OpenAI's CLIP

Surya || 10th Feb 2021

Some history…

- Supervised Learning (ResNet models)
  - ImageNet
    - Crowdsourced images + Class labels [one out of 1000 classes]
    - 15 million samples

Images
Labels

- Semi-Supervised Learning
  - Few shot variants
  - Some (Image, Label) samples + Lots of (Image, No-label)
  - Mean Teacher, VAT, MixMatch

Images
Labels

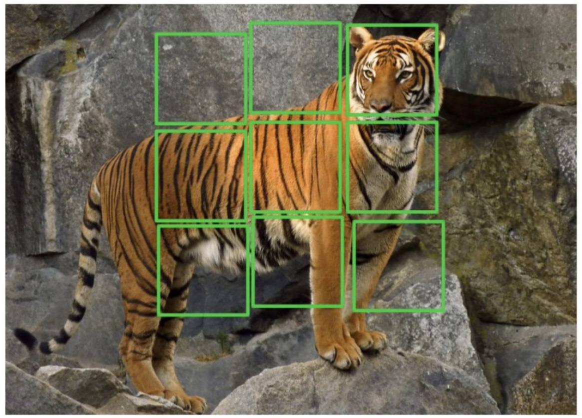
Images

- Transfer Learning
  - Use **ResNet weights** from net trained on ImageNet as encoder
  - Finetune on smaller dataset

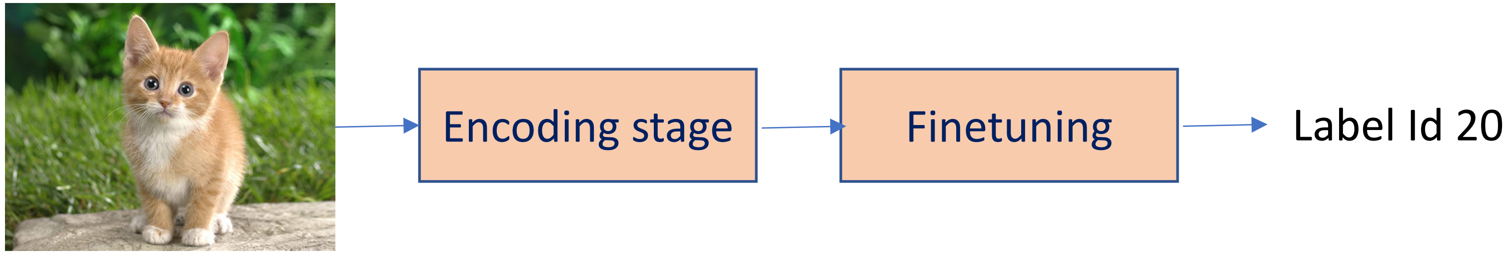
Images
Labels

Some history…

- Self-Supervised Learning
  - Inspired from text pretraining
  - Language models –
    - predict center word given context words
    - predict next character given previous character
  - Designing Pretext tasks
    - predict center pixel given surrounding pixels
    - crop images, randomize – predict the correct order
  - Get labelled data for free!
    - Caveat: quality of label from human > quality of label from jigsaw puzzles
  - CPC, CPCv2, MoCo, SimCLR, SimSiam

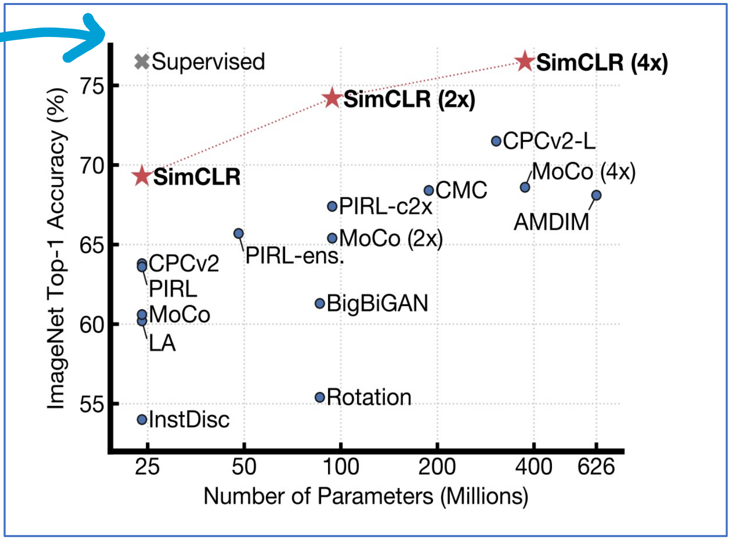


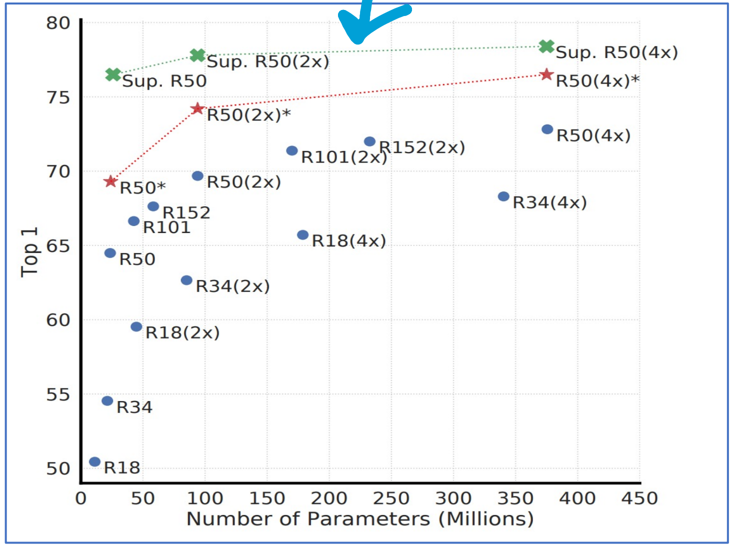
Some history…



	Train (Encoding Stage)	Finetuning stage	Testing Stage
Supervised	Data : ImageNet with labels Model : ResNet Output : ResNet Encoder	Data : Pascal data with labels Model : ResNet Encoder + RCNN classifier on top Output : Finetuned Enc + Classifier	Data : Pascal test data Model : Finetuned Enc + Classifer Output : Label
Self supervised	Data : Imagnet ***without*** labels Model : SimCLR Output : Encoder	Data : Pascal data with labels Model : SimCLR Encoder + Linear / RCNN on top Output : Finetuned Enc + Classifier	Data : Pascal test data Model : Finetuned Enc + classifier Output : Label

Some history…





Self-supervised SimCLR seems to be worse than Supervised?

- Yes, that's expected but
- No human supervision! Except linear classifier finetuning
- **Very good representations**

Self-supervised models get better with more data and compute – while supervised models stagnate

- Because nothing much to learn from just 1000 labels
- **Not scalable**

Text waiting for ImageNet moment?

- Representations learned from self-supervised clearly outperform previous techniques
  - More data-efficient [for finetuning]
- However, still not achieved the task-agnostic qualities of BERT, GPT-x representations
  - So – technically – vision is waiting for its BERT moment
- Upshot : Move away from ML-grade labelling (class labels); Build larger models

Method	Architecture	Label fraction 1% Top 5	Label fraction 10% Top 5
Supervised baseline	ResNet-50	48.4	80.4
Methods using other label-propagation:			
Pseudo-label	ResNet-50	51.6	82.4
VAT+Entropy Min.	ResNet-50	47.0	83.4
UDA (w. RandAug)	ResNet-50	-	88.5
FixMatch (w. RandAug)	ResNet-50	-	89.1
S4L (Rot+VAT+En. M.)	ResNet-50 (4×)	-	91.2
Methods using representation learning only:			
InstDisc	ResNet-50	39.2	77.4
BigBiGAN	RevNet-50 (4×)	55.2	78.8
PIRL	ResNet-50	57.2	83.8
CPC v2	ResNet-161(∗)	77.9	91.2
SimCLR (ours)	ResNet-50	75.5	87.8
SimCLR (ours)	ResNet-50 (2×)	83.0	91.2
SimCLR (ours)	ResNet-50 (4×)	**85.8**	**92.6**

Table 7. ImageNet accuracy of models trained with few labels.

Semi supervised

Contrastive Language Image Pretraining [CLIP]

Encoding stage → Finetuning → Label Id 20

	Train (Encoding Stage)	Finetuning stage	Testing Stage
Supervised	Data : ImageNet with labels Model : ResNet Output : ResNet Encoder	Data : Pascal data with labels Model : ResNet Encoder + RCNN classifier on top Output : Finetuned Enc + Classifier	Data : Pascal test data Model : Finetuned Enc + Classifer Output : Label
Self supervised	Data : Imagnet without labels Model : SimCLR Output : Encoder	Data : Pascal data with labels Model : SimCLR Encoder + Linear / RCNN on top Output : Finetuned Enc + Classifier	Data : Pascal test data Model : Finetuned Enc + classifier Output : Label
CLIP	Data : CLIP dataset (im-tx) Model : CLIP Output : Image Text encoder	***NO FINETUNING***	**Data : Pascal test data + List of Labels** **Model : Image Text encoders** **Output : Label**

ZERO SHOT

CLIP

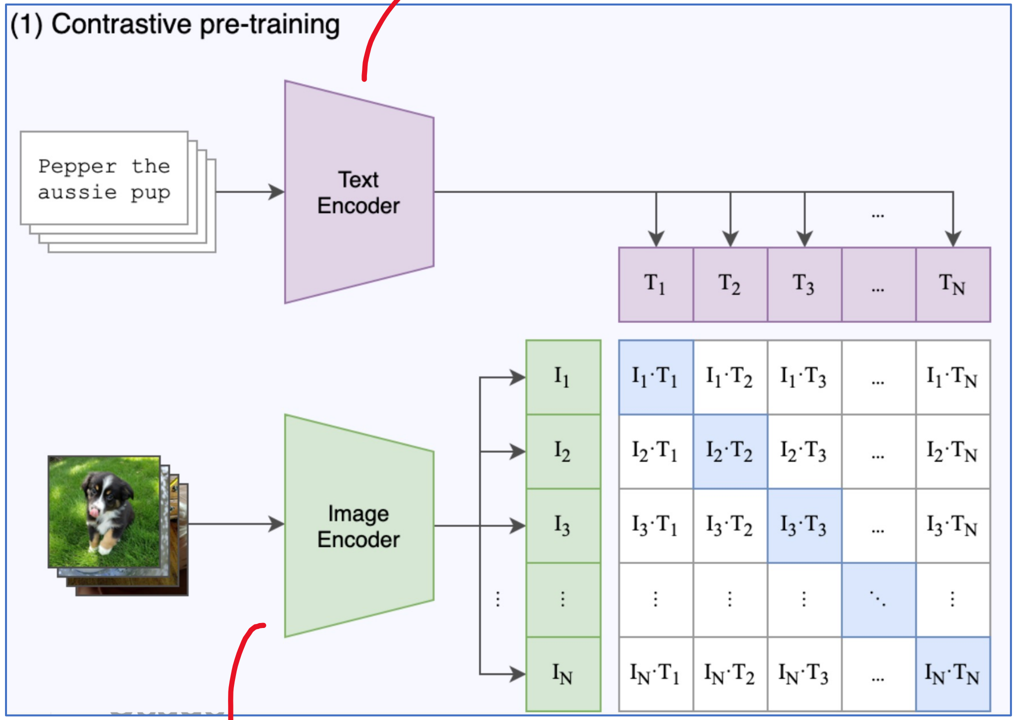
- Move away from labels – use natural language supervision instead
  - No need to collect ML grade data / crowdsourced labels anymore
  - Scalable! Take (image, text) pairs from the Internet
  - Use the power of language models
  - **300 million pairs!**
- Utilize the label meaning
  - Don't just take Label ID mapping at test
  - **Prompt Engineering**
  - "cat" -> "a photo of one {cat}"
- Zero-shot inference
  - No gradients / backprop!
  - Just like GPT-3
    - Translation, QA, NER, Coreference, etc
    - All of them can be framed as prompts

```
'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 jpeg corrupted photo of a {}.',
'a blurry photo of the {}.',
'a photo of the {}.',
'a good photo of the {}.',
'a rendering of the {}.',
'a {} in a video game.',
'a photo of one {}.',
'a doodle of a {}.',
'a close-up photo of the {}.',
'a photo of a {}.',
'the origami {}.',
'the {} in a video game.',
'a sketch of a {}.',
'a doodle of the {}.',
'a origami {}.',
'a low resolution photo of a {}.',
```

CLIP

GPT-x



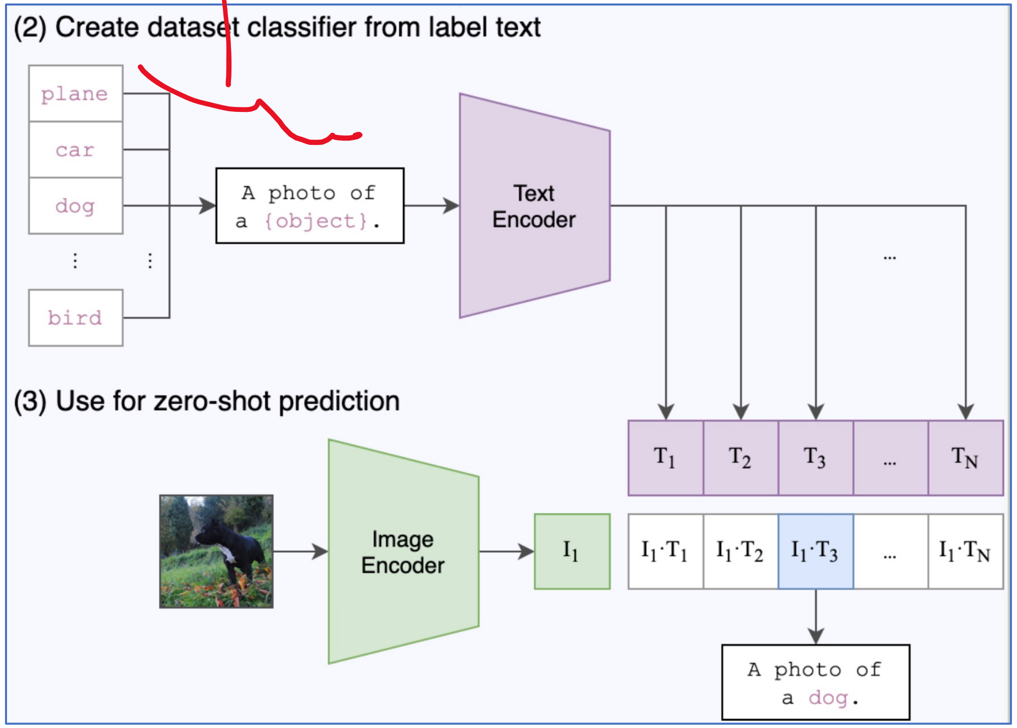
Training

ViT
(Transformer)

$\max_i I_i T_i$

$\min_{i \neq j} I_i T_j$

convert to prompt



Inference

CLIP - Example

```python
import torch
import clip
from PIL import Image
```

```python
device = "cuda" if torch.cuda.is_available() else "cpu"
model, preprocess = clip.load("ViT-B/32", device=device)
```

(1)

```python
image = preprocess(Image.open("CLIP.png")).unsqueeze(0).to(device)
text = clip.tokenize(["a diagram", "a dog", "a cat"]).to(device)
```

(2)

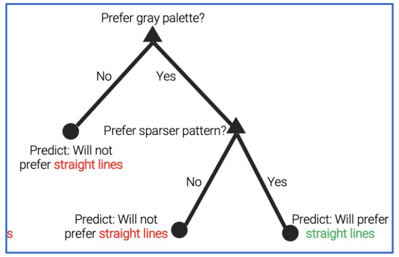
```python
with torch.no_grad():
    image_features = model.encode_image(image)
    text_features = model.encode_text(text)
```

(3)

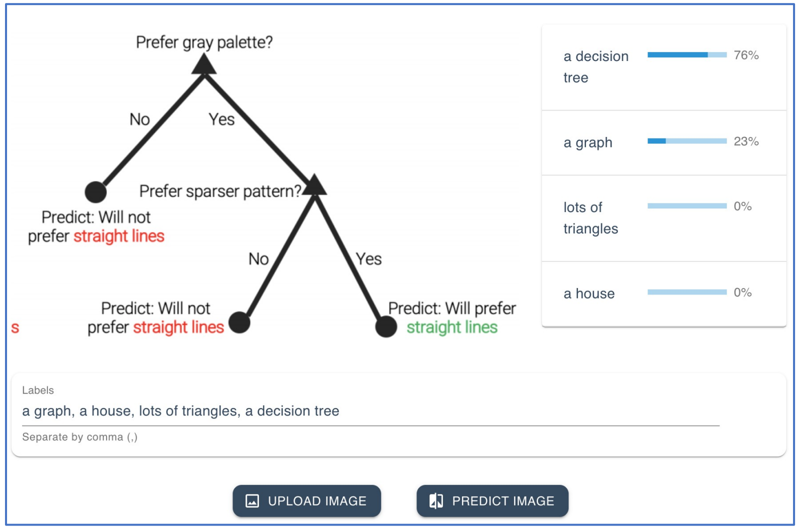
```python
    logits_per_image, logits_per_text = model(image, text)
    probs = logits_per_image.softmax(dim=-1).cpu().numpy()
```

(4)

```python
print("Label probs:", probs)  # prints: [[0.9927937  0.00421068 0.00299572]]
```

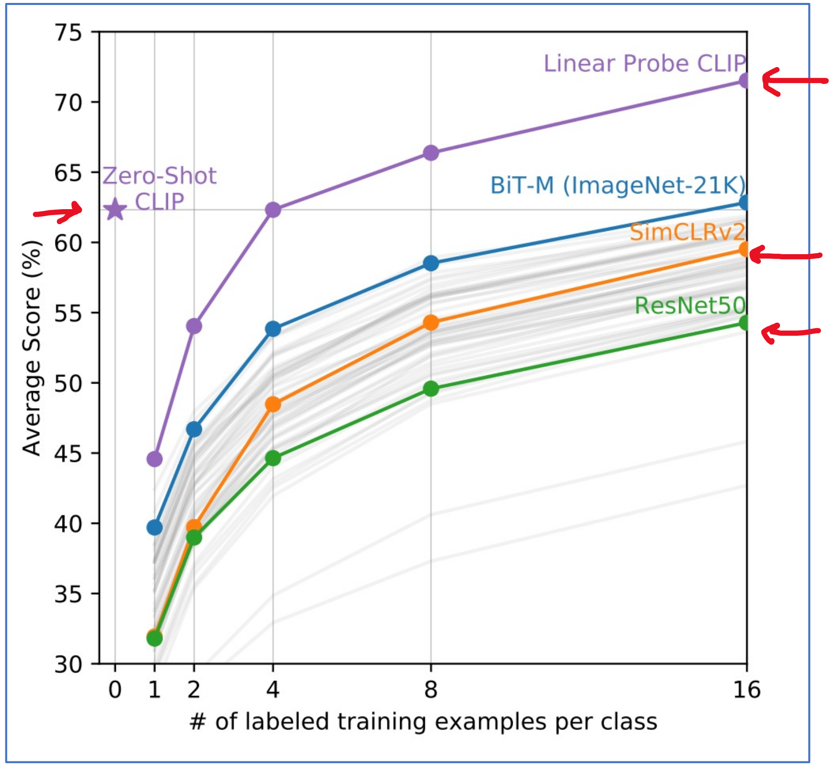


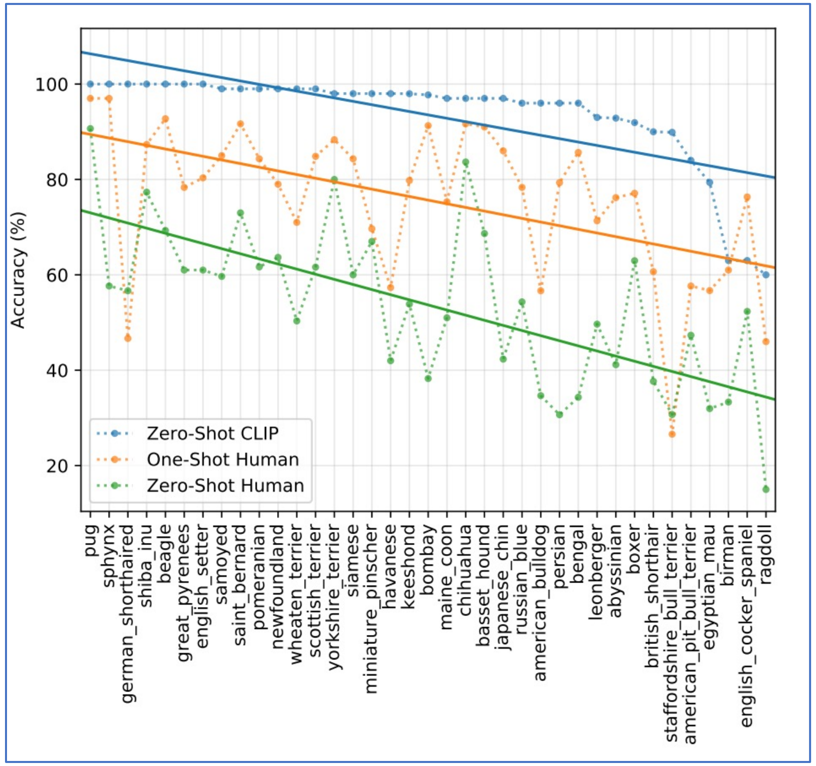
A graph
A house
Lots of triangles
A decision tree



Lots of demos - https://clip.kiri.ai/
Curated list - https://bit.ly/3jzRVAt [Image -> Text, Text -> Image]

CLIP Experiments – Good Representations





One shot accuracy jump

- Zero-shot CLIP much better than ResNet50 trained on ImageNet [in low labels regime]
- Also better than SimCLR [even without linear probe]
- Peculiar : CLIP + Linear Probe is worse than Zero-shot CLIP for 2-3 labels

Human performance improves with one-shot, two-shot cases

CLIP and Humans find difficulty in classifying for the same classes

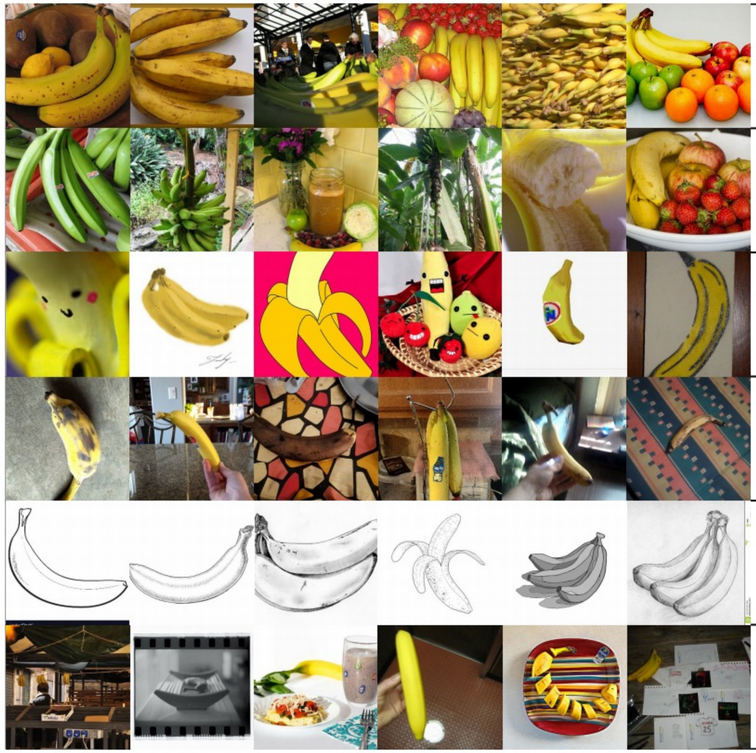
CLIP Experiments - Robustness

- Models trained (supervised) on ImageNet don’t transfer well even to small distribution shifts, such as: Sketches of same images, cartoons, diff backgrounds

- Zero-shot CLIP vastly outperforms supervised models against this distribution shift

- So, whenever we feel like using ResNet encoder – it is better to try CLIP encoder instead ☺

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%

Limitations / Comments

- Zero-shot CLIP isn't perfect
  - Only 88% accuracy on MNIST
  - Not so good with OCR / text-in-image scenarios
  - No matter how much data you scrap, there will always be out-of-data stuff
- Designing of Class labels [Prompt Engg]
  - Bad ways to use – classify surveillance images as "criminal" , "suspect" , etc.
  - You can use *any* class labels and get a free classifier
- Expected improvements
  - Counterintuitive drop in accuracy for 1-shot, 2-shot
  - More push towards avoiding finetuning
  - More contextual knowledge in text labels – physical, societal, geographical, etc.

Credits : Mark Saraoufim