Learning Transferable Visual Models From Natural Language Supervision

Presenter: Dian Jiao

Learning Transferable Visual Models From Natural Language Supervision

CLIP: Contrastive Language–Image Pre-training

- Powerful transfer learning capabilities (Zero-shot)
- A visual model that associates images with related textual descriptions
- Supervised by using natural language
- Contrastive Learning

Contrastive Language-Image Pre-training

Image Encoder:

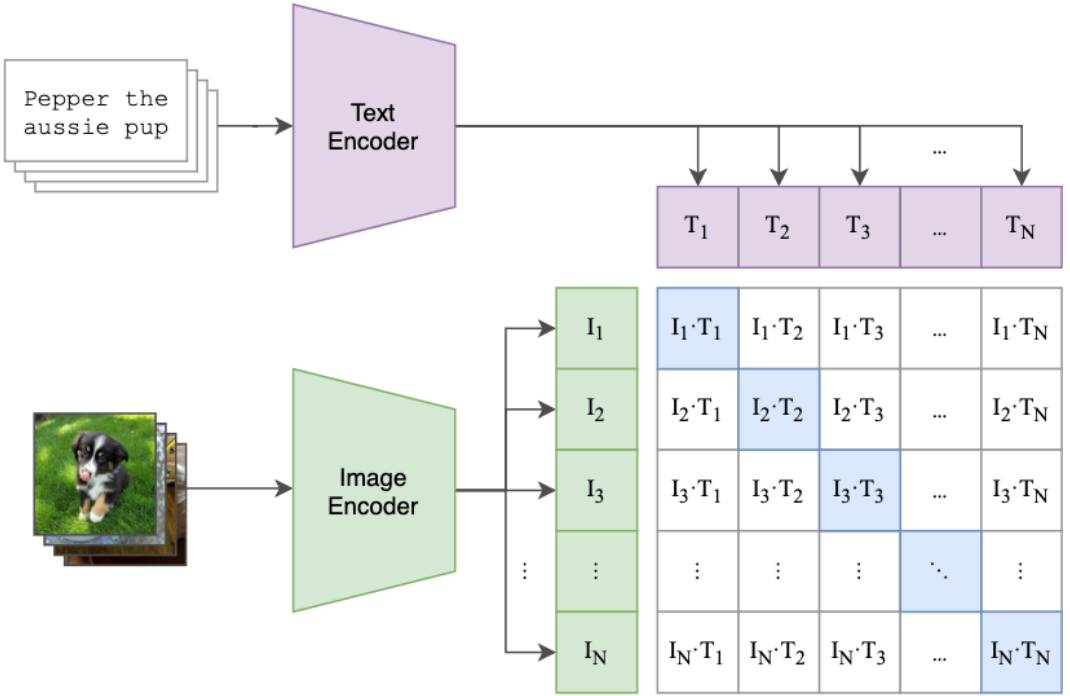
ResNet-based or ViT-based model

Text Encoder:

Transformer-based model

Input:

Images & Textual Descriptions



Pseudocode of Training

Projection layer:

- The features extracted by the image and text encoders are mapped to the multimodal space by matrix multiplication

Positive and negative samples:

- Each element of matrix n is the inner product similarity of an image vector and a text vector
- Elements on the diagonal of n are positive samples, and the rest of the elements are negative samples

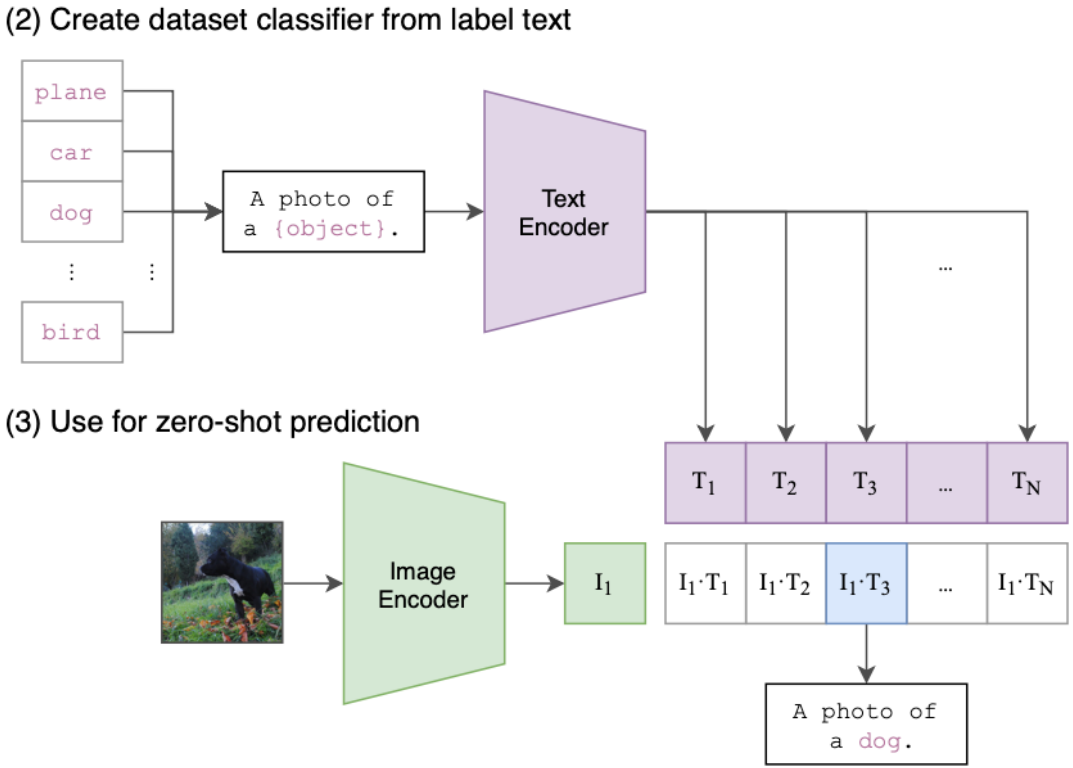
Loss: average cross entropy loss

```
# 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
```

Zero-shot image classification

Input: Images & Textual Prompts

Output: top-ranked prompt in similarity



Prediction examples

Food101

guacamole (90.1%) Ranked 1 out of 101 labels

- ✓ a photo of **guacamole**, a type of food.
- ✗ a photo of **ceviche**, a type of food.
- ✗ a photo of **edamame**, a type of food.
- ✗ a photo of **tuna tartare**, a type of food.
- ✗ a photo of **hummus**, a type of food.

SUN397

television studio (90.2%) Ranked 1 out of 397 labels

- ✓ a photo of a **television studio**.
- ✗ a photo of a **podium indoor**.
- ✗ a photo of a **conference room**.
- ✗ a photo of a **lecture room**.
- ✗ a photo of a **control room**.

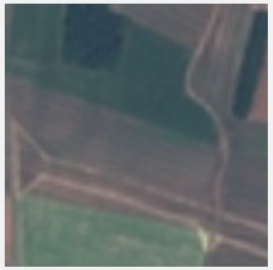
Youtube-BB

airplane, person (89.0%) Ranked 1 out of 23 labels

- ✓ a photo of a **airplane**.
- ✗ a photo of a **bird**.
- ✗ a photo of a **bear**.
- ✗ a photo of a **giraffe**.
- ✗ a photo of a **car**.

EuroSAT

annual crop land (46.5%) Ranked 4 out of 10 labels



- ✗ a centered satellite photo of **permanent crop land**.
- ✗ a centered satellite photo of **pasture land**.
- ✗ a centered satellite photo of **highway or road**.
- ✓ a centered satellite photo of **annual crop land**.
- ✗ a centered satellite photo of **brushland or shrubland**.

Natural Language Supervision

- Scalability: Easier to scale compared to traditional crowd-sourced labeling. No need for 1-of-N "gold label".
- Rich Source: Exploits abundant supervision present in internet text.
- Zero-Shot Transfer: Connects visual representation to language for flexible application without the need for retraining.
- Beyond Representation: Not just learning visual representation, connecting it to language enables zero-shot transfer.

Sufficiently Large Dataset

Existing Datasets:

MS-COCO & Visual Genome

- High-quality
- ~100,000 training photos

YFCC100M

- Large but varied metadata quality
- Initial: 100 million photos
- After filtering: 15 million photos

Creating WIT (WebImageText):

- Source: Various public internet sources
- Size: 400 million (image, text) pairs
- Construction: Image-text pairs from 500,000 queries
- Class-balanced: Up to 20,000 pairs per query

Similar word count to WebText (used for GPT-2)Aims to harness richer, more diverse visual-linguistic knowledge

Selecting an Efficient Pre-Training Method

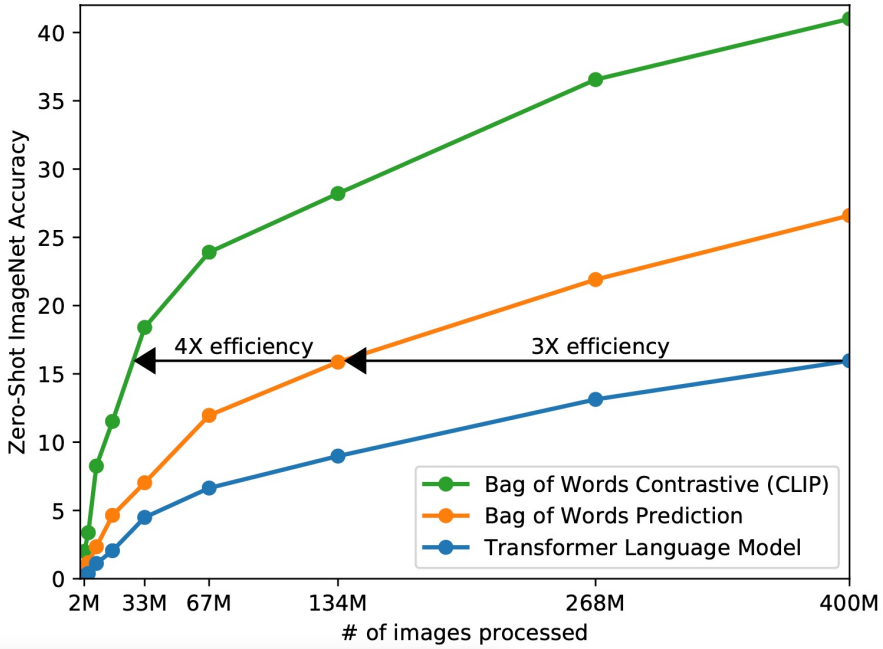
Why contrastive learning?

Initial Approach

- Jointly train an image CNN and text transformer from scratch to predict image captions
- 3x slower than a simpler baseline that predicts a bag-of-words encoding of the same text

Contrastive Learning

- Focus shifted to predicting which text pairs with which image, without exact words.
- Swapping to a contrastive objective led to a 4x efficiency improvement in zero-shot transfer to ImageNet.



CLIP is much more efficient at zero-shot transfer than our image caption baseline

Selecting an Efficient Pre-Training Method

Overfitting isn't a concern due to the large dataset.

Settings:

- Trained from scratch.
- No initializing of encoders with pretrained weights.
- Linear projection used instead of non-linear for the multi-modal embedding.
- Removed the text transformation function; many image-text pairs were single sentences.
- Only data augmentation: random square crop.
- Temperature parameter optimized during training, eliminating hyper-parameter tuning.

Model Choosing and Training

Text Encoder: Transformer

Image Encoder: 5 ResNets, 3 Vision Transformers

32 epochs for all models

Batch size: 32,768

Many techniques were applied to reduce memorize use

Best Model:

ViT-L/14@336px

12 days on 256 V100 GPUs

Pre-trained at 336-pixel resolution for an additional epoch (FixRes).

Prompt Engineering and Ensembling

Why prompt engineering?

- Polysemy: Labels with multiple meanings (e.g., "crane" in ImageNet).
- Context: Label ambiguity due to lack of context (e.g., "boxer" in Oxford-IIIT Pet dataset).

How to do prompt engineering?

Default Prompt: "A photo of a {label}."

- Improves ImageNet accuracy by 1.3%.

Customizing prompts:

- Oxford-IIIT Pets : "A photo of a {label}, a type of pet."
- Satellite image classification datasets: "a satellite photo of a {label}."

Prompt Engineering and Ensembling

Ensemble over different context prompts:

- "A photo of a big {label}."
- "A photo of a small {label}."

Embedding space ensembling:

- Cache a single set of averaged text embeddings.
- Same compute cost as single classifier when averaged over many predictions.

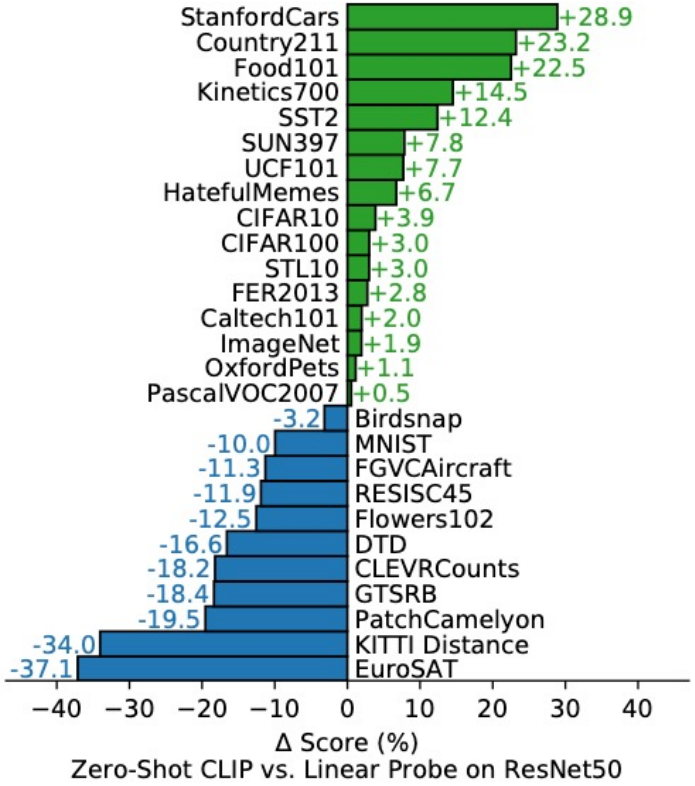
Results:

- Improved performance across datasets.
- ImageNet: 80 different context prompts improve performance by an extra 3.5%.

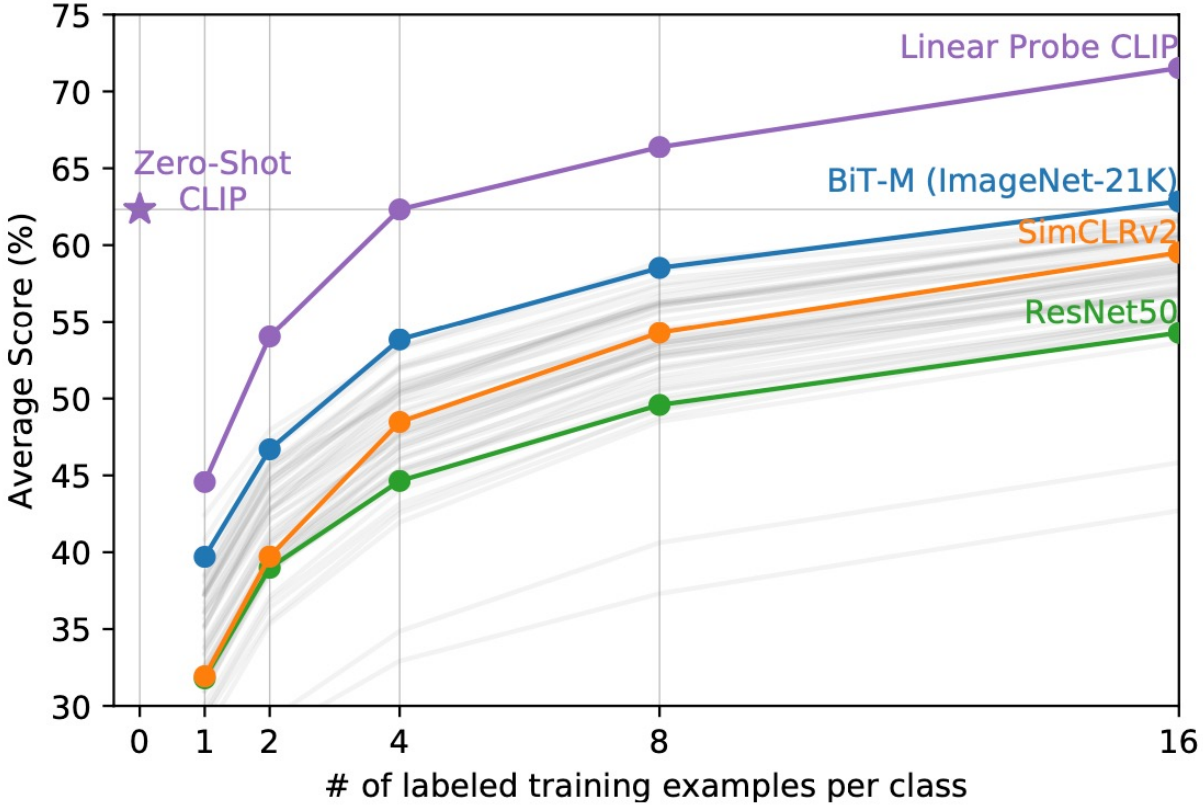
```
imagenet_templates = [
    'a bad photo of a {}.',
    'a photo of many {}.',
    'a sculpture of a {}.',
    'a photo of the hard to see {}.',
    'a low resolution photo of the {}.',
    'a rendering of a {}.',
    'graffiti of a {}.',
    'a bad photo of the {}.',
    'a cropped photo of the {}.',
    'a tattoo of a {}.',
    'the embroidered {}.',
    'a photo of a hard to see {}.',
    'a bright photo of a {}.',
    'a photo of a clean {}.',
    'a photo of a dirty {}.',
    'a dark photo of the {}.',
    'a drawing of a {}.',
    'a photo of my {}.',
    ...
    'a tattoo of the {}.',
]
```

Zero-Shot Transfer Performance

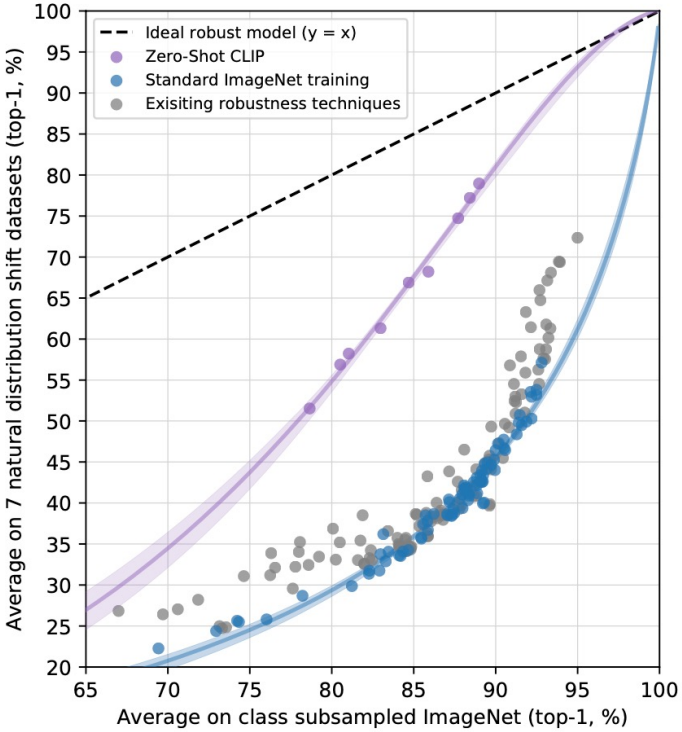
- Zero-shot CLIP is competitive with a fully supervised baseline.
- Evaluate on 27 dataset
- Good results for dataset that easy to prompt engineering and ensembling



Few-Shot Transfer Performance



Zero-Shot 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%

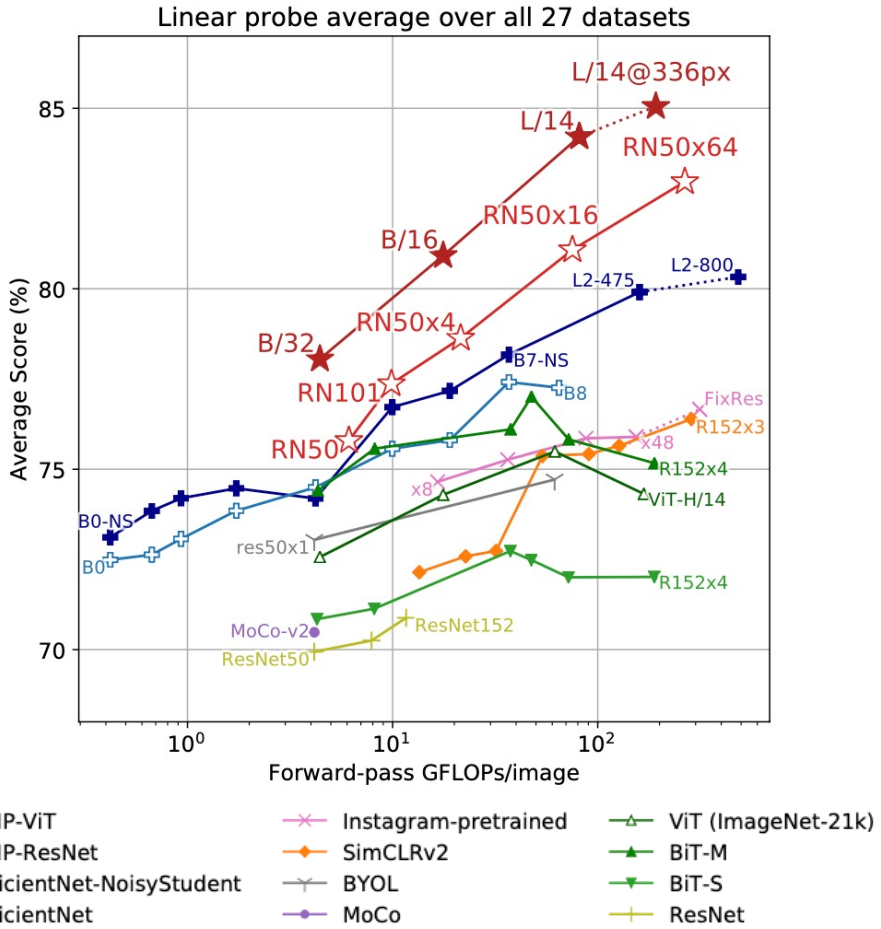
Representation Learning

Two Approaches:

- Linear classifier on extracted representations: limited flexibility, highlights failures, and provides clear feedback
- End-to-end fine-tuning: flexible but can mask failures to learn general representations

For CLIP:

- linear classifier approach mirrors zero-shot classifiers, aiding comparison and analysis

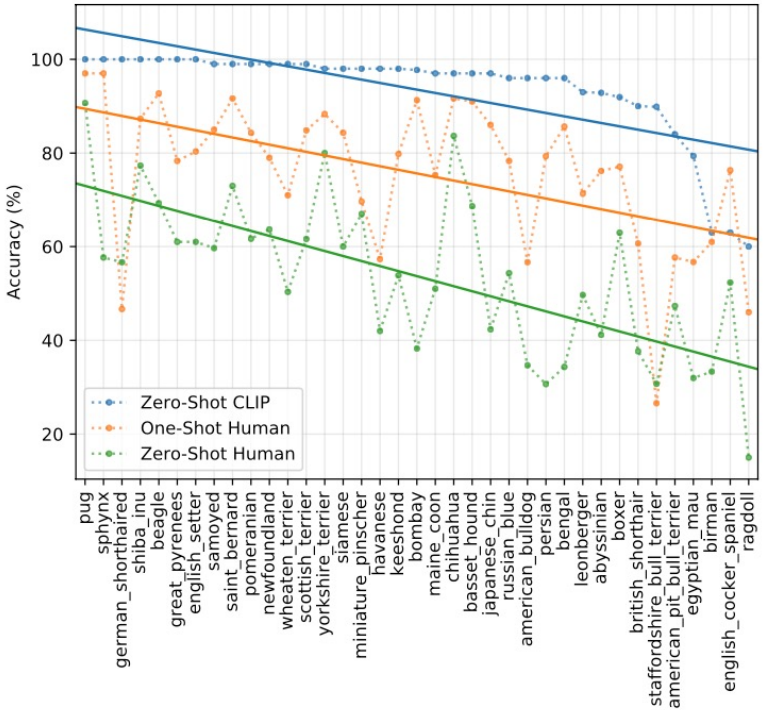


Linear probe performance of CLIP models in comparison with SOTA computer vision models

Compare with human

	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

The hardest problems for CLIP also tend to be the hardest problems for humans.



Interesting Application

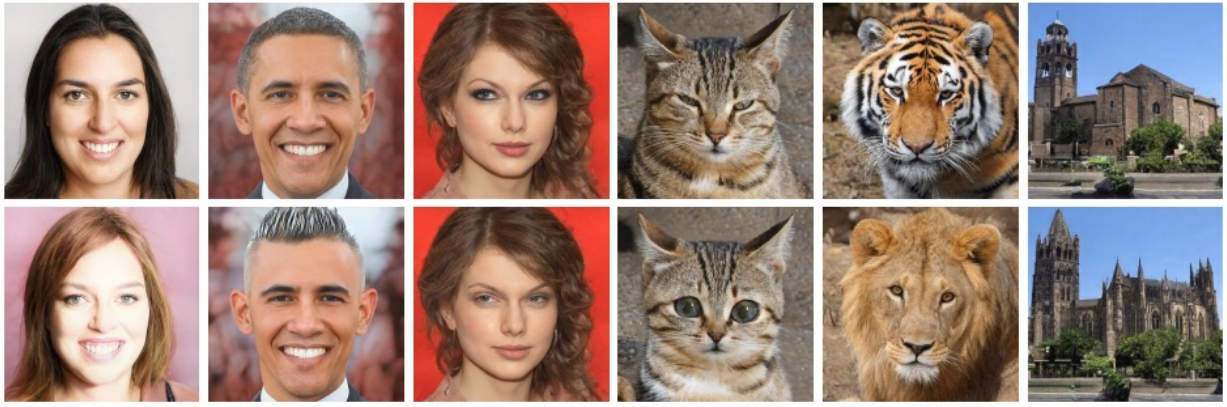
The text prompt used to drive each manipulation appears under each column.

Top row: input images

Bottom row: results

StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery

Or Patashnik[†*] Zongze Wu[‡*] Eli Shechtman[§] Daniel Cohen-Or[†] Dani Lischinski[‡]
[‡]Hebrew University of Jerusalem [†]Tel-Aviv University [§]Adobe Research



Interesting Application

- Synthesizes images from text by gradient descent over RGBA Bezier curves, minimizing cosine distance between the CLIP encodings of generated images and description prompts.

- not require learning a new model.

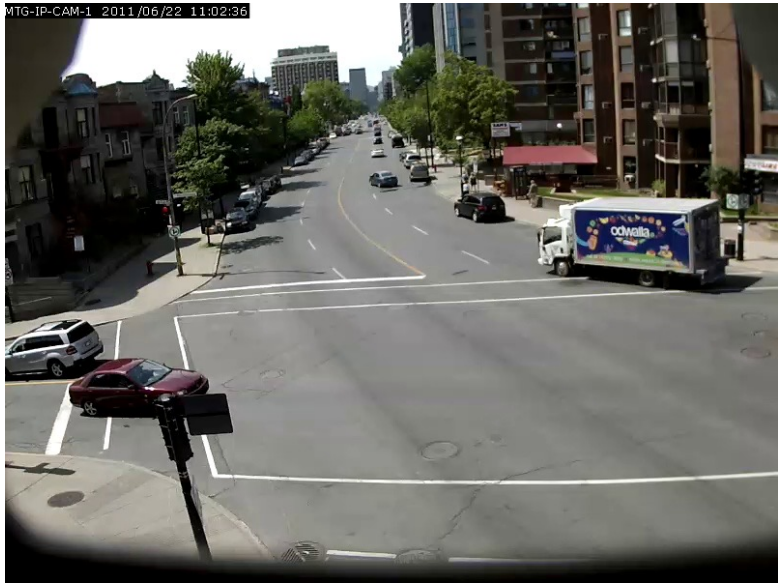
CLIPDraw: Exploring Text-to-Drawing Synthesis through Language-Image Encoders

Kevin Frans[1,2], **L. B. Soros**[1] **and Olaf Witkowski**[1,3,4]
[1]Cross Labs, Cross Compass Ltd., Tokyo, Japan
[2]Massachusetts Institute of Technology, Cambridge, MA, USA
[3]Earth-Life Science Institute, Tokyo Institute of Technology, Japan
[4]College of Arts and Sciences, University of Tokyo, Japan
kvfrans@csail.mit.edu

"A drawing of a cat". "Horse eating a cupcake". "A 3D rendering of a temple". "Family vacation to Walt Disney World". "Self".

Interesting Application

Video Retrieval: Contrastive Language-Image Forensic Search



A truck with the text "odwalla”

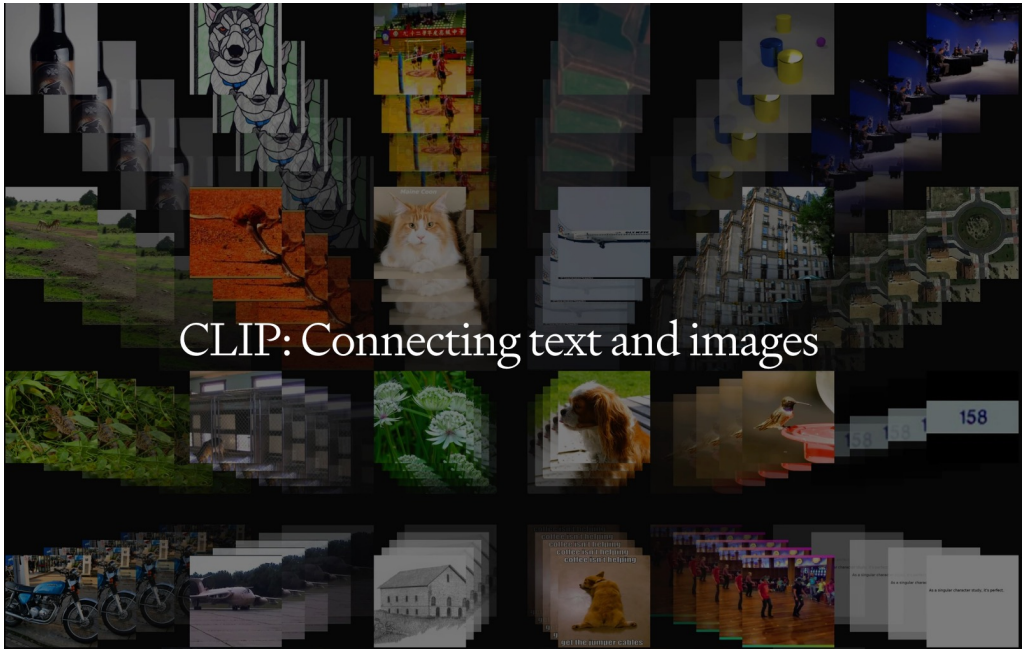
A white BMW car

Limitations

- Zero-shot performance is well below the SOTA
- Especially weak on abstract tasks such as counting
- Poor on out-of-distribution data such as MNIST
- Susceptible to adversarial attacks
- Dataset selection in the eval suite, use of large validation sets for prompt engineering
- Social biases

Summary

- Intuitive and simple ideas
- Efficient implementation
- Excellent performance
- Generalized and robust



Quiz 1

How does the CLIP model primarily learn to associate images with text during its training (a deep learning technique)?

Quiz 1

How does the CLIP model primarily learn to associate images with text during its training (a deep learning technique)?

Contrastive Learning

Quiz 2

In the CLIP model, which operation is used to compute the similarity between image and text embeddings by yielding a scalar value based on their corresponding components?

Quiz 2

In the CLIP model, which operation is used to compute the similarity between image and text embeddings by yielding a scalar value based on their corresponding components?

Dot Product