TTIC 31230, Fundamentals of Deep Learning

David McAllester, Autumn 2024

Maximizing Mutual Information

Contrastive Coding

Cotraining

The Visual Transformer (ViT)

Maximizing Mutual Information

Assume a population of pairs (x, y).

- x might be an image and y might be the text of a caption for image x (CLIP).

- x might be an video frame and y video frame a second later.

- x might be a window of a sound wave and y a later window (Wav2Vec).

- x and y are different transformations of an image z such as translation, rotation, color shift, or cropping. (SimCLR,DINO)

Maximizing Mutual Information

The Information Bottleneck Method Tishby, Pereira, Bialek, (1999).

Assume a distribution on pairs (x, y) and an encoder $P_{\mathrm{enc}}(z|x)$.

$$\mathrm{enc}^* = \underset{\mathrm{enc}}{\mathrm{argmax}}\, I(z, y) - \beta I(z, x)$$

Maximizing Mutual Information

The methods discussed here — contrastive coding and cotraining — can be interpreted as optimizing only the first term in the information bottleneck.

We take the encoder to be deterministic $\mathrm{enc}(x) \in R^d$ and maximize mutual information with y.

$$\mathrm{enc}^* = \underset{\mathrm{enc}}{\mathrm{argmax}}\ I(\mathrm{enc}(x), y)$$

Here there is no incentive in this objective for $\mathrm{enc}(x)$ to retain information unrelated to y.

Maximizing Mutual Information.

$$\mathrm{enc}^* = \underset{\mathrm{enc}}{\mathrm{argmax}}\; I(\mathrm{enc}(x), y)$$

It would be natural to try to maximize a variational lower bound on $I(\mathrm{enc}(x), y)$.

In the applications discussed here we expect hundreds of bits of mutual information.

Unfortunately one can prove that no formal lower bound establishing hundreds of bits of mutual information is possible without an exponential (2^{100}) number of training examples.

Formal Limitations on the Measurement of Mutual Information, David McAllester, Karl Stratos, (November 2018)

Surrogate Objectives and Hardness Parameters

Since variation optimization of mutual information is not possible, contrastive learning and cotraining each introduce a surrogate objective.

Each surrogate objective has a parameter — the hardness parameter — that controls the difficulty of the optimization problem.

Increasing the hardness parameter makes training more difficult but should increase the resulting mutual information $I(\mathrm{enc}(x), y)$ when training is successful.

Contrastive Coding

Representation Learning with Contrastive Predictive Coding, van den Oord, Li and Vinyals (DeepMind, 2018)

CLIP: **Learning Transferable Visual Models From Natural Language Supervision** (OpenAI, February 2021)

The Contrastive Coding Desiderata

We draw a batch of pairs $(x_1, y_1), \ldots, (x_B, y_B)$.

We select one of the x values.

We train a classifier that, when given one of the x values and the batch $(y_1, \ldots, y_B)$ of y values, must determine which y was paired with the given x.

The Contrastive Coding Surrogate Objective

We train two encoders enc_x and enc_y with $\mathrm{enc}_x(x) \in R^d$ and $\mathrm{enc}_y(y) \in R^d$.

$$\mathrm{enc}_x^*, \mathrm{enc}_y^* = \underset{\mathrm{enc}_x,\mathrm{enc}_y}{\mathrm{argmin}} \begin{array}{l} E_{(b,x_b,y_1,\ldots,y_B)} \\ +E_{(b,y_b,x_1,\ldots,x_B)} \end{array} \begin{array}{l} \left[-\ln P_{\mathrm{enc}_x,\mathrm{enc}_y}(b|x_b, y_1, \ldots, y_B)\right] \\ \left[-\ln P_{\mathrm{enc}_x,\mathrm{enc}_y}(b|y_b, x_1, \ldots, x_B)\right] \end{array}$$

$$\begin{aligned} P_{\mathrm{enc}_x,\mathrm{enc}_y}(b|x, y_1, \ldots, y_B) &= \underset{b}{\mathrm{softmax}}\ \mathrm{enc}_x(x)^\top \mathrm{enc}_y(y_b) \\ P_{\mathrm{enc}_x,\mathrm{enc}_y}(b|y, x_1, \ldots, x_B) &= \underset{b}{\mathrm{softmax}}\ \mathrm{enc}_y(y)^\top \mathrm{enc}_x(x_b) \end{aligned}$$

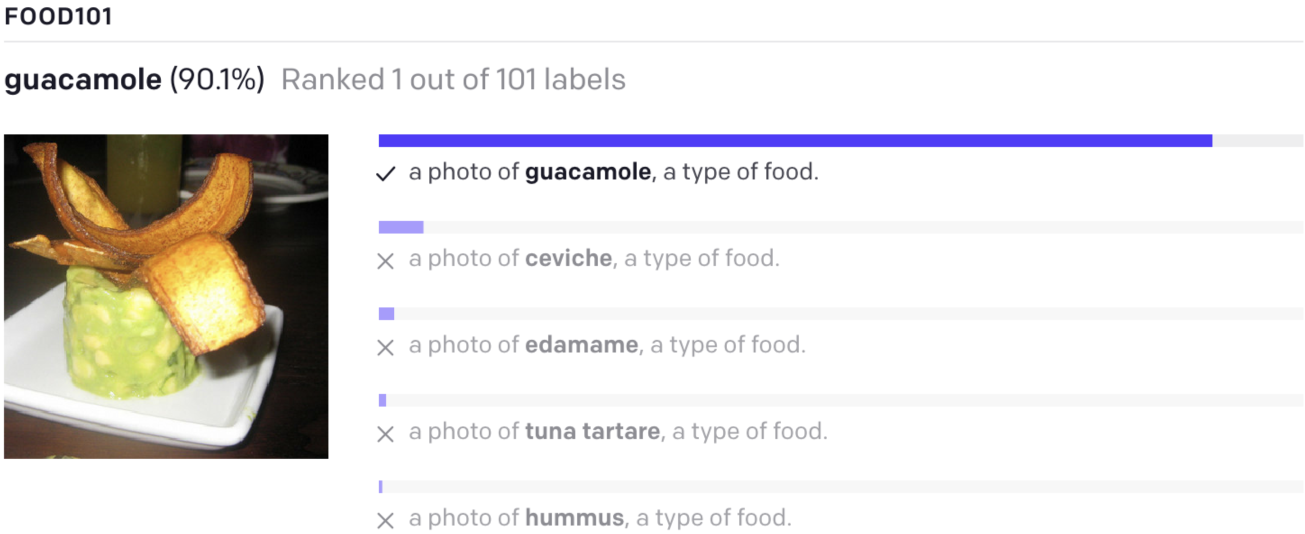
The Contrastive Coding Hardness Parameter

$$\begin{aligned}\mathcal{L}(\mathrm{enc}_x, \mathrm{enc}_y) & = & E_{(b,x_b,y_1,\ldots,y_B)}\left[-\ln P_{\mathrm{enc}_x,\mathrm{enc}_y}(b|x_b, y_1, \ldots, y_B)\right] \\ & & +\; E_{(b,y_b,x_1,\ldots,x_B)}\left[-\ln P_{\mathrm{enc}_x,\mathrm{enc}_y}(b|y_b, x_1, \ldots, x_B)\right]\end{aligned}$$

The hardness parameter is the batch size B. Making B large makes the classification task harder.

In CLIP $B = 2^{15} = 32{,}768$.

Zero-Shot Image Classification



FOOD101

guacamole (90.1%) Ranked 1 out of 101 labels

✓ a photo of **guacamole**, a type of food.

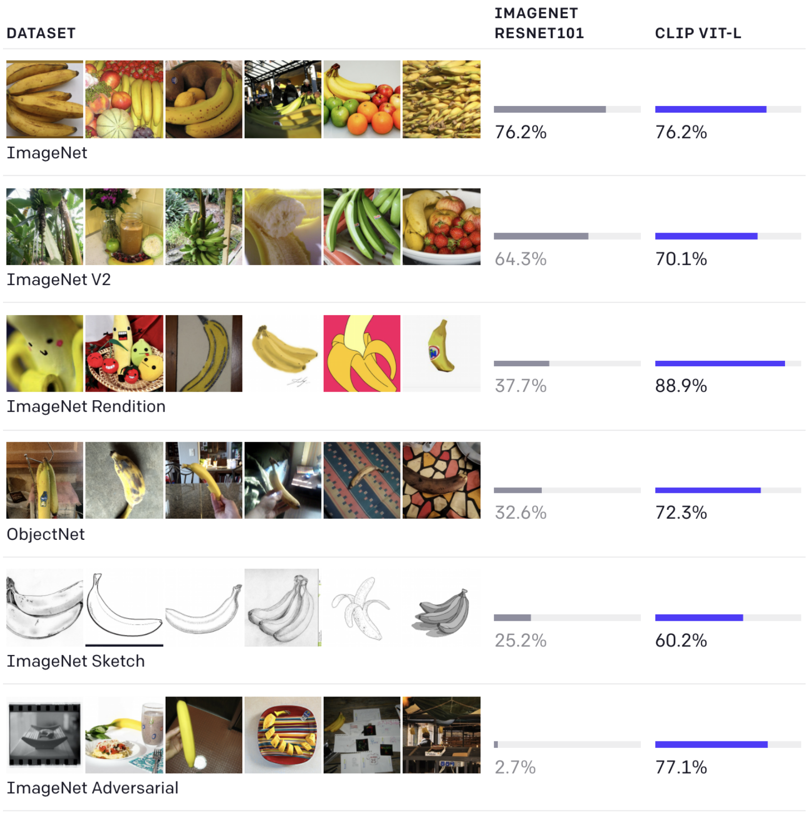
✗ a photo of **ceviche**, a type of food.

✗ 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.

Zero-Shot Image Classification



DATASET	IMAGENET RESNET101	CLIP VIT-L
ImageNet	76.2%	76.2%
ImageNet V2	64.3%	70.1%
ImageNet Rendition	37.7%	88.9%
ObjectNet	32.6%	72.3%
ImageNet Sketch	25.2%	60.2%
ImageNet Adversarial	2.7%	77.1%

Although both models have the same accuracy on the ImageNet test set, CLIP's performance is much more representative of how it will fare on datasets that measure accuracy in different, non-ImageNet settings. For instance, ObjectNet checks a model's ability to recognize objects in many different poses and with many different backgrounds inside homes while ImageNet Rendition and ImageNet Sketch check a model's ability to recognize more abstract depictions of objects.

Cotraining

Combining labeled and unlabeled data with co-training Avrim Blum, and Tom Mitchell (1998)

PAC Generalization Bounds for Co-training, Dasgupta, Littman, McAllester (2001)

DINOv1: **Emerging Properties in Self-Supervised Vision Transformers**, (Meta, Inria, April 2021)

iBOT: **Image BERT Pre-Training with Online Tokenizer**, Zhou et al. (ByteDance, January 2022)

DINOv2: **Learning Robust Visual Features without Supervision**, (Meta, Inria, February 2024)

Cotraining

We again consider a population distribution on pairs (x, y).

In cotraining we assume a label set $k \in \{1, \ldots, K\}$.

The labels have no a-priori meaning. Meaning will emerge from the training.

We train two classifiers $P_\Phi(k|x)$ and $P_\Psi(k|y)$ under a cotraining objective.

Cotraining Desiderata

In cotraining we seek to find two classifiers $P_\Phi(k|x)$ and $P_\Psi(k|y)$ such that on typical pairs (x, y) the two classifiers agree on the label.

(1) For a given pair (x, y) we want each classifier to be confident in its label. Formally, we want that in expectation over the draw of a pair (x, y) the entropies $H(P_\Phi(k|x))$ and $H(P_\Psi(k|y))$ are both small.

(2) For a given pair (x, y) we want the classifiers to agree on the label. Formally, we want that in expectation over a draw of a pair (x, y) the cross-entropies $H(P_\Phi(k|x), P_\psi(k|y))$ and $H(P_\psi(k|y), P_\Phi(k|x))$ are both small.

(3) We want that over different draws of (x, y) the distributions of predicted labels covers all the labels. Formally, we want the marginal distributions $P_\Phi(k)$ and $P_\Psi(k)$ to have large entropy.

The Cotraining Surrogate Objective

$$\Phi^*, \Psi^* = \operatorname*{argmin}_{\Phi,\Psi} \; E_{(x,y)\sim\mathrm{Pop}} \left\{ \begin{array}{rl} & H(P_\Phi(k|x), P_\Psi(k|y)) \\ + & H(P_\Psi(k|y), P_\Phi(k|x)) \\ & \\ - & \beta(H(P_\Phi(k)) + H(P_\Psi(k))) \end{array} \right.$$

Here we have subsumed the criterion (1) – that the classifiers are confident – into the cross-entropy terms. Making the cross entropies small requires confidence of both classifiers as well as agreement.

The Cotraining Hardness Parameter

$$\Phi^*, \Psi^* = \operatorname*{argmin}_{\Phi,\Psi}\; E_{(x,y)\sim \mathrm{Pop}} \left\{ \begin{array}{ll} & H(P_\Phi(k|x), P_\Psi(k|y)) \\ + & H(P_\Psi(k|y), P_\Phi(k|x)) \\ & \\ - & \beta(H(P_\Phi(k)) + H(P_\Psi(k))) \end{array} \right.$$

The Hardness parameter is K, the number of labels. Getting agreement on all labels, when all labels are used, is hard when the number of labels is large.

in DINOv2 $K = 2^{10} + 2^9 = 1536$.

Collapse of the Surrogate Objective

$$\Phi^*, \Psi^* = \underset{\Phi,\Psi}{\mathrm{argmin}}\ \ E_{(x,y)\sim\mathrm{Pop}} \left\{ \begin{array}{ll} & H(P_\Phi(k|x), P_\Psi(k|y)) \\ + & H(P_\Psi(k|y), P_\Phi(k|x)) \\ & \\ - & \beta(H(P_\Phi(k)) + H(P_\Psi(k))) \end{array} \right.$$

SGD on the this surrogate objective collapses into making both $P_\Phi(k|x)$ and $P_\Psi(k|y)$ uniform in k.

This maximizes $H(P_\Phi(k))$ and $H(P_\Psi(k))$ and gives

$$\begin{array}{l} \Phi = \underset{\Phi}{\mathrm{argmin}}\, E_{(x,y)}\ H(P_\Phi(k|x), P_\Psi(k|y)) + H(P_\Psi(k|y), P_\Phi(k|x)) \\ \Psi = \underset{\Psi}{\mathrm{argmin}}\, E_{(x,y)}\ H(P_\Phi(k|x), P_\Psi(k|y)) + H(P_\Psi(k|y), P_\Phi(k|x)) \end{array}$$

Hence this is a local optimum of SGD.

Collapse of the Surrogate Objective

More generally whenever $P_\Phi(k|x)$ and $P_\Psi(k|y)$ agree we have.

$$\begin{aligned}
\Phi &= \operatorname*{argmin}_{\Phi} E_{(x,y)}\; H(P_\Phi(k|x), P_\Psi(k|y)) + H(P_\Psi(k|y), P_\Phi(k|x)) \\
\Psi &= \operatorname*{argmin}_{\Psi} E_{(x,y)}\; H(P_\Phi(k|x), P_\Psi(k|y)) + H(P_\Psi(k|y), P_\Phi(k|x))
\end{aligned}$$

So the gradients on Φ and Ψ vanish.

We can improve the cross-entropies by reducing both $H(P_\Phi(k|x))$ and $H(P_\Psi(k|y))$ simultaneously. However, the amount of improvement is quadratic (second order) in the change in parameters.

It seems that SGD is attracted to saddle points!

DINOv1

We will discuss the simpler version DINOv1. DINOv2 is similar but with a variety of engineering improvements.

The main contribution of DINO is a method for avoiding collapse of the training into uniform distributions $P_\Phi(k|x)$ and $P_\Psi(k|y)$.

They their innovation “self-distillation”.

The basic idea of self-distillation is to do a brute-force enforcement the desiderata (1)-(3) given above.

Self-Distillation

They construct a "teacher network" $\tilde{P}_\Phi(k|x)$ which uses the same parameters as $P_\Phi(k|x)$ but which is modified to enforce (encourage) the desiderata.

They then replace the cross-entropy losses in the cotraining objective with

$$H(sg(\tilde{P}_\Phi(k|x)), P_\Psi(k|y)) \text{ and } H(sg(\tilde{P}_\Psi(k|y)), P_\Phi(k|x)).$$

We train $P_\Psi(k|y)$ to model $\tilde{P}_\Phi(k|x)$ and train $P_\Phi(k|x)$ to model $\tilde{P}_\Psi(k|y)$. sg is the stop-gradient operator.

Why the stop-gradient sg?

$$H(\tilde{P}_\Phi(k|x), P_\Psi(k|y)) = E_{k \sim \tilde{P}_\Phi(k|x)}\left[-\ln P_\Psi(k|y)\right]$$

Computing the gradient with respect to Φ is problematic in any case — optimizing a sampling distribution is problematic in general.

We simply avoid the issue by not trying to compute this gradient.

We sample from $\tilde{P}_\Phi(k|x)$ and compute gradients on Ψ.

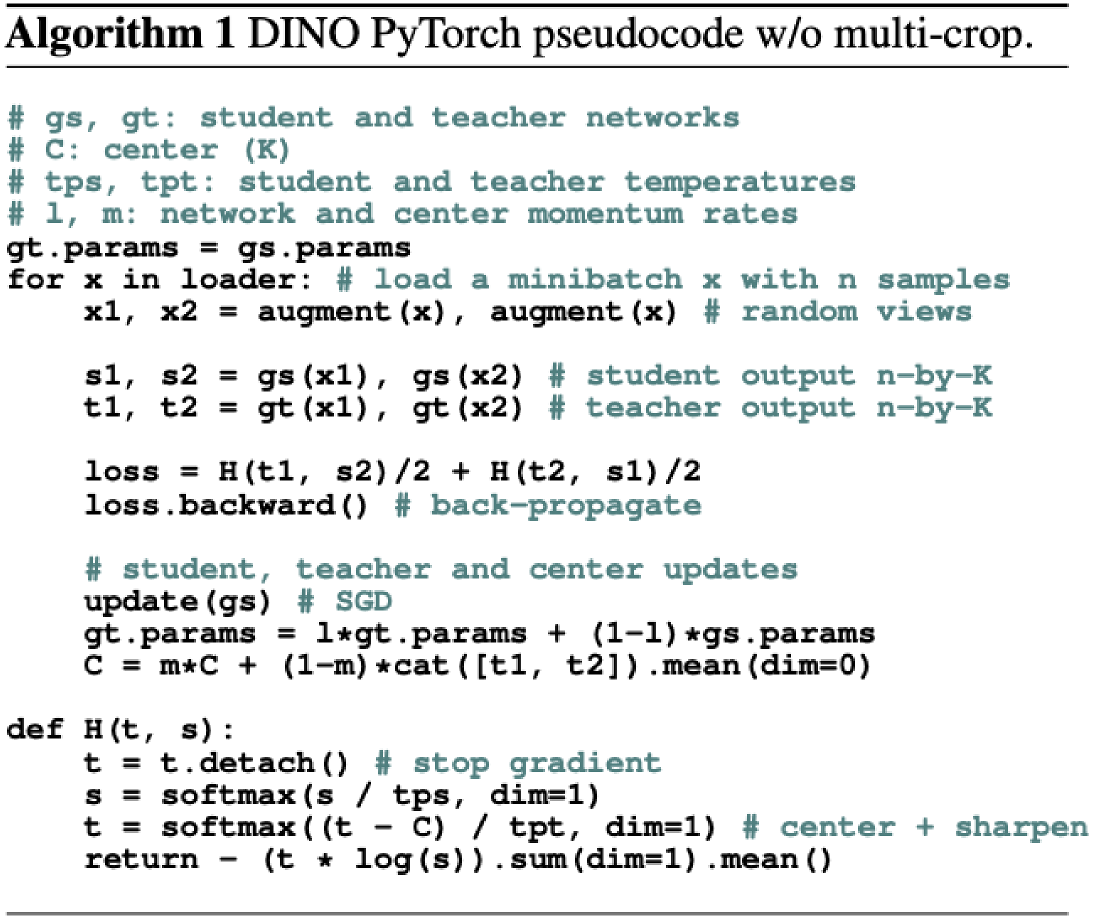
Encouraging the Desiderata

Increasing Confidence (Sharpening): They put a temperature parameter into the softmax of $\tilde{P}_\Phi(k|x)$. This allows $\tilde{P}_\Psi(k|x)$ to be defined at a lower temperature (higher β) which "sharpens" the distribution.

Diversifying the Label Usage (Centering): For each category k we compute an EMA over the draw of pairs (x, y) of the score (the logit) $s_\Phi(k, x)$. Denote this EMA value by $\overline{s}(k)$. The softmax defining the distribution $\tilde{P}_\Phi(k|x)$ is then

$$\tilde{P}_\Phi(k|x) = \underset{k}{\mathrm{softmax}}\ \beta(s_\Phi(k|x) - \overline{s}(k))$$

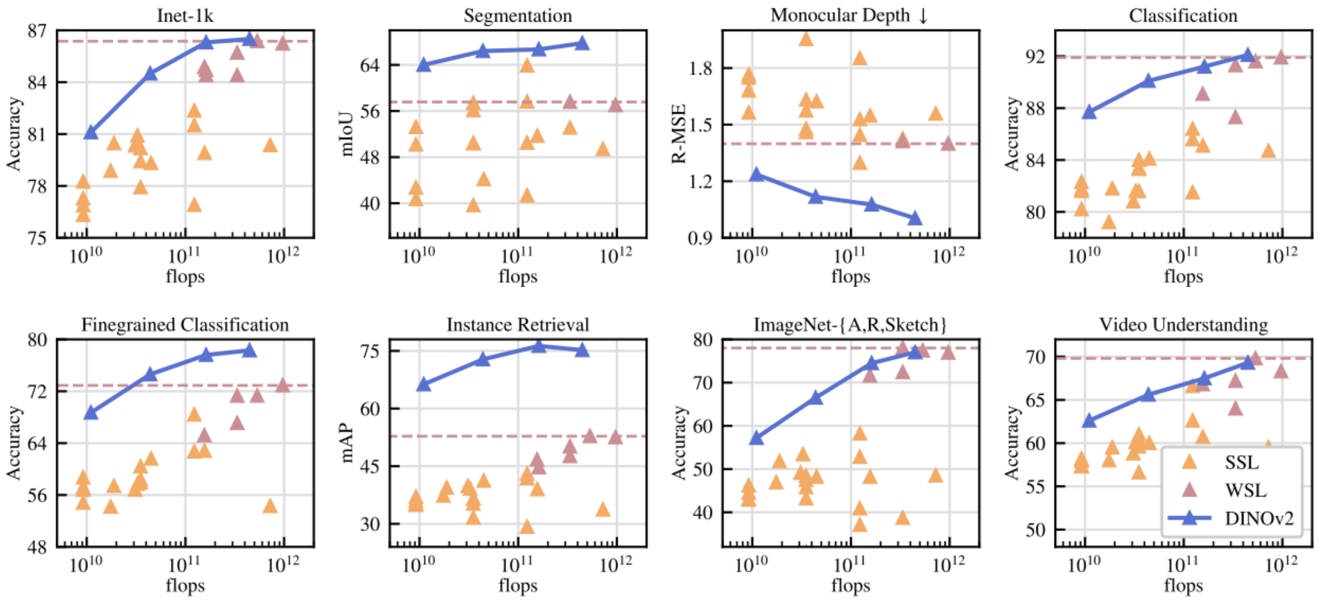
DIVOv1 PyTorch Pseudo-code



Algorithm 1 DINO PyTorch pseudocode w/o multi-crop.

```
# gs, gt: student and teacher networks
# C: center (K)
# tps, tpt: student and teacher temperatures
# l, m: network and center momentum rates
gt.params = gs.params
for x in loader: # load a minibatch x with n samples
    x1, x2 = augment(x), augment(x) # random views

    s1, s2 = gs(x1), gs(x2) # student output n-by-K
    t1, t2 = gt(x1), gt(x2) # teacher output n-by-K

    loss = H(t1, s2)/2 + H(t2, s1)/2
    loss.backward() # back-propagate

    # student, teacher and center updates
    update(gs) # SGD
    gt.params = l*gt.params + (1-l)*gs.params
    C = m*C + (1-m)*cat([t1, t2]).mean(dim=0)

def H(t, s):
    t = t.detach() # stop gradient
    s = softmax(s / tps, dim=1)
    t = softmax((t - C) / tpt, dim=1) # center + sharpen
    return - (t * log(s)).sum(dim=1).mean()
```

Results for DINOv2



The Vision Transformer (ViT)

Part of the motivation for DINOv1 (Meta, April 2021) was to demonstrate the power of visual transformers (Google Brain, October 2020).

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, Alexey Dosovitskiy et al. (Oct 2020, Google Brain)

ViT

An image I with height X, width Y and $C = 3$ color channels has shape $I[X, Y, C]$. In a ViT this is processed by a linear layer with stride s resulting in a layer L with shape $L[X/s, Y/s, N]$. This processing is equivalent to a Conv2d layer with stride s but without an activation function (there is no ReLU).

for $x, y, n, \Delta x, \Delta y, c \quad L[x, y, n]$ += $W[n, \Delta x, \Delta y, c]I[sx+\Delta x, sy+\Delta Y, c]$

In the original ViT paper the stride s is taken to be 16 (hence the title). DINOv2 uses $s = 14$.

Presumably for each neuron n the tensor $W[n, \Delta X, \Delta Y, C]$ converges on some Haar wavelet.

ViT

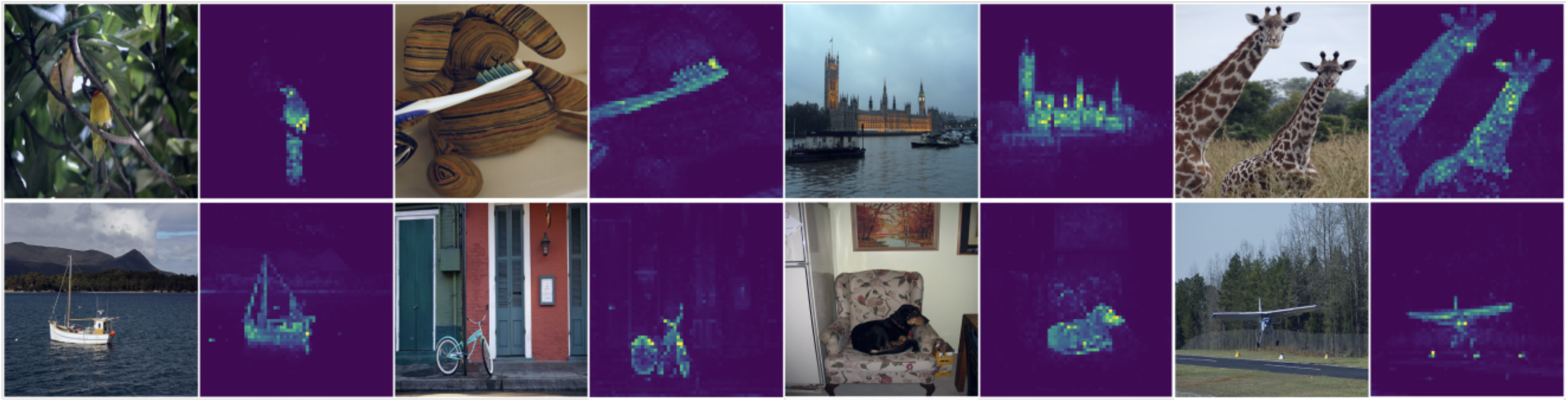
We now have a tensor $L(X, Y, N)$. We treat this as $X \times Y$ positions with an "embedding vector" at each position.

At each position the embedding vector is concatenated with a position embedding. The position embeddings can be trained so there is no need to make a distinction between "linear" vs. "spatial" position embeddings (this would presumably not be true for relative position embeddings.)

An additoinal non-image position is added to represent the image as a whole. The vector computed for the additonal position, which I will write as $\mathrm{enc}_x(x)$, is interpreted as the feature vector for image x.

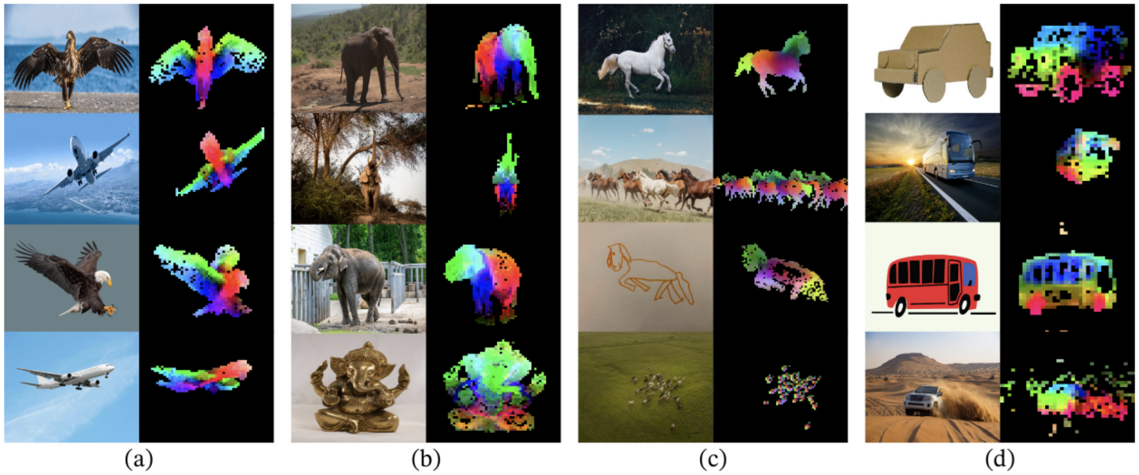
In DINOv2 each component $\mathrm{enc}_x(x)[k]$ of the image embedding is interpreted as the score of label k.

Attention Gives Segmentation in DINOv1



This is showing the attention that the whole image position pays to the other image positions.

PCA Gives Part Matching in DINOv2



For each column we pool the top transformer layer vectors across the four images of the column and do PCA on that pool of vectors.

Each image is segmented by thresholding the largest PCA component.

The three colors in the image then correspond to the strength of the three largest PCA components. We then get part matching across different images.

END