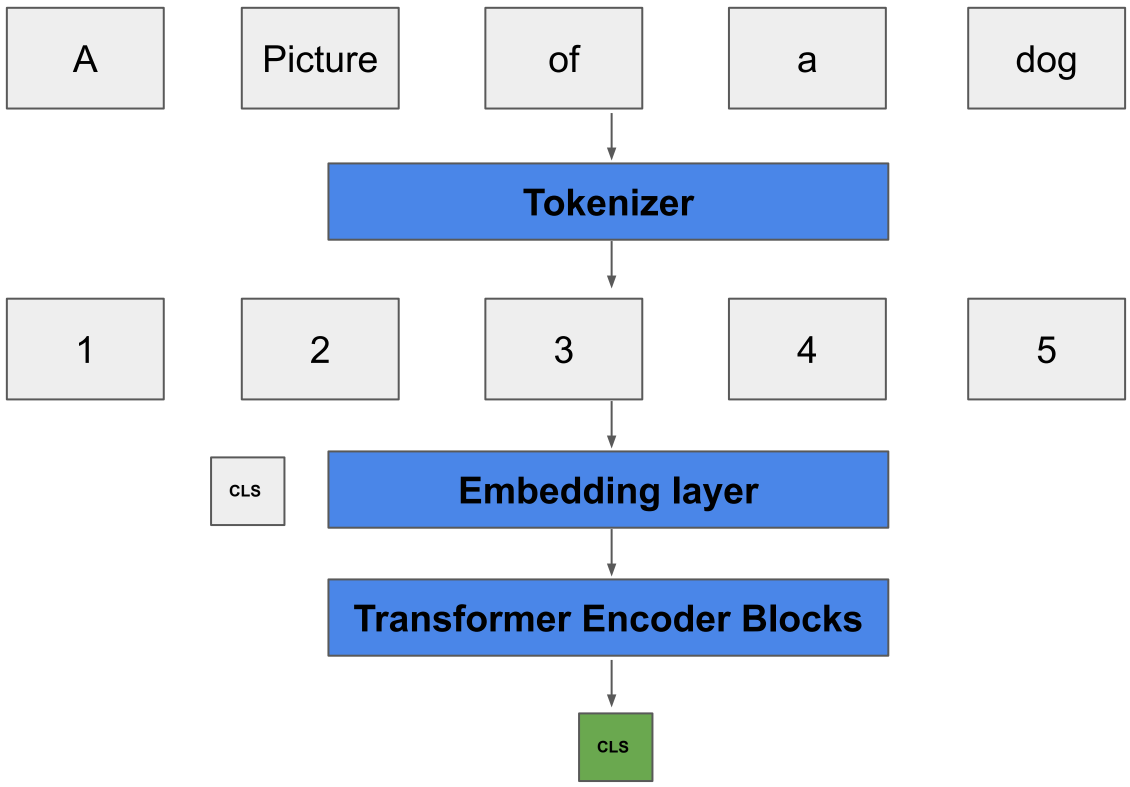
The building blocks to achieve this



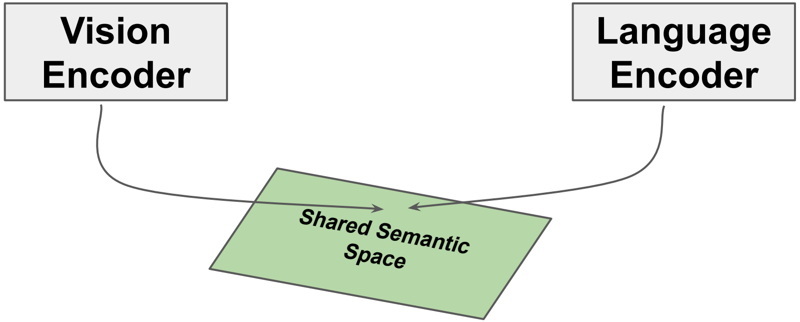
The Vision Block

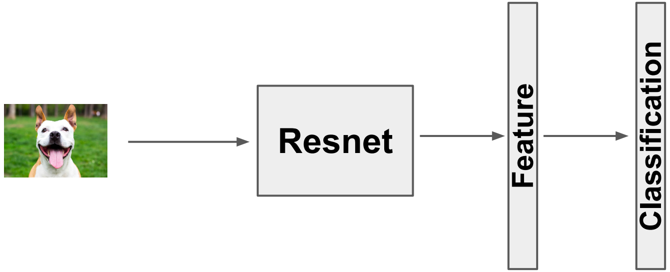


The language block



The Contrastive loss vs Classification loss





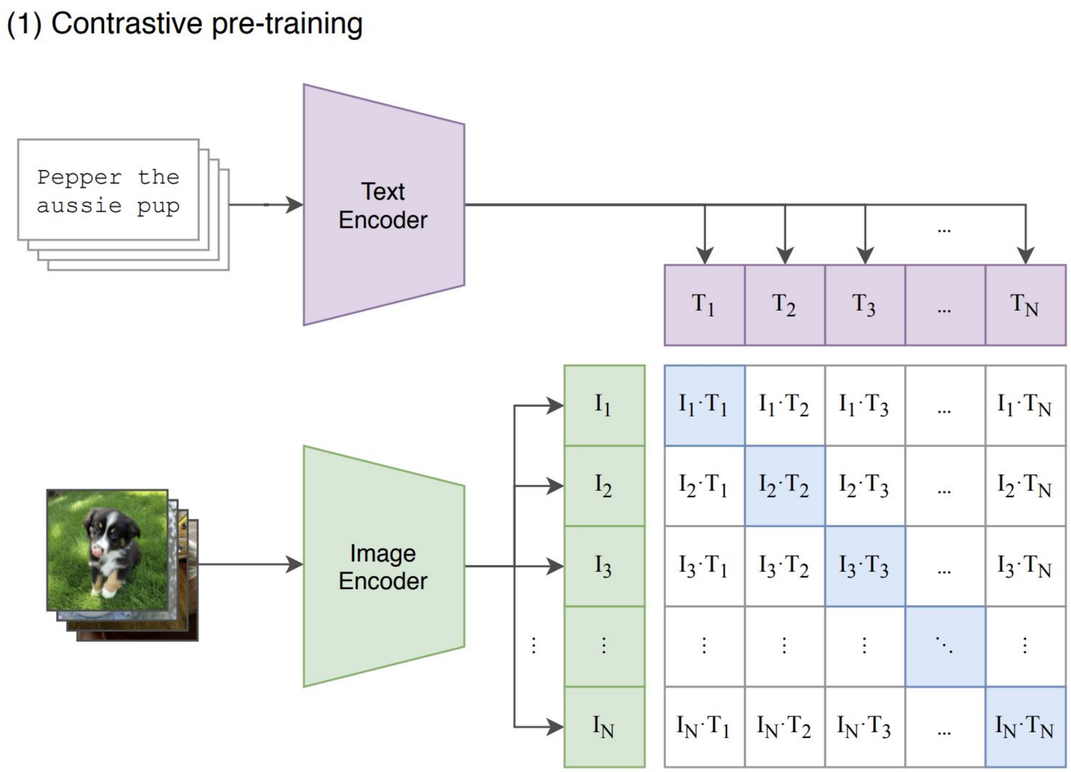
The two representations of the same concepts should be aligned together

The embedding of the image should be classified into the correct class

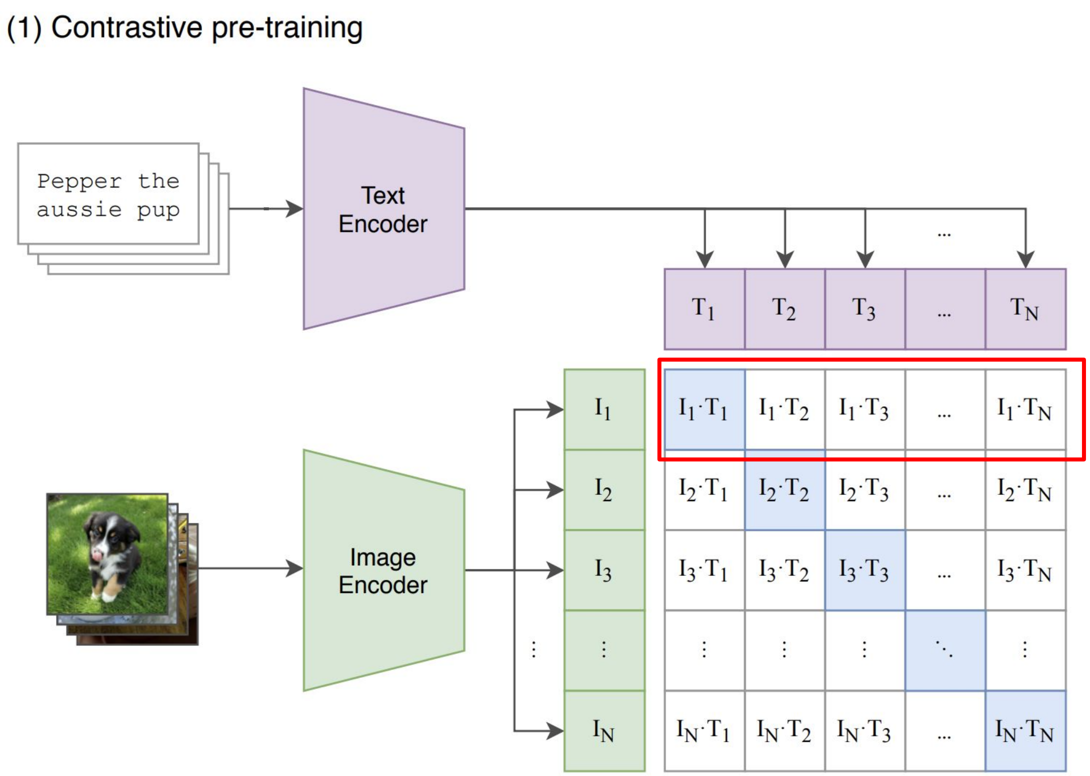
$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)},$$

The contrastive loss function

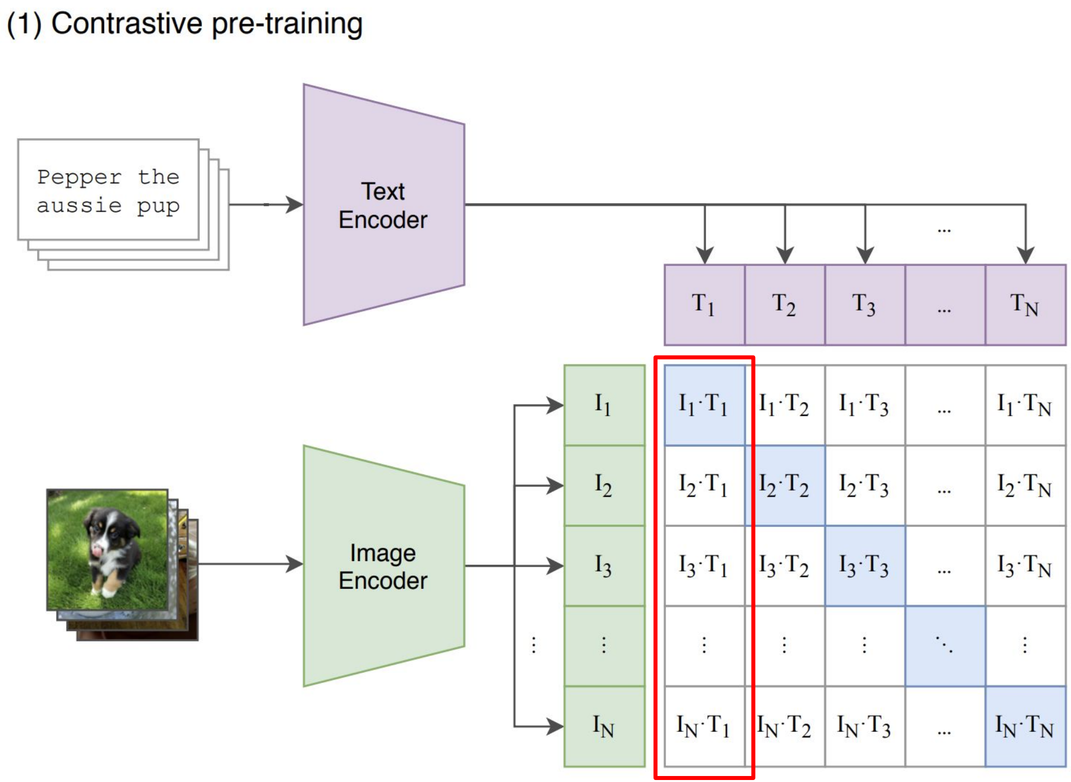
The CLIP Model while training



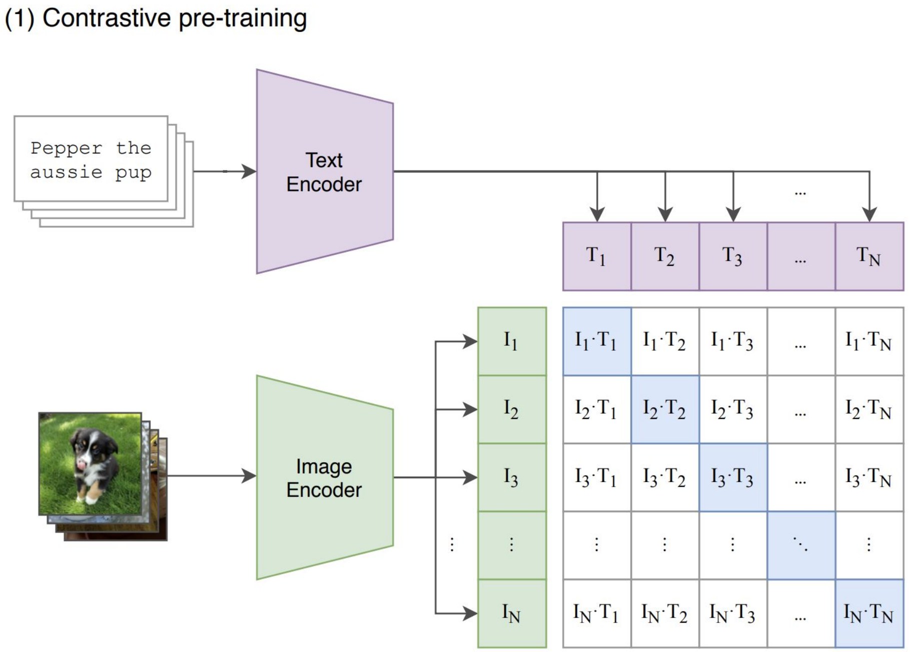
The CLIP Model while training



The CLIP Model while training

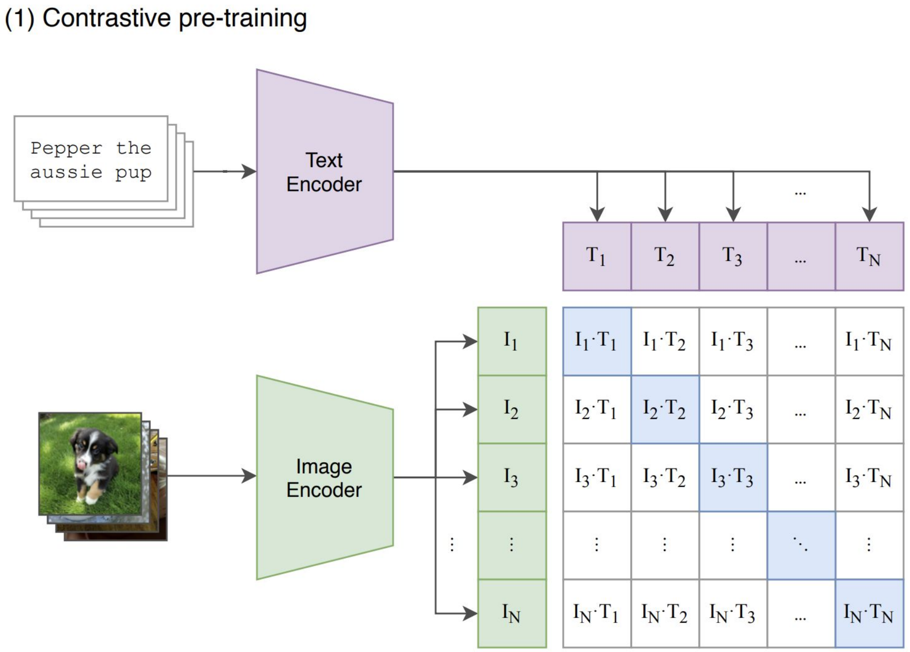


The CLIP Model while training




```
# 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
```

The CLIP Model while training



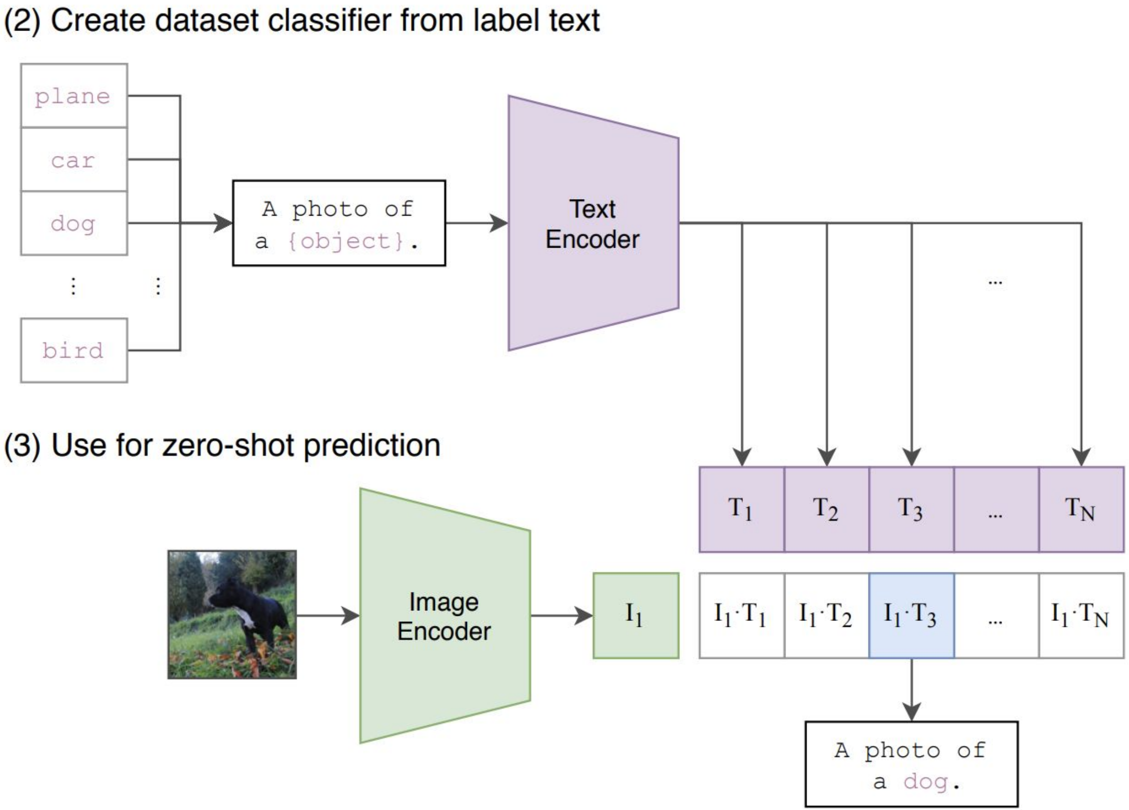

```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned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

Contrastive loss is batch size and data hungry!

Training details

- Closed source WIT dataset consisting of 400M image and caption pairs
- Batch size of 32,000
- Trained on 256V100 for 12 days

Using the pretrained backbone for inference



The Zero-Shot Transfer paradigm

- CLIP is trained on 400M Image Caption pairs from the internet
- This training data has covered almost all concepts available
- Use this pretrained model to transfer to datasets using language prompts

Measuring Zero-shot transfer across diverse CV datasets

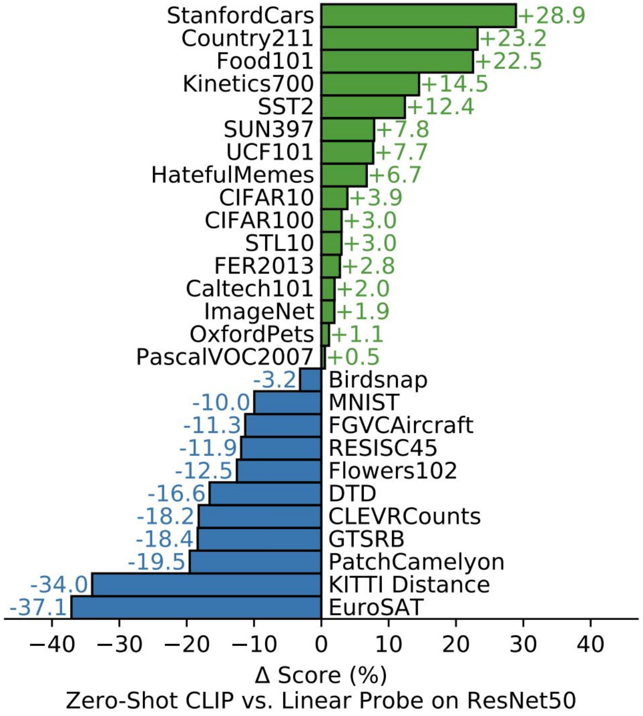


Figure 5. **Zero-shot CLIP is competitive with a fully supervised baseline.** Across a 27 dataset eval suite, a zero-shot CLIP classifier outperforms a fully supervised linear classifier fitted on ResNet-50 features on 16 datasets, including ImageNet.

Prompt Engineering and ensembling can improve performance without any retraining!

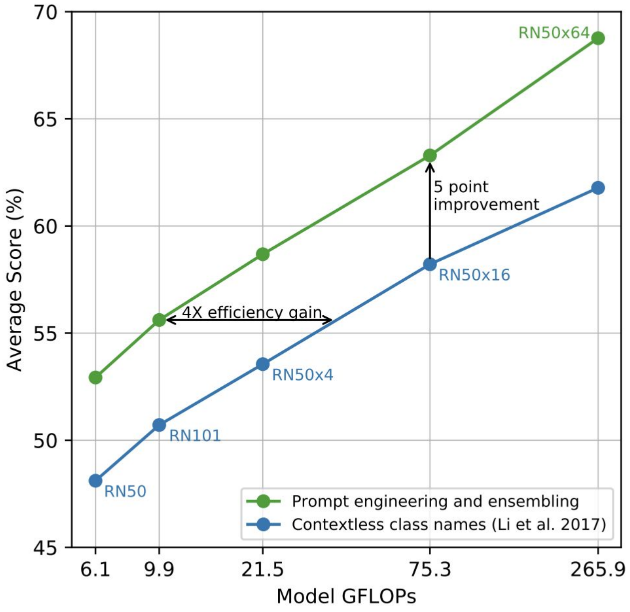


Figure 4. **Prompt engineering and ensembling improve zero-shot performance.** Compared to the baseline of using contextless class names, prompt engineering and ensembling boost zero-shot classification performance by almost 5 points on average across 36 datasets. This improvement is similar to the gain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

Prompt Engineering and ensembling can improve performance without any retraining!

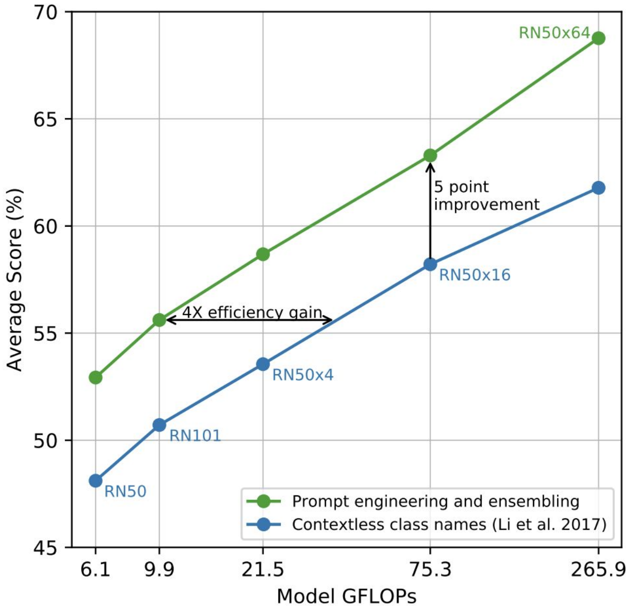


Figure 4. **Prompt engineering and ensembling improve zero-shot performance.** Compared to the baseline of using contextless class names, prompt engineering and ensembling boost zero-shot classification performance by almost 5 points on average across 36 datasets. This improvement is similar to the gain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

- a photo of a _
- a photo of many _
- a drawing of a _
- a painting of the _
- a pixelated photo of the _

A total of 80 such handcrafted prompts

Contrastive objective vs caption prediction

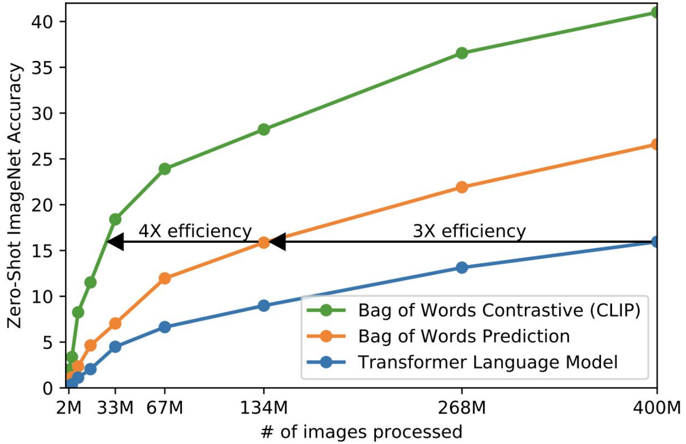


Figure 2. **CLIP is much more efficient at zero-shot transfer than our image caption baseline.** Although highly expressive, we found that transformer-based language models are relatively weak at zero-shot ImageNet classification. Here, we see that it learns 3x slower than a baseline which predicts a bag-of-words (BoW) encoding of the text (Joulin et al., 2016). Swapping the prediction objective for the contrastive objective of CLIP further improves efficiency another 4x.

Linear probing protocol

- Take a pretrained representation/ classification model
- Keep the feature extractor frozen and only train the linear classification layer using training dataset supervision
- Evaluate on the test set

Linear probe Evaluation and data efficiency

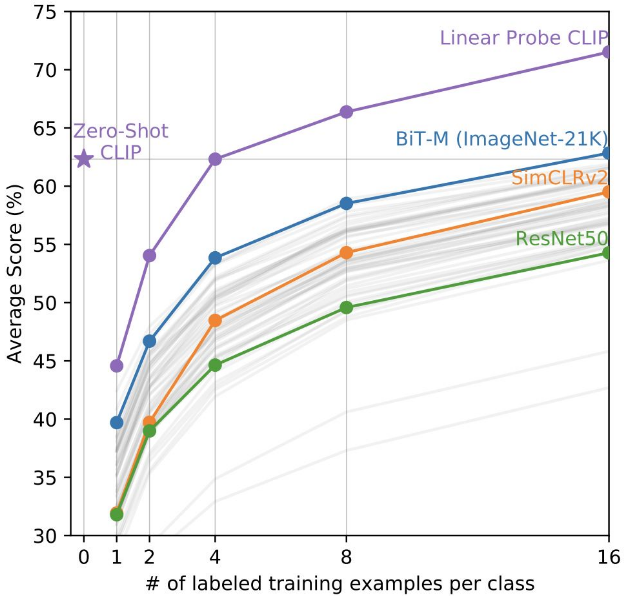


Figure 6. **Zero-shot CLIP outperforms few-shot linear probes.** Zero-shot CLIP matches the average performance of a 4-shot linear classifier trained on the same feature space and nearly matches the best results of a 16-shot linear classifier across publicly available models. For both BiT-M and SimCLRv2, the best performing model is highlighted. Light gray lines are other models in the eval suite. The 20 datasets with at least 16 examples per class were used in this analysis.

Linear probe Evaluation and data efficiency

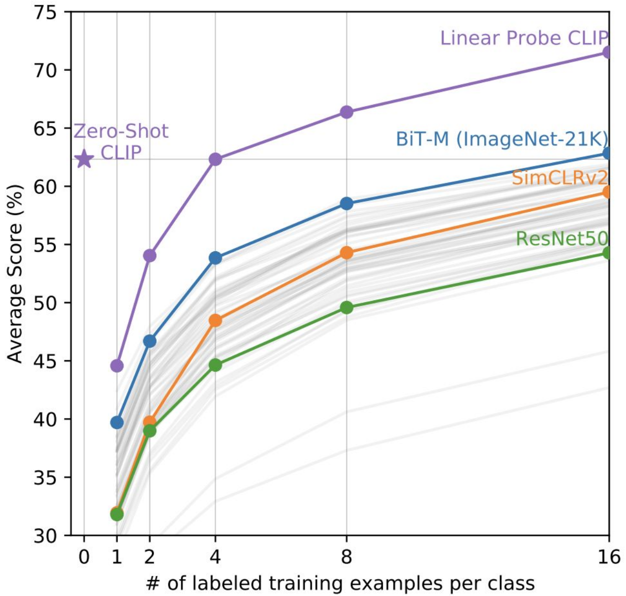


Figure 6. **Zero-shot CLIP outperforms few-shot linear probes.** Zero-shot CLIP matches the average performance of a 4-shot linear classifier trained on the same feature space and nearly matches the best results of a 16-shot linear classifier across publicly available models. For both BiT-M and SimCLRv2, the best performing model is highlighted. Light gray lines are other models in the eval suite. The 20 datasets with at least 16 examples per class were used in this analysis.

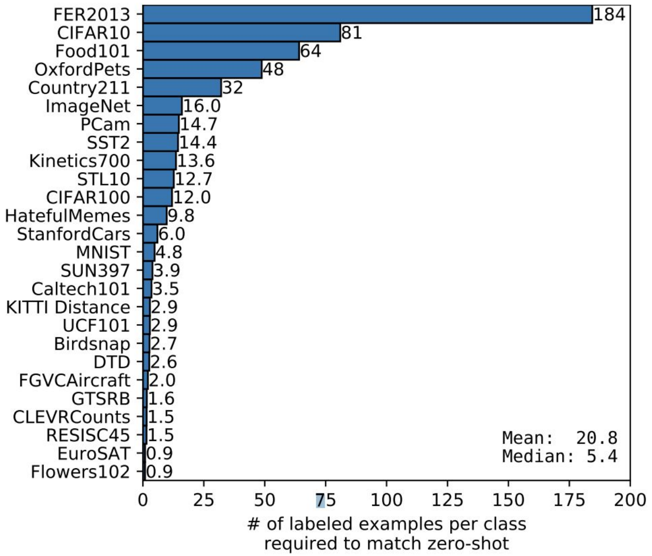


Figure 7. **The data efficiency of zero-shot transfer varies widely.** Calculating the number of labeled examples per class a linear classifier on the same CLIP feature space requires to match the performance of the zero-shot classifier contextualizes the effectiveness of zero-shot transfer. Values are estimated based on log-linear interpolation of 1, 2, 4, 8, 16-shot and fully supervised results. Performance varies widely from still underperforming a one-shot classifier on two datasets to matching an estimated 184 labeled examples per class.

Zeroshot performance correlates with linear probe

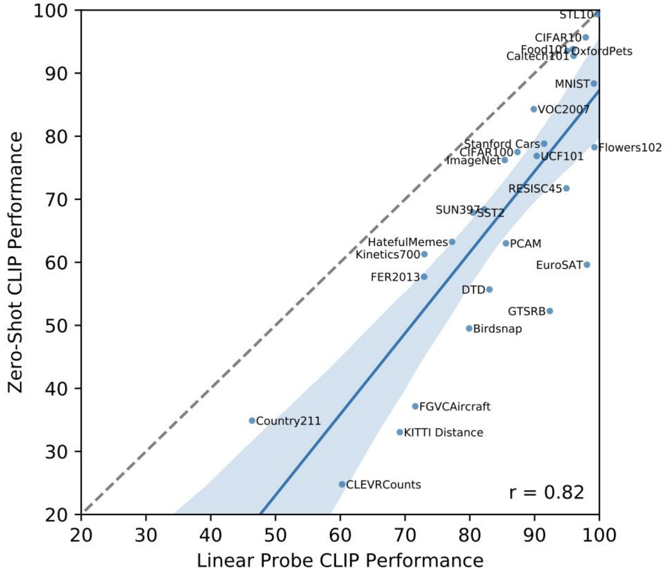
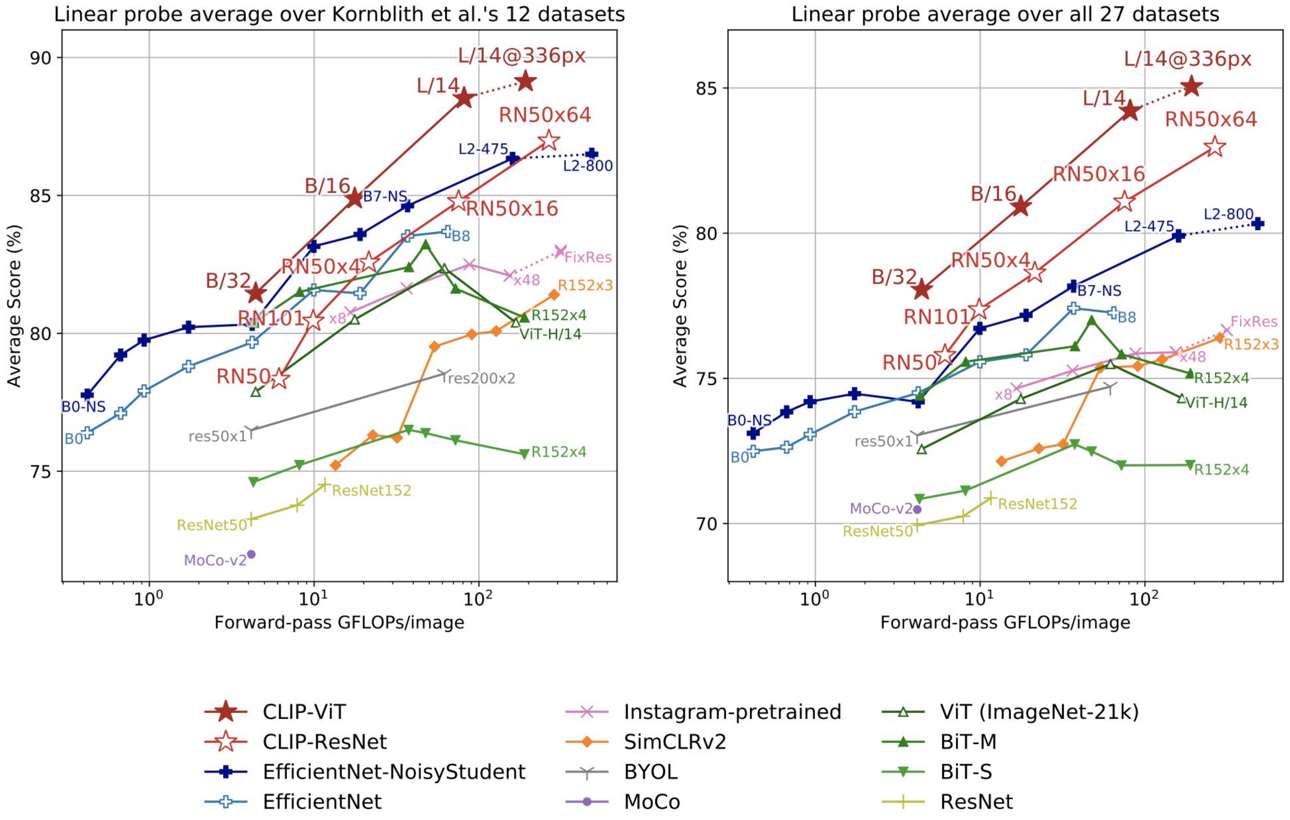


Figure 8. **Zero-shot performance is correlated with linear probe performance but still mostly sub-optimal.** Comparing zero-shot and linear probe performance across datasets shows a strong correlation with zero-shot performance mostly shifted 10 to 25 points lower. On only 5 datasets does zero-shot performance approach linear probe performance (≤3 point difference).

CLIP representations outperform SSL approaches



CLIP features vs Imagenet features

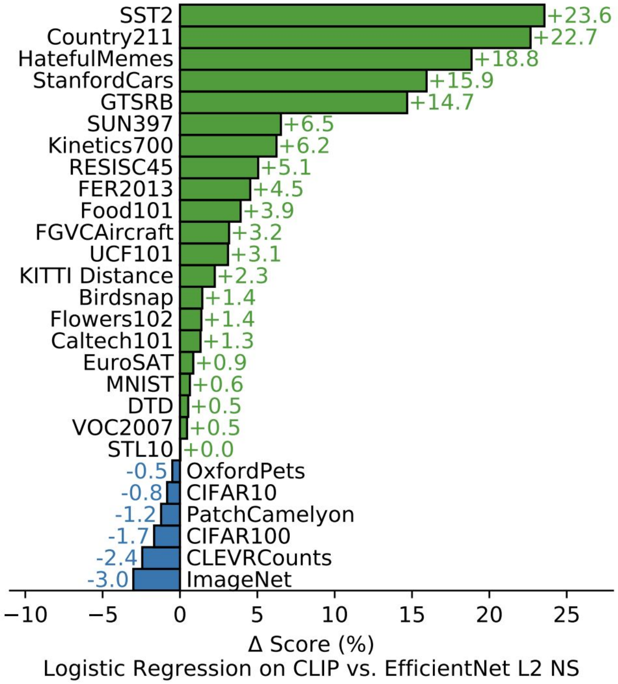
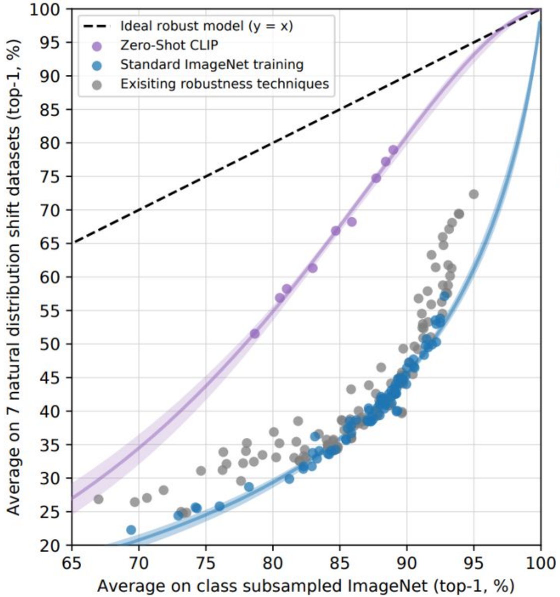
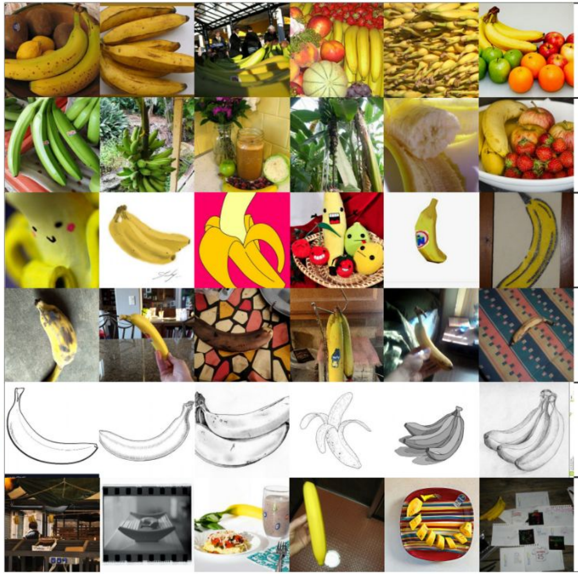


Figure 11. **CLIP's features outperform the features of the best ImageNet model on a wide variety of datasets.** Fitting a linear classifier on CLIP's features outperforms using the Noisy Student EfficientNet-L2 on 21 out of 27 datasets.

Robustness to distribution shift



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%

Comparison to human performance

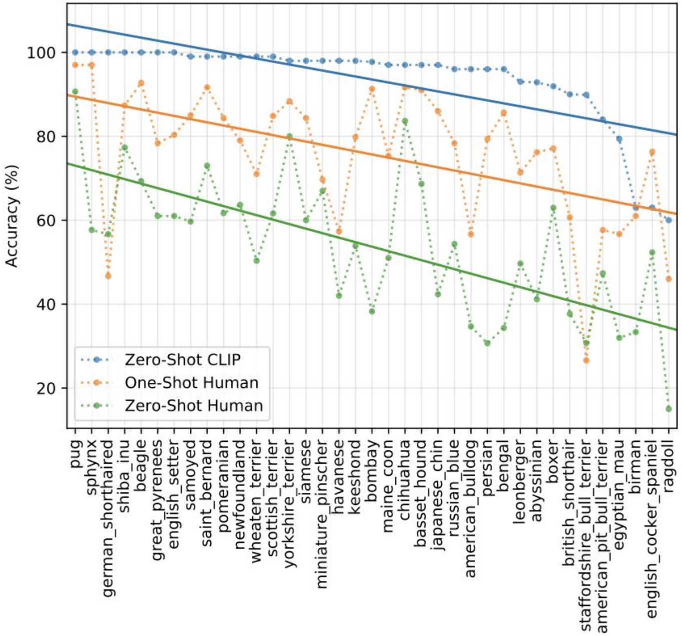


Figure 16. The hardest problems for CLIP also tend to be the hardest problems for humans. Here we rank image categories by difficulty for CLIP as measured as probability of the correct label.

Beyond this paper.

CLIP features continue to be very general across multiple topics including

- Detection
- Segmentation in Images
- Segmentation in 3D scenes
- NERFs

Limitations

- While CLIP has one model that generalizes to many datasets, it is below the SOTA performance on most datasets
- CLIP's setup is classification focused and can not directly work on other CV tasks
- While CLIP generalizes to distribution shifts, it does not generalize to datasets that are out of distribution in its pretraining e.g. MNIST, Satellite Images etc
- While CLIP can generate classifiers on the fly, it still requires on hand crafting the classification space
- CLIP is not data efficient,
- CLIP's dataset is closed source. Open source initiatives have recollected it.

Conclusion

- CLIP provides a novel paradigm to train a single model on large amount of data
- This single pretrained model can achieve competitive performance on wide variety of tasks
- CLIP features are more general than ImageNet and allow for open set learning
- Open source is awesome! CLIP models have been reproduced at https://github.com/mlfoundations/open_clip