

CLIP and DINO

Earl Wong

Historical

- In 1990, LeCun, et. al. introduced the idea of using a convolutional neural network (CNN) for digit classification.
- “ ... scan the input image with a single neuron that has a local receptive field, and store the states of this neuron in corresponding locations in a layer called a feature map. This operation is equivalent to convolution with a small size kernel, followed by a squashing function.”
- This construct produced excellent results for handwritten digit recognition.

Historical

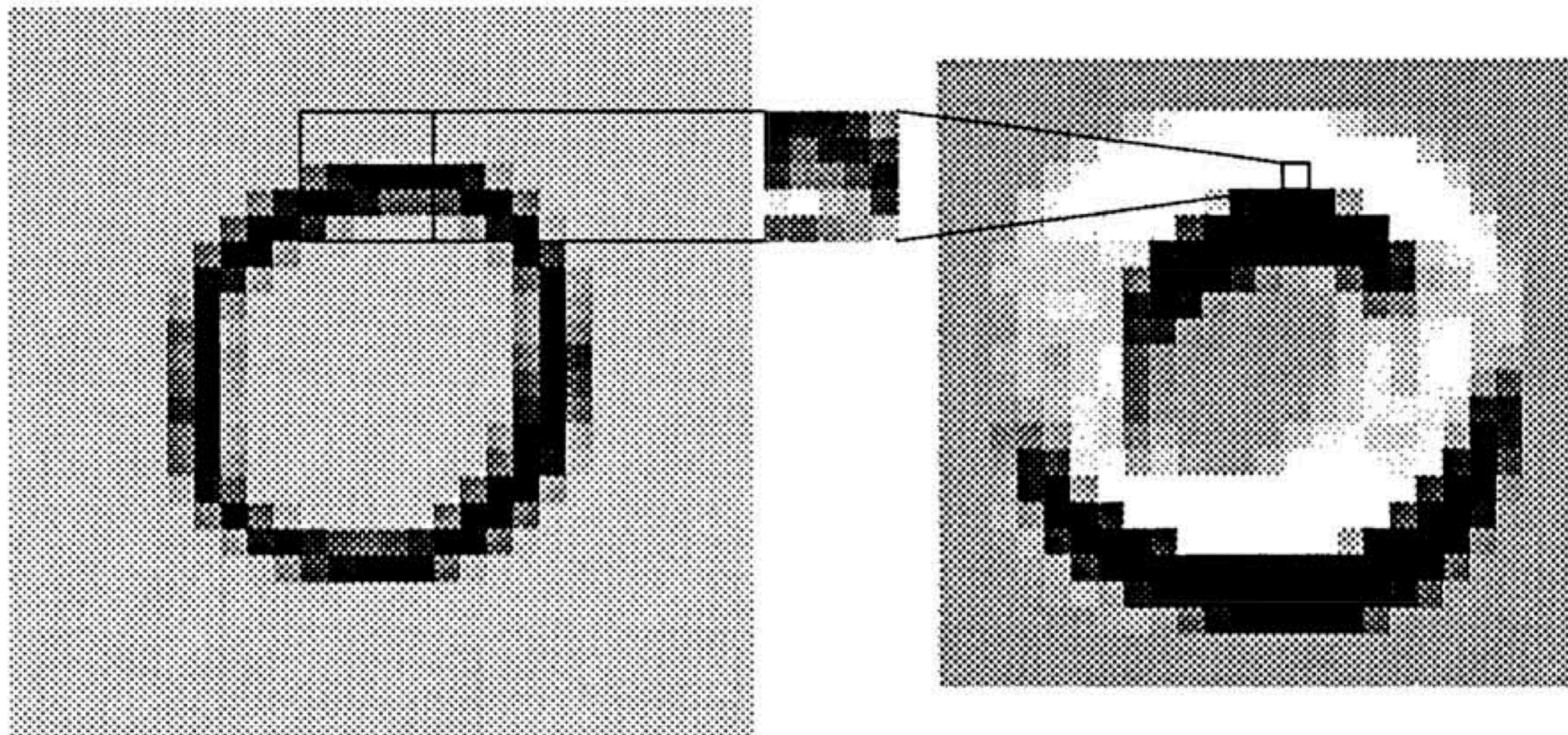


Figure 3: Input image (left), weight vector (center), and resulting feature map (right). The feature map is obtained by scanning the input image with a single neuron that has a local receptive field, as indicated. White represents -1, black represents +1.

Historical

- Next, let's fast forward to 2012, and the introduction of the first deep convolutional neural network (AlexNet) applied to the ImageNet Large Scale Visual Recognition Challenge (ILSVRC).
- The feature maps created by AlexNet contained both low level feature maps (consistent with edge and corner detectors, etc.)
...
- ... and in the deeper layers, high level semantic, feature maps / semantic information.
- AlexNet achieved state of the art results for image classification.

Historical

- i.e. Computer vision had finally emerged from the “dark age” of toy problems, with CIFAR10 and 32x32 input images.
- By 2015, ResNets had become the emerging winner for image classification.
- This would soon be followed by Yolo for object detection (2016) and Mask R-CNN for image segmentation (2017).

The Fundamental Shortcoming

- Although these networks produced state of the art results, they relied on a training strategy predicated on supervised learning.
- i.e. Labeled training data.
- Many companies that specialized in generating training data for computer vision applications emerged from this approach.
- At the same time, concerns for this approach continued to grow louder.
- Specifically: 1) The cost of the solution (labeled training data) and 2) the deployment issues (brittleness) resulting from the approach.

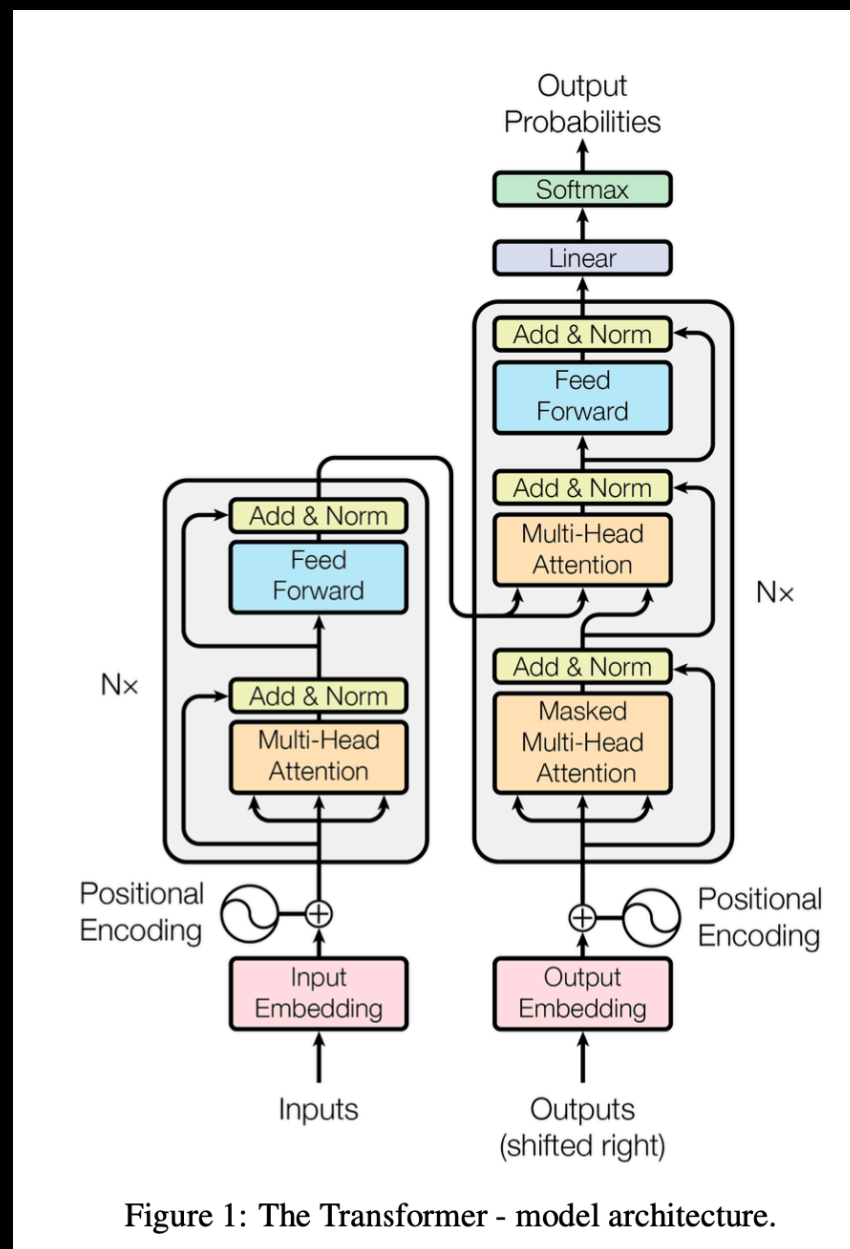
The Next Wave

- The “workarounds” / improvements took several different paths.
- Although the fundamental machinery for learning - back propagation, cross entropy loss and MAE / MSE loss, etc. - remained intact, other components would undergo change.
- i.e. A strong competitor to the CNN architecture would emerge.
- i.e. The need for manually labeled training data would diminish.

The Transformer

- In 2017, Vaswani, et. al. introduced the transformer architecture, designed for machine translation tasks.
- “We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely.”
- Some of the major strengths of the transformer included 1) the ability to easily scale, 2) the ability to address longer term time dependencies using an attention mechanism and 3) faster training.

The Transformer



$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

The Transformer

- Although (initially) targeted for text, the authors also saw the potential for images, video and audio.

Specifically:

- Whereas CNN's relied on increasing the “effective” receptive field to obtain information from different parts of an image, transformers could easily relate one part of an image to any other part, using attention.
- Whereas CNN's gained power by stacking blocks with a different number of feature maps at different spatial resolutions, transformers gained power by stacking an identical transformer block.

The Transformer

- In 2020, Dosovitskiy, et. al., demonstrated the approximate parity of Transformers vs. CNN's for vision.
- Although trained using labeled data, Dosovitskiy's results illustrated the potential of Vision Transformers (ViT), as the ViT's size / capacity increased.
- In addition, it was also shown that ViT's trained faster than their CNN counterparts.

Vision Transformer (ViT)

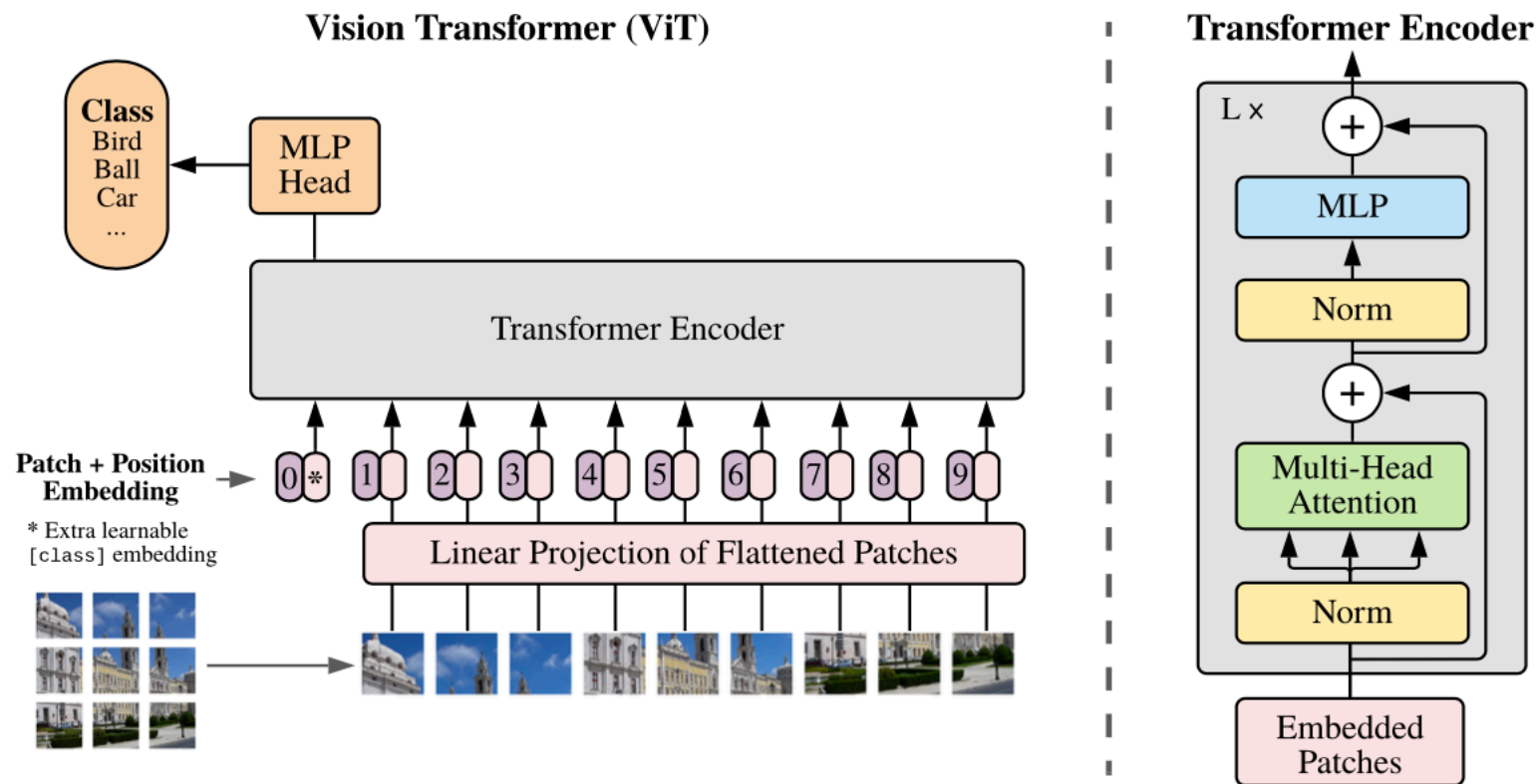


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable “classification token” to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

Comparisons With Convnet Based Approaches


	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	88.55 ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	90.72 ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	99.50 ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	—
CIFAR-100	94.55 ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	—
Oxford-IIIT Pets	97.56 ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	—
Oxford Flowers-102	99.68 ± 0.02	99.74 ± 0.00	99.61 ± 0.02	99.63 ± 0.03	—
VTAB (19 tasks)	77.63 ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	—
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Table 2: Comparison with state of the art on popular image classification benchmarks. We report mean and standard deviation of the accuracies, averaged over three fine-tuning runs. Vision Transformer models pre-trained on the JFT-300M dataset outperform ResNet-based baselines on all datasets, while taking substantially less computational resources to pre-train. ViT pre-trained on the smaller public ImageNet-21k dataset performs well too. *Slightly improved 88.5% result reported in [Touvron et al. \(2020\)](#).

Contrastive Language Image Pre-training (CLIP)

- In 2021, Radford, et. al., introduced some new thinking into the computer vision deep learning workflow.
- In response to the existing approach for training deep networks using labeled training data, the following was suggested:
- “Learning directly from raw text about images is a promising alternative, which leverages a much broader source of supervision.”

The Simple Extension

- Children learn, by first identifying objects: Pig, Cow, etc.
- Then, comes the simple description, like the sound it makes.
- Fisher Price Classic: See and Say Toy A young child with curly hair, wearing a blue shirt, is smiling and holding a yellow Fisher Price See and Say toy. The toy is a circular ring with various colorful icons of animals and objects.
- Hence, while first generation training datasets contained labels of the identifying object, the second generation training datasets should also contain simple and representative descriptions of the object.

CLIP

- What could be the potential source of this text-image information?
- Captions from images scraped from the internet.

How many could be acquired?

- 400 million.

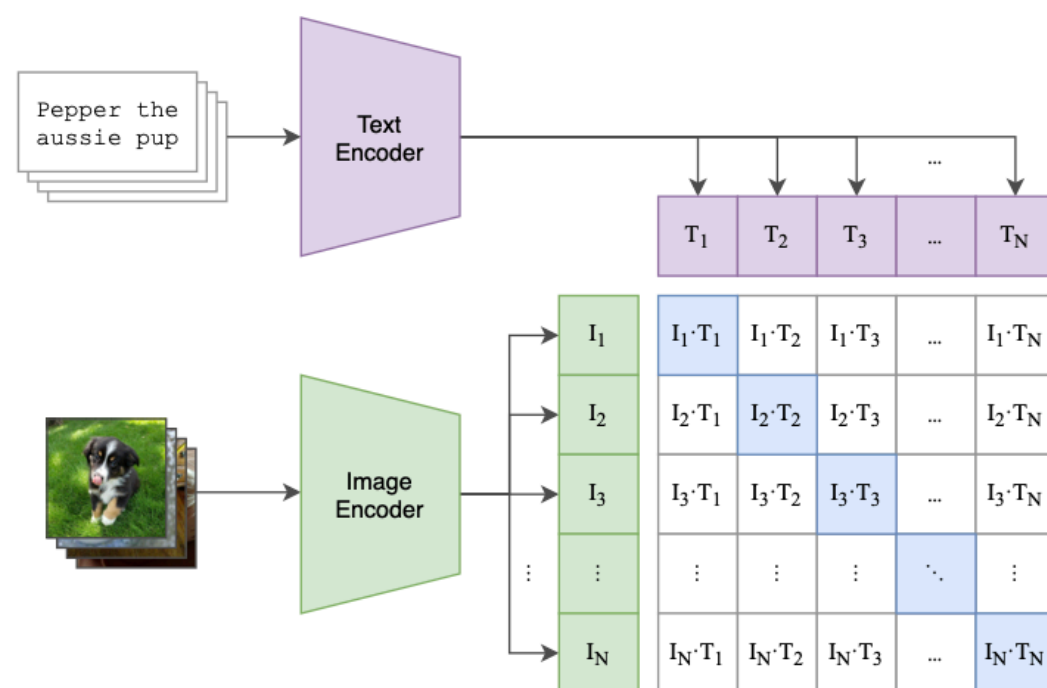
CLIP

What were the benefits and what was needed to implement the idea?

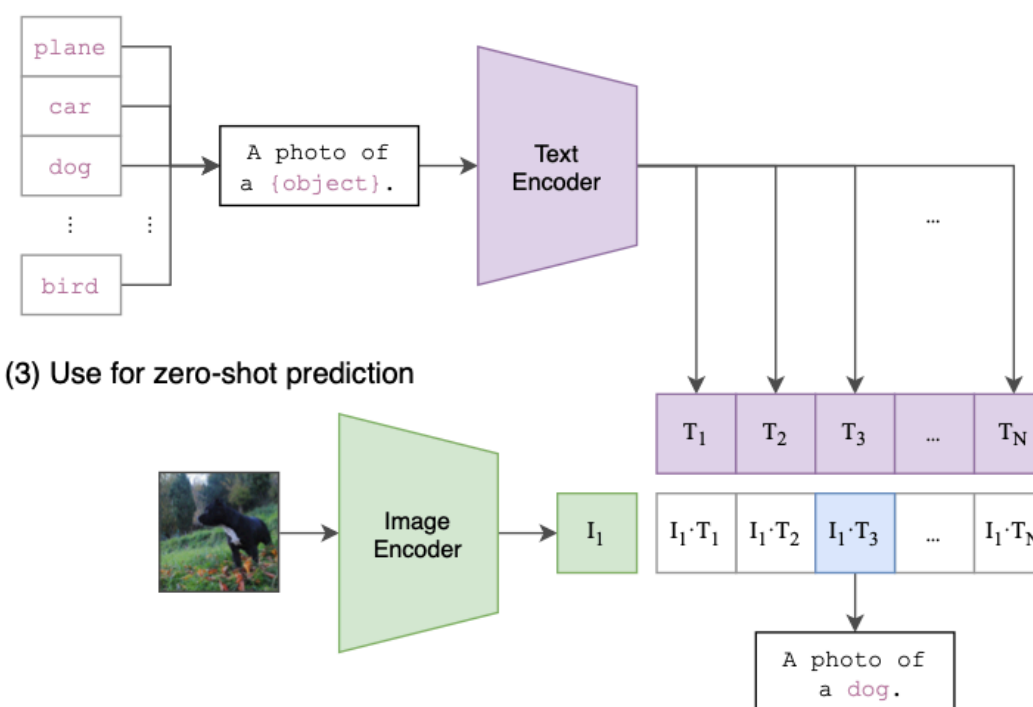
- Benefit1) More category coverage, without the manual (human) effort.
- Benefit2) Richer category information without the manual (human) effort.
- Needed) A model that was favorable for consuming the richer information.

CLIP

(1) Contrastive pre-training



(2) Create dataset classifier from label text



(3) Use for zero-shot prediction

Figure 1. Summary of our approach. While standard image models jointly train an image feature extractor and a linear classifier to predict some label, CLIP jointly trains an image encoder and a text encoder to predict the correct pairings of a batch of (image, text) training examples. At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes.

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.

CLIP Nuances

- Classical supervision is still used.
- Only now, the supervision involves a rich, text description of the object and not just a one hot label.
- Manual labor is still needed to produce the text description.
- Only now, the text description is the byproduct of a captioned image from a news article, etc. and can be freely scraped from the internet.

Interesting Follow on Questions

- Is CLIP more “one shot like”, because the text-image training data contains more categories than ImageNet?
- Does this argument have merit, because CLIP does not perform well on categories with finer descriptions. i.e. Dog, but what type of dog?
- What if we added text descriptions to ImageNet (in addition to the category label) and (now) ran everything through CLIP?
- What if we implemented CLIP using an RNN text encoder instead of a transformer?
- Would CLIP (RNN-ResNet) produce a better result than CLIP (RNN-ViT) since the text descriptions in captions is not very long?

More Interesting Points and Questions

- Qualitative argument: Using contrastive loss with text-image pairs, produces richer training information.
- Yet - sigLIP produces better results than CLIP, using non-contrastive text-image pairs.
- Empirical argument: ViT's perform better at capturing the rich feature descriptors versus their convnet (ResNet) counterparts.
- Suppose there was no useful information at the boundary of the images (beyond the effective receptive field). Would the ViT and the convnet result (then) mathematically converge?
- Are image resolution distributions for ImageNet and the 400 million text-image pairs (scraped for CLIP) approximately the same?

Distillation No Labels (DINO)

- In a parallel development, Caron, et. al., released DINO in 2021.
- Unlike CLIP and SigLIP, DINO did not learn feature representations using text-image pairs.
- In DINO, immediate supervision was NOT used, since no labeled data was used - either directly (like with ImageNet) or indirectly (like with text-image pairs).
- Instead, DINO employed a modified teacher-student architecture, to learn it's feature representations.

DINO

- DINO used identical Vision Transformer (ViT) networks for its teacher and student.
- For any given image, a set of transformations (1-1 or crop) were applied, generating a new set of images.
- From this new set, the teacher was shown the global images while the student was shown the cropped images.
- A cumulative, “set based” cross entropy loss was then computed, by aggregating the losses from the different permutations.
- This aggregate cross entropy loss was then back propagated ONLY through the student network.

DINO

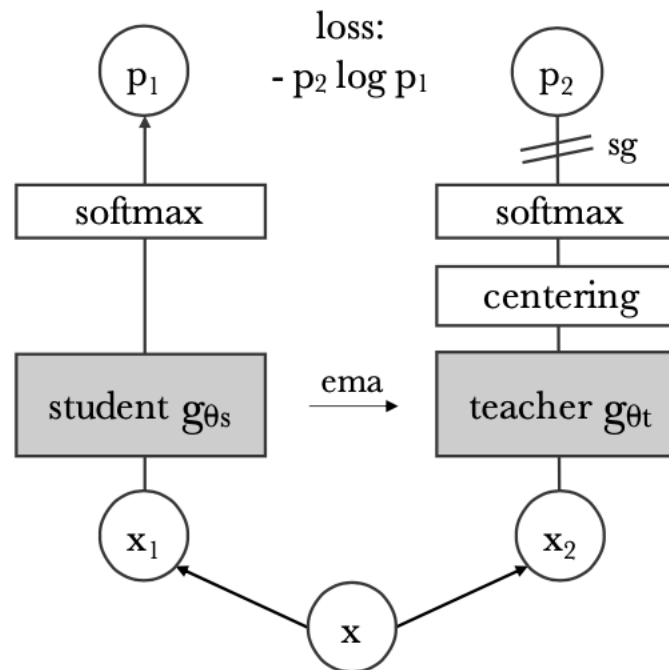


Figure 2: **Self-distillation with no labels.** We illustrate DINO in the case of one single pair of views (x_1, x_2) for simplicity. The model passes two different random transformations of an input image to the student and teacher networks. Both networks have the same architecture but different parameters. The output of the teacher network is centered with a mean computed over the batch. Each networks outputs a K dimensional feature that is normalized with a temperature softmax over the feature dimension. Their similarity is then measured with a cross-entropy loss. We apply a stop-gradient (sg) operator on the teacher to propagate gradients only through the student. The teacher parameters are updated with an exponential moving average (ema) of the student parameters.

DINO

- Upon initial inspection, doubts should arise, regarding how the architecture and training recipe, manages to learn meaningful feature representations.
- Easy question: How does the teacher network learn it's representations, if there is no back propagation through the teacher?
- The teacher network learns from an exponential moving average of the student weights = momentum encoder.
- Harder question: If there are no labels (and /or text descriptions), where is the information coming from?
- The information is contained in the image set created for each training image, and the process of showing the teacher the global images and the student the cropped images. By construction, multi-resolution learning is occurring?

DINO

- Hard question: But why would the DINO architecture learn anything at all?
- i.e. The cross entropy loss is dependent on both the student and the teacher outputs, and the teacher starts with zero knowledge. You can't bootstrap from zero.
- i.e. In addition, the student is actually teaching the teacher via the momentum encoder.

DINO

- Let's start with the second issue.
- By construction, the variance (of the results / information) in teacher network will always be smaller than the student network.
- This is because ensemble averaging is being performed in the teacher network.
- Hence, the teacher will always possess the better, lower variance information.
- This better information then “shows up” during the loss computation for back propagation.

DINO

- Now, to the former question.
- If both the teacher and the student were shown non-overlapping crops from the created set, meaningful representations would be impossible to learn, since the two images would be uncorrelated, sans a texture image.
- If both the teacher and the student were shown the same crops or the same global images from the created set, meaningful representations would also be impossible to learn, since the two images are identical = no information.
- The fact that the teacher only sees the global images while the student only sees the crops, and that the architecture forces a loss to be computed between the two correlated inputs, results in the learning.
- The learning can be interpreted as learning the transformations needed to map one input image output, to the other.

DINO

- Having said this, the number of training images is large and varied.
- Hence, the output signal could collapse / disappear / become very small for different image pairs.
- To prevent this, various tricks were employed, to prevent the teacher output from collapsing.
- Specifically, centering was used, to prevent any specific output dimension from dominating.
- However, centering then encouraged collapse of the output to a uniform distribution.
- To counteract THIS, sharpening was then applied.

Summary

- The improved successor of CLIP is SigLIP [2023].
- SigLIP uses a different loss function than CLIP, producing significantly better results.
- The improved successor of DINO is DINOv2 [2023].
- Although no revolutionary changes were introduced, many small changes to DINO significantly improved the output result for DINOv2.

Summary

- CLIP can be viewed as a significantly improved extension to ImageNet based training, by incorporating a language component.
- In contrast, DINO learns improved feature maps, by removing constraints from the learning process - no labeled information or descriptions.
- Qualitatively, we would expect the learned feature maps from DINO to be better than those from CLIP, when used for fundamental vision tasks such as segmentation, etc.
- In general, the feature representations learned for downstream tasks using CLIP and DINO have significantly improved.