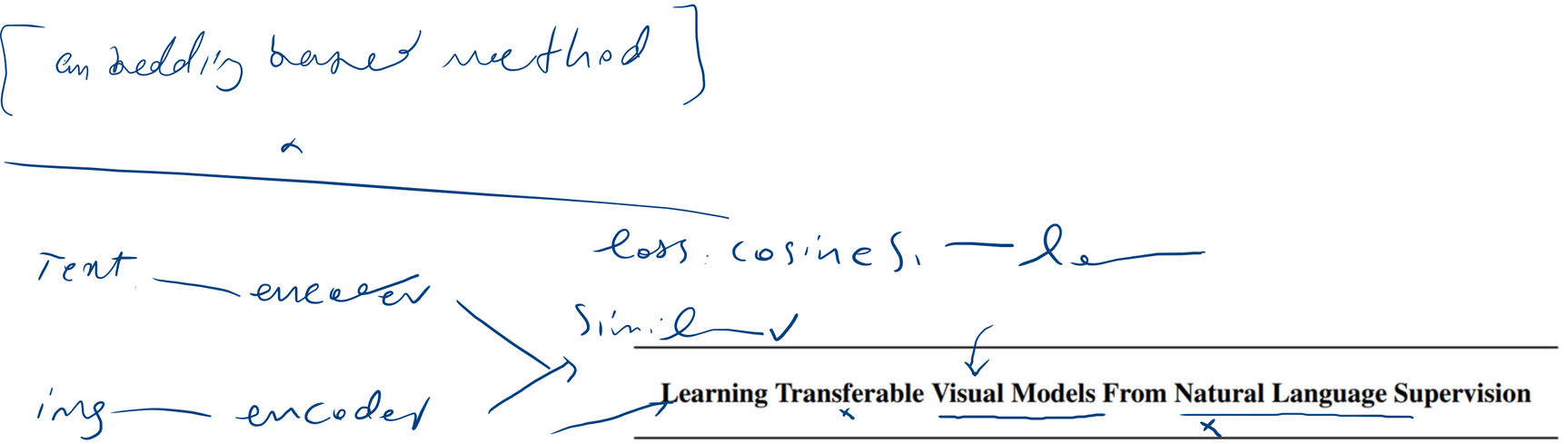
CLIP

Contrastive Language-Image Pre-training

رفرنس



Learning Transferable Visual Models From Natural Language Supervision

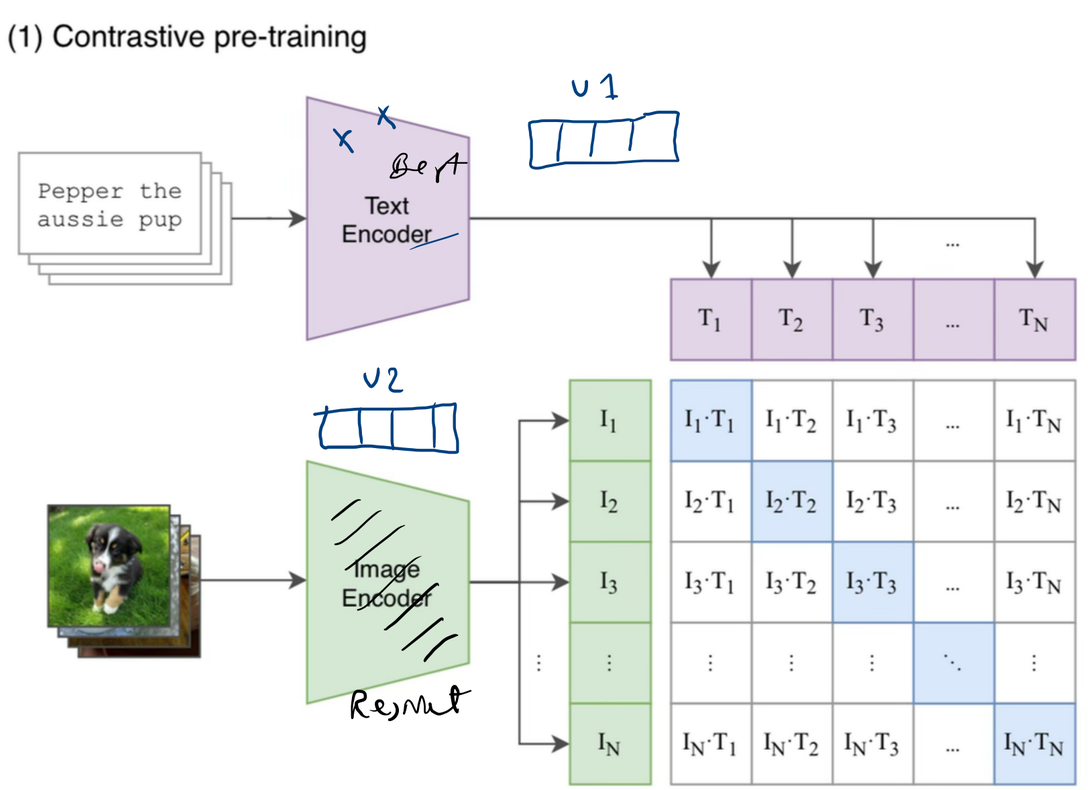
Alec Radford [*1] Jong Wook Kim [*1] Chris Hallacy [1] Aditya Ramesh [1] Gabriel Goh [1] Sandhini Agarwal [1]
Girish Sastry [1] Amanda Askell [1] Pamela Mishkin [1] Jack Clark [1] Gretchen Krueger [1] Ilya Sutskever [1]

[*]Equal contribution [1]OpenAI, San Francisco, CA 94110, USA.
Correspondence to: <{alec, jongwook}@openai.com>.

https://github.com/OpenAI/CLIP

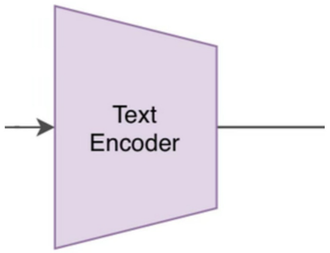
Feb 26, 2021

معماری CLIP (در مرحله آموزش)



مدل های مورد استفاده برای Text Encoder

LSTM, GRU, RNN, ...



۲

The text encoder is a Transformer (Vaswani et al., 2017) with the architecture modifications described in Radford et al. (2019). As a base size we use a 63M-parameter 12-layer 512-wide model with 8 attention heads. The trans-

Transformer

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

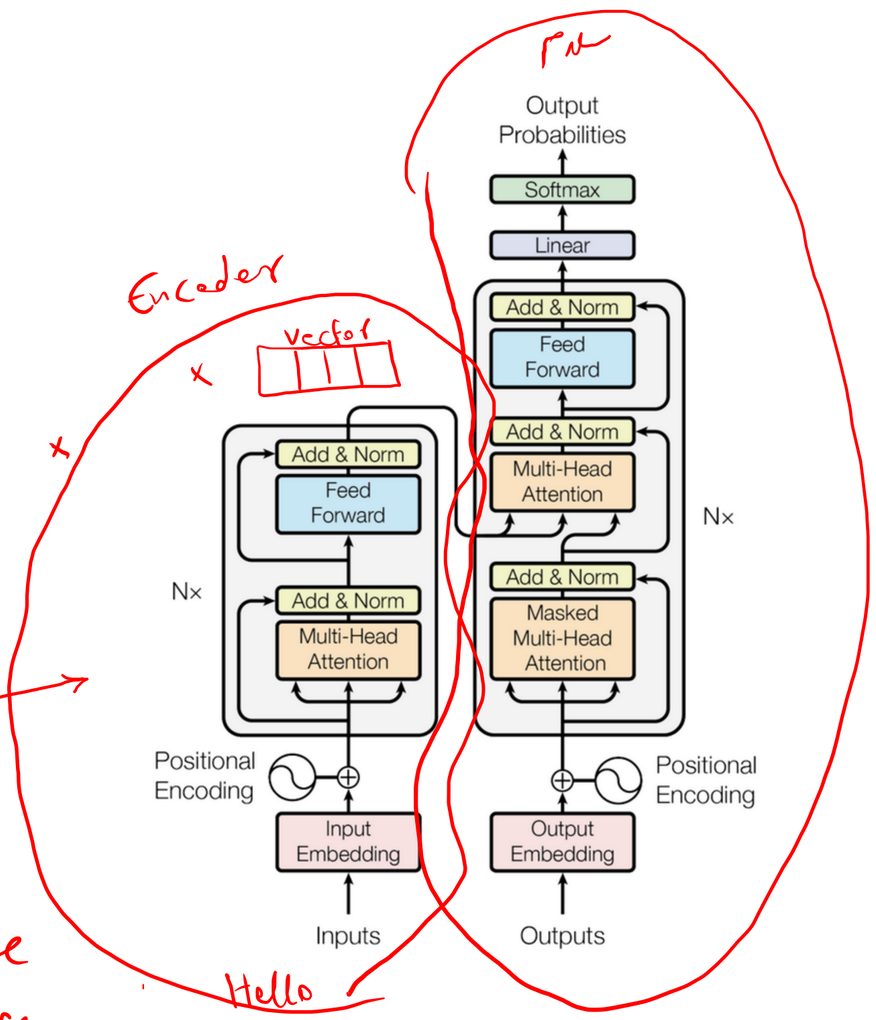
معماری مدل Transformer

BERT

Encoder

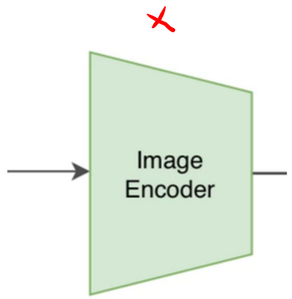
GPT

Decoder



مدل های مورد استفاده برای Image Encoder

Resnet 50
Resnet-101
Resnet 50x40

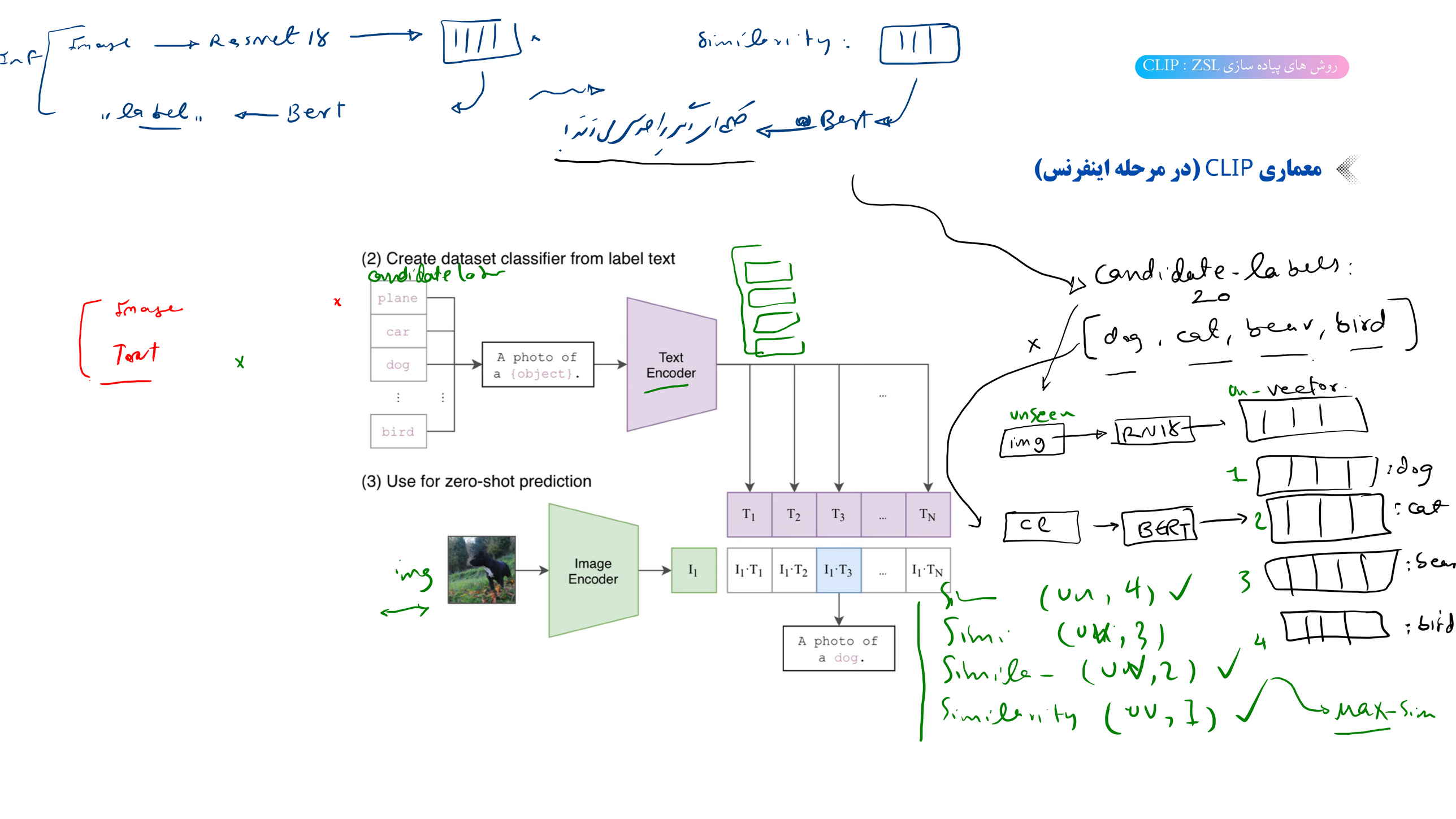


We train a series of 5 ResNets and 3 Vision Transformers. For the ResNets we train a ResNet-50, a ResNet-101, and then 3 more which follow EfficientNet-style model scaling and use approximately 4x, 16x, and 64x the compute of a ResNet-50. They are denoted as RN50x4, RN50x16, and RN50x64 respectively. For the Vision Transformers we train a ViT-B/32, a ViT-B/16, and a ViT-L/14. We train all models for 32 epochs. We use the Adam optimizer (Kingma & Ba, 2014) with decoupled weight decay regularization (Loshchilov & Hutter, 2017) applied to all weights that are not gains or biases, and decay the learning rate using a cosine schedule (Loshchilov & Hutter, 2016). Initial hyper-

- ResNet-50
- ResNet-101
- ResNet-50x4
- ResNet-50x16
- ResNet-50x64

- ViT-B/32
- ViT-B/16
- ViT-B/14

معماری CLIP (در مرحله اینفرنس)



مقایسه ی مدل های Image Encoder

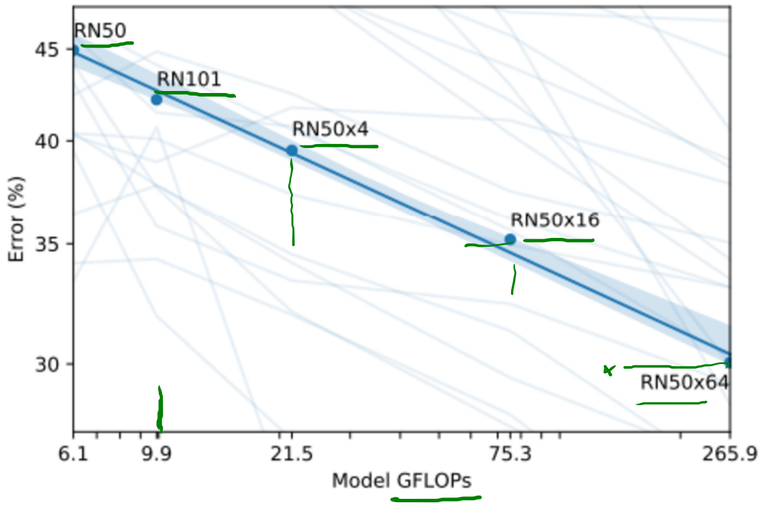


Figure 9. **Zero-shot CLIP performance scales smoothly as a function of model compute.** Across 39 evals on 36 different datasets, average zero-shot error is well modeled by a log-log linear trend across a 44x range of compute spanning 5 different CLIP models. Lightly shaded lines are performance on individual evals, showing that performance is much more varied despite the smooth overall trend.

شبه کد CLIP

```
# 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.

نتایج و مقایسه ی مدل

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%

نتایج و مقایسه ی مدل

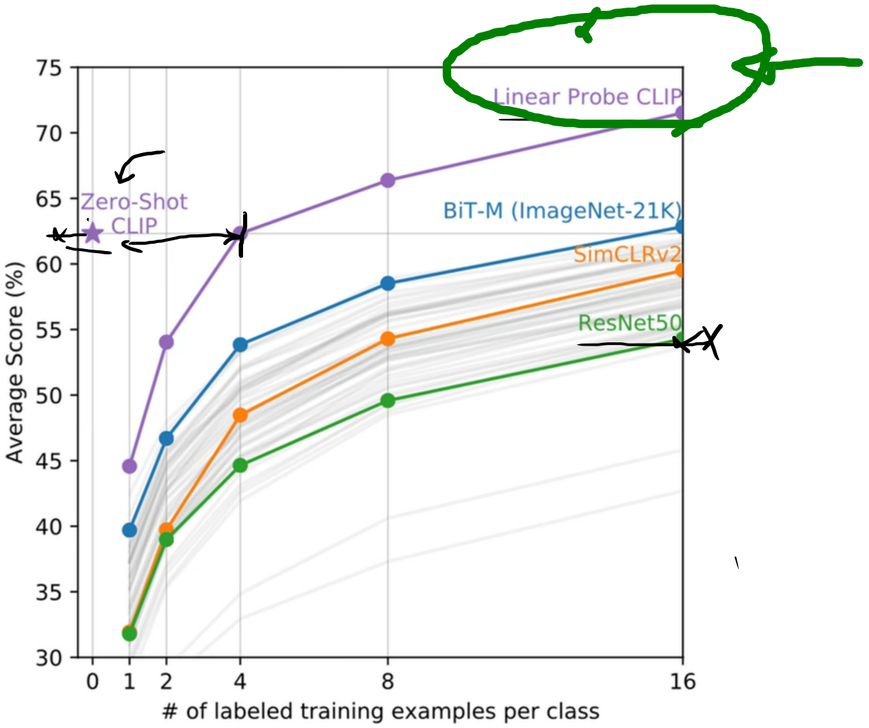
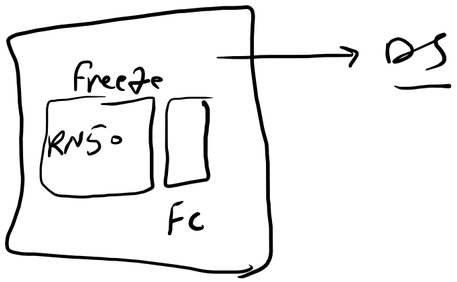
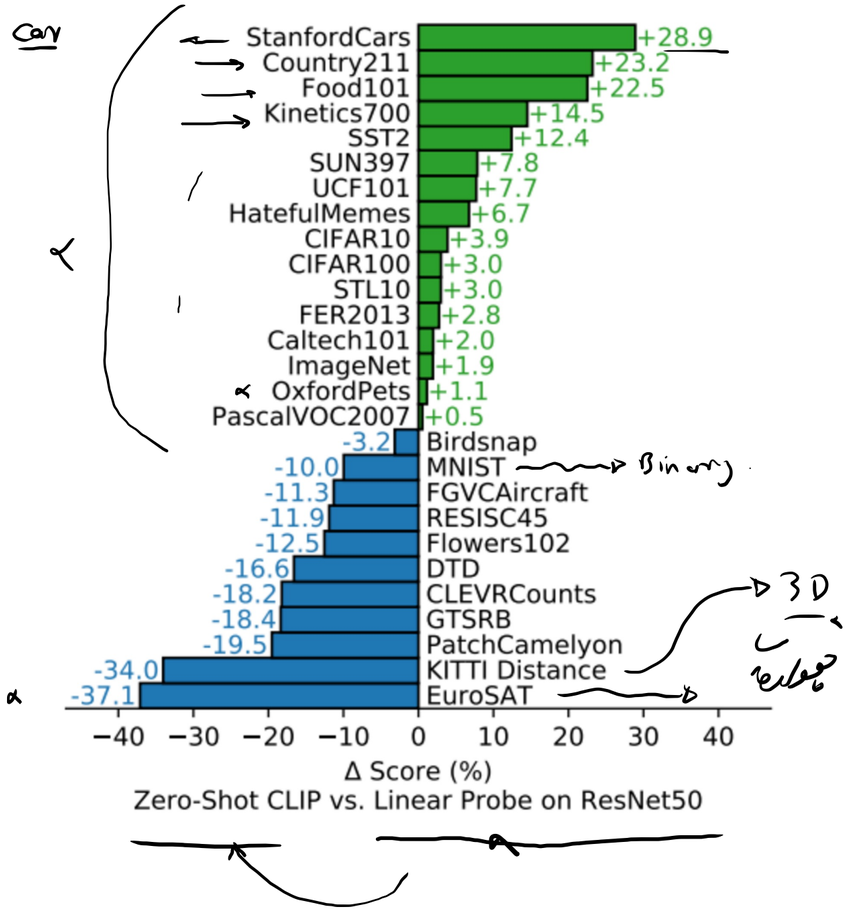


Figure 6. **Zero-shot CLIP outperforms few-shot linear probes.** Zero-shot CLIP matches the average performance of a 4-shot linear classifier trained on the same feature space and nearly matches the best results of a 16-shot linear classifier across publicly available models. For both BiT-M and SimCLRv2, 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.

نتایج و مقایسه ی مدل





نتایج و مقایسه ی مدل

	Accuracy	Majority Vote on Full Dataset	Accuracy on Guesses	Majority Vote Accuracy on Guesses
Zero-shot human	53.7	57.0	69.7	63.9
Zero-shot CLIP	**93.5**	**93.5**	**93.5**	**93.5**
One-shot human	75.7	80.3	78.5	81.2
Two-shot human	75.7	85.0	79.2	86.1

Table 2. Comparison of human performance on Oxford IIT Pets. As in Parkhi et al. (2012), the metric is average per-class classification accuracy. Most of the gain in performance when going from the human zero shot case to the human one shot case is on images that participants were highly uncertain on. "Guesses" refers to restricting the dataset to where participants selected an answer other than "I don't know", the "majority vote" is taking the most frequent (exclusive of ties) answer per image.

برخی نکات جالب

Resource

(۱)

their local batch of embeddings. 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. For the ViT-L/14 we also pre-train at a higher 336

Data

(۲)

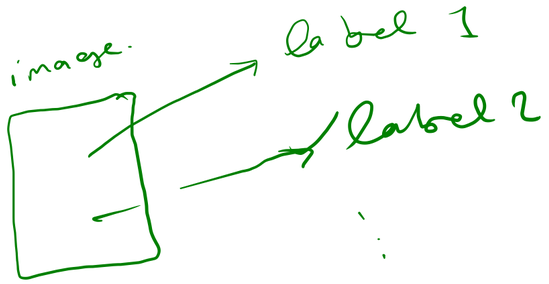
derestimate the potential of this line of research. To address this, we constructed a new dataset of 400 million (image, text) pairs collected form a variety of publicly available sources on the Internet. To attempt to cover as broad a set

بریم سراغ کدش!

CLIP

ment

sets.



The End