CLIP: Connecting text and images

EECS 598

Jiachen Liu

2024/1

SymbioticLab

UNIVERSITY OF MICHIGAN

Agenda

1. What is CLIP?
2. How does CLIP work?
3. Why CLIP matters?

"CLIP (Contrastive Language-Image Pre-Training) is a neural network trained on a variety of **(image, text) pairs**.

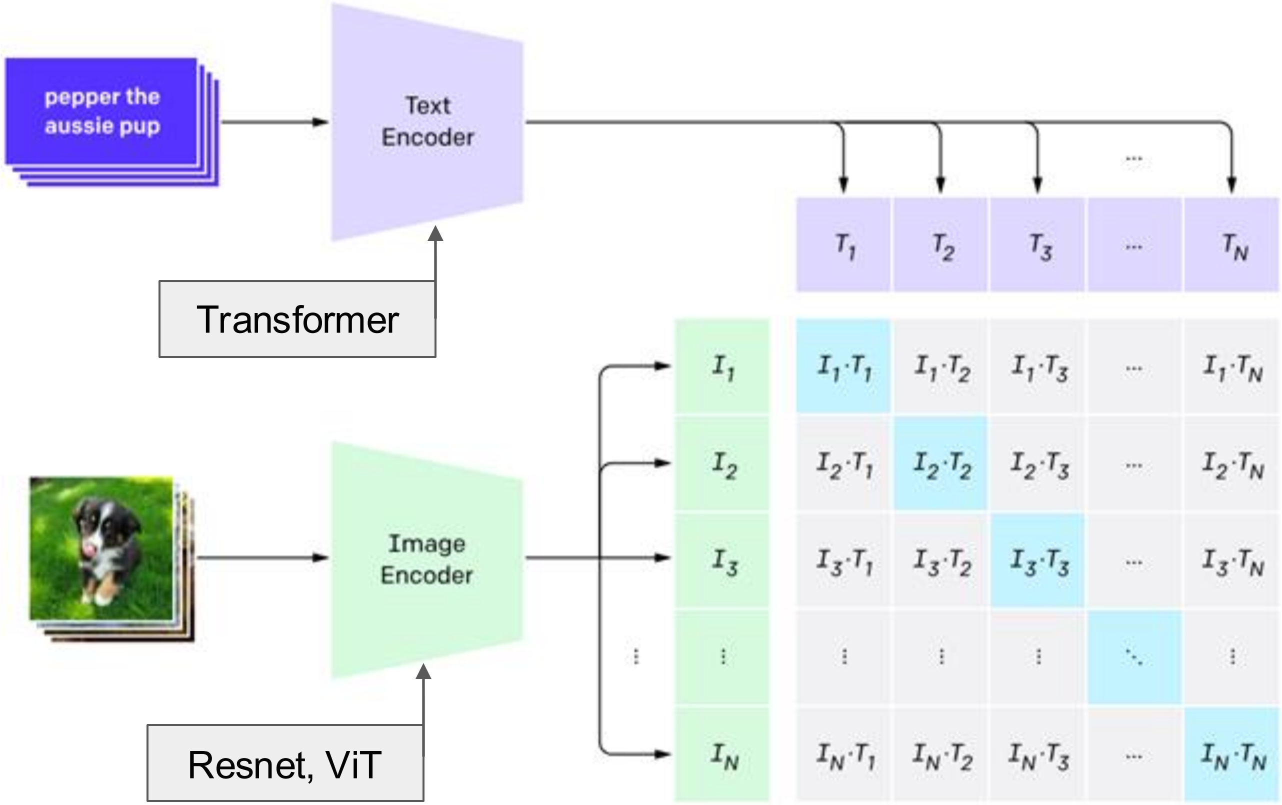
It can be instructed in natural language to **predict the most relevant text snippet** given an image, without directly optimizing for the task, similarly to the **zero-shot** capabilities of GPT-2 and 3."

https://github.com/openai/CLIP

What is CLIP?

1. **Open Source:** The model is created and open-sourced by OpenAI.
2. **Multimodal: CLIP** combines Natural Language Processing and Computer Vision.
3. **Contrastive Learning:** CLIP is trained on a huge **dataset of 400 million (image, text) pairs** collected from the internet
   a. With Contrastive Learning, CLIP is trained to learn that similar text-image should be close in the latent space, while dissimilar ones should be far apart.
4. **Zero-shot learning** enables the generalization of unseen labels, without having explicitly trained to classify them.
   a. For example, all ImageNet models are trained to recognize 1000 specific classes. CLIP is not bound by this limitation.

How does Contrastive Language-Image Pre-Training Work?



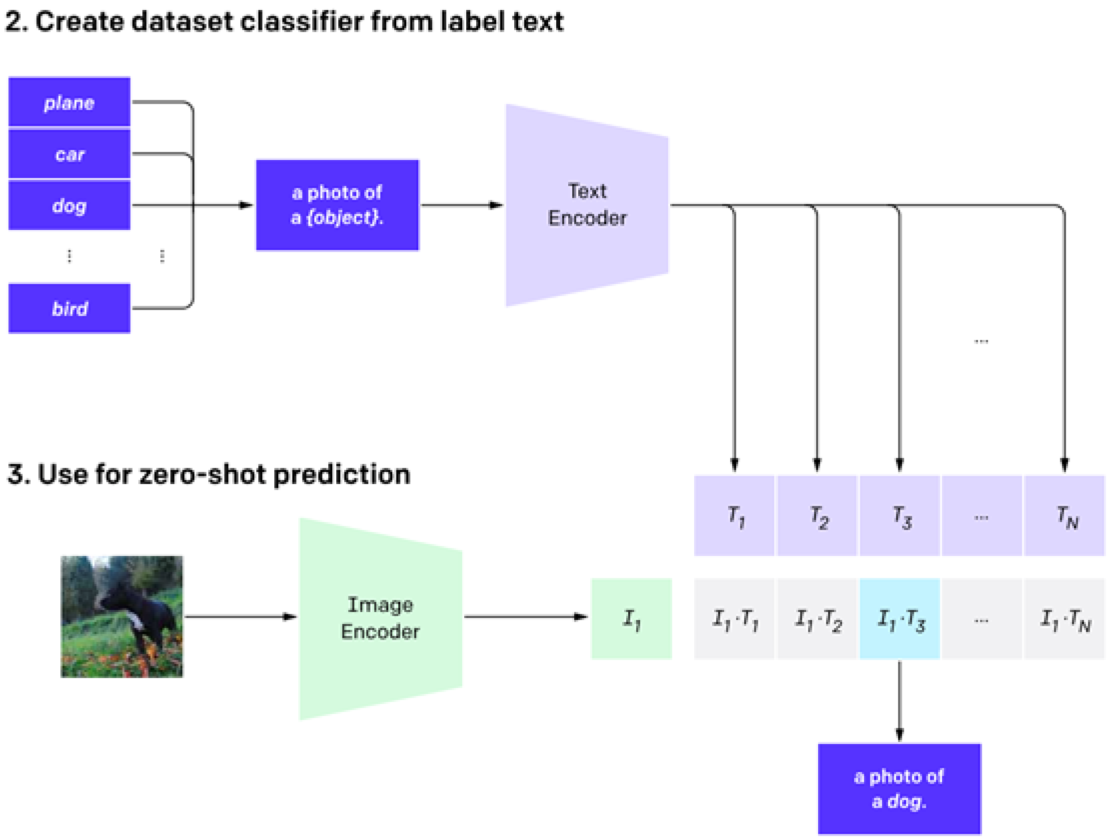
https://openai.com/research/clip

How does Contrastive Language-Image Pre-Training Work?

```
# 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
```

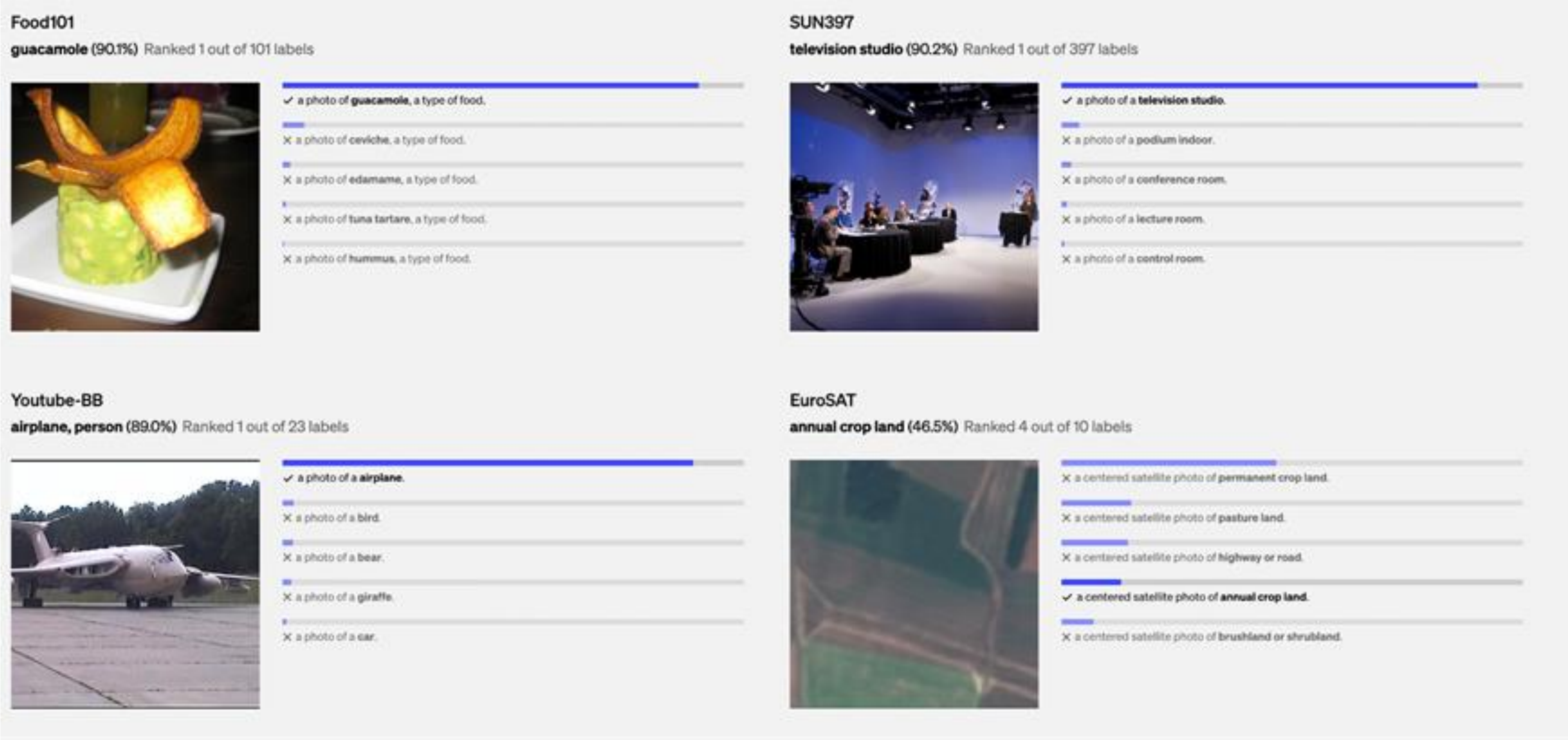

Learning Transferable Visual Models From Natural Language Supervision

How does CLIP Predict?



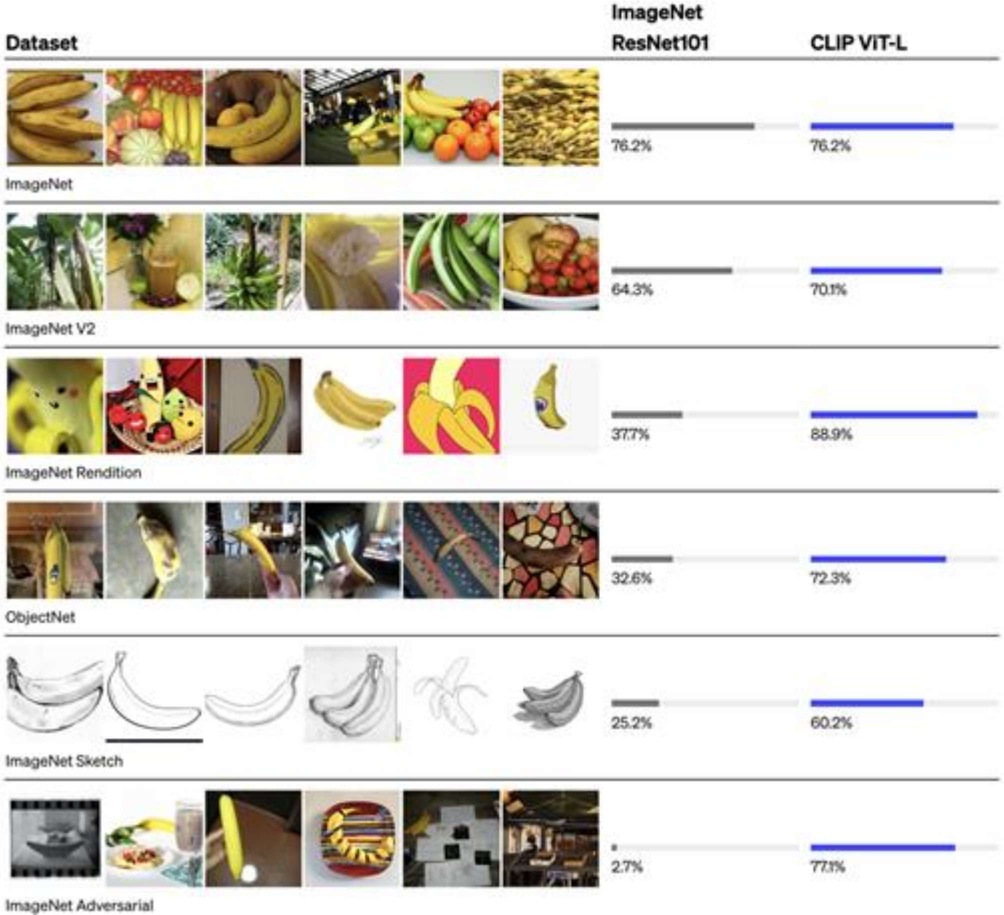
https://openai.com/research/clip

Results of CLIP



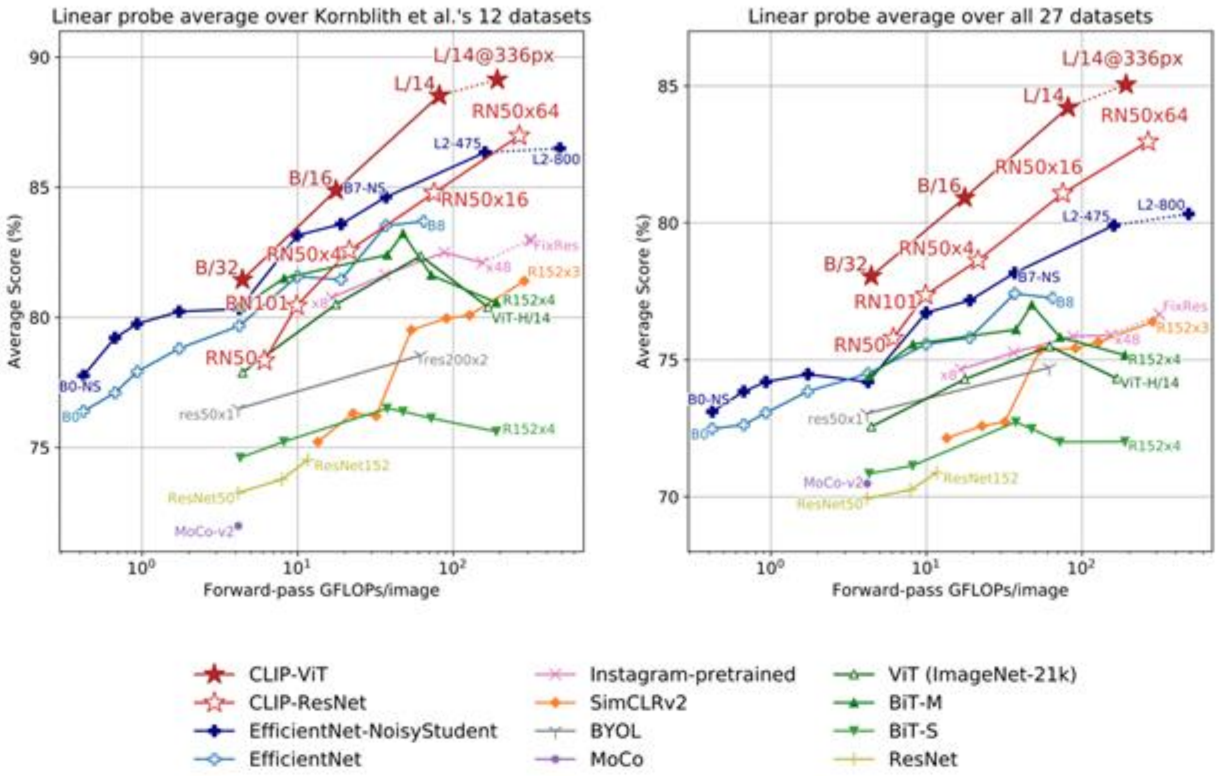
https://openai.com/research/clip

Results of CLIP



https://openai.com/research/clip

Experiment Results of CLIP



Recap: Why does CLIP Matter?

1. Bridge Visual and Textual Understanding
2. Zero-Shot Learning Capabilities
3. Robustness Against Adversarial Attacks
4. Training data efficiency

Training Resources for CLIP

1. Training data size: 400,000,000 image-text pairs.

2. Training time: 30 GPU days across 592 V100 GPUs.

3. Training cost: $1,000,000 on AWS on-demand instances

Challenges and Limitation of CLIP

1. Computation-Intensive
2. Struggles on more abstract or systematic tasks
    a. Count the number of objects in an image
    b. Predict how close the nearest car is in a photo.
3. CLIP also still has poor generalization to images not covered in its pre-training dataset.
4. Sensitive to wording or phrasing

https://openai.com/research/clip

Follow-up Works of CLIP

1. Object detection
2. Image segmentation
3. Motion detection
4. Video/Image search
5. Multimodality
6. Image generation
7. …

Q&A