Self-Supervised Learning

Self-Supervised Learning

Supervision Signal from the Data Itself

Why Self-Supervised Learning?

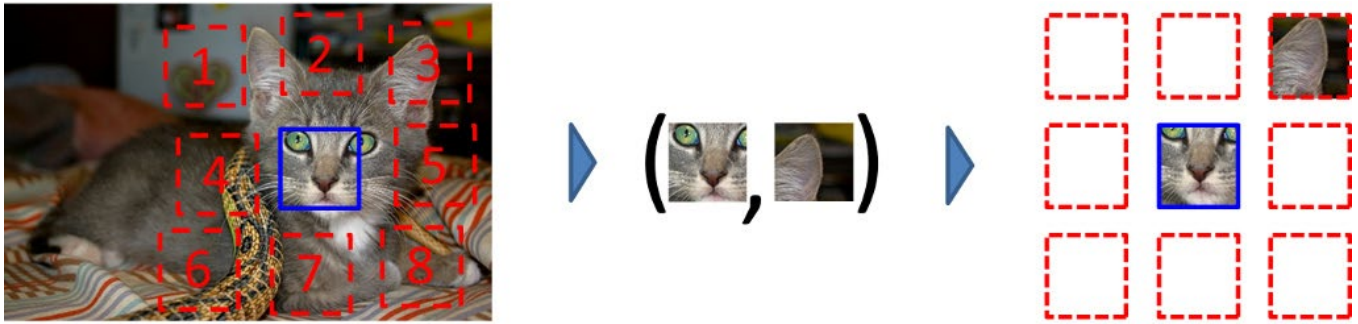
- Originally researchers learnt neural networks features (CNNs) by supervision from annotated data (e.g., classification on ImageNet).
- But there is a limited amount of supervised data and an enormous amount of unsupervised data. Self-supervised learning can use a lot more data. (LLMs).
- Researchers noticed that neural network features tended to be similar for different visual tasks – e.g., classification and semantic segmentation. So maybe there is a set of neural features common for all tasks which can be learnt without supervision.
- Self-supervised learning is attractive for researchers who are influenced by human vision.

Self-Supervised Learning: Backbone and Heads.

- After features have been learnt by self-supervised learning then they can be applied to *downstream tasks* – e.g., object classification and semantic segmentation – by training simple *decision heads* which have few parameters and can be trained with a small amount of data.
- Recall that neural networks have a *backbone,* which extracts image features, and a *head* which performs the task. So self-supervised learns the backbone.

- Self-supervised learning works by defining proxy tasks, which do not require supervision, and then training neural networks to perform them. Researchers explored several types of proxy tasks.

Predicting Context



Predicting Orientation

Y = 1

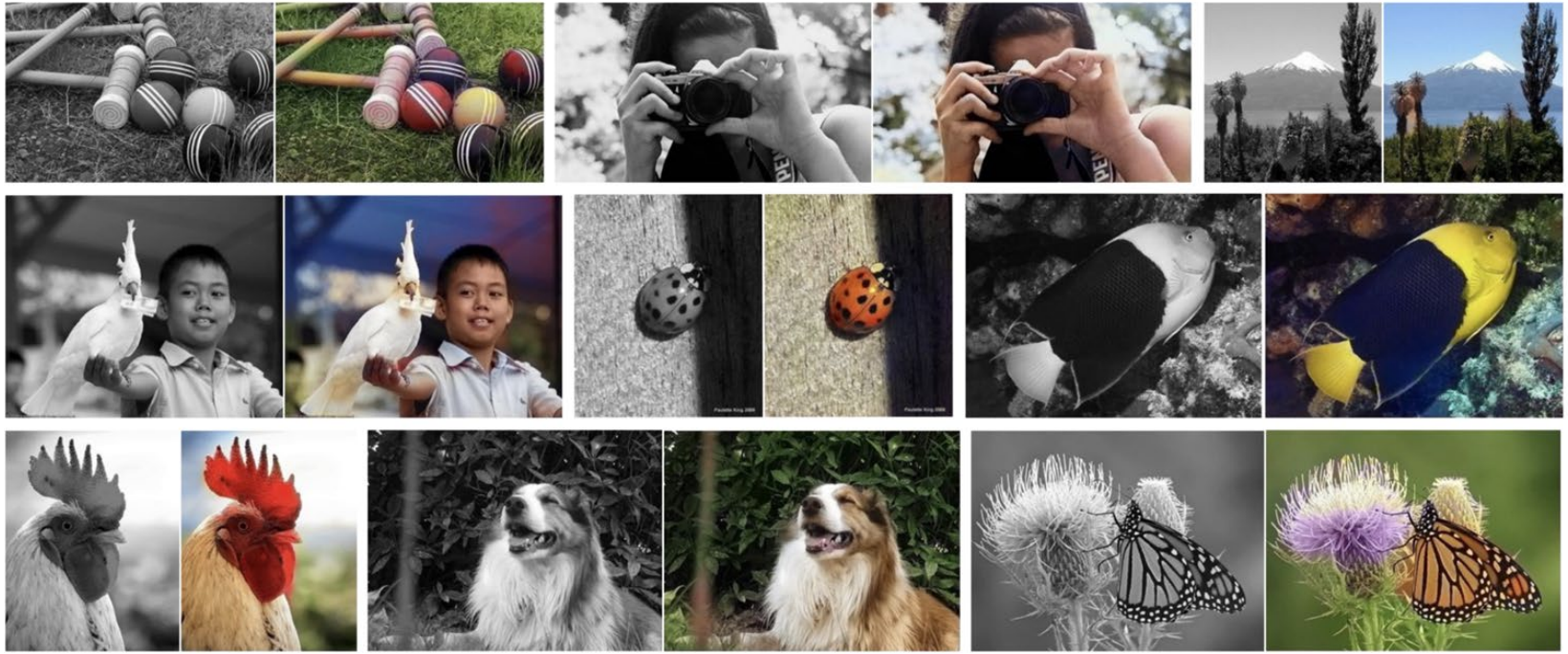
Y = 2

Y = 3

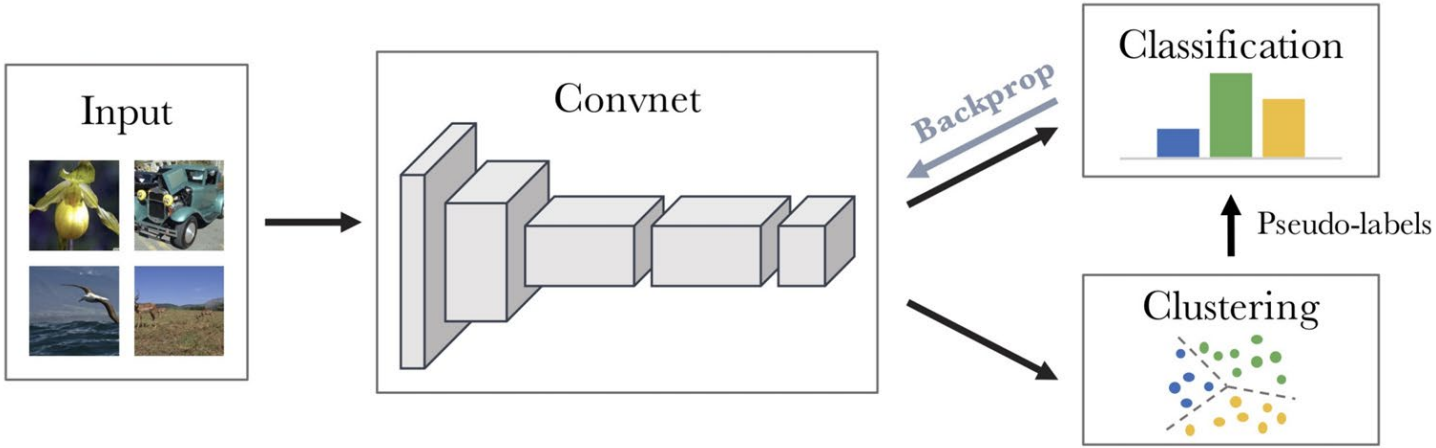
Y = 4

Predicting Patch Organization

Predicting Color



Predicting (Unsupervised) Cluster ID



...Unsupervised Learning motivated contrastive learning

- K-Means cluster → Views from one image as one cluster

Instance Recognition

or

Contrastive Learning

Contrastive Learning was the first popular self-supervised learning technique. It is also useful for supervised learning, see earlier in the course.

Contrastive Learning: Query and Keys. +ve and –ve.

…from the same instance

query

positive key

randomly sampled…

negative keys

Contrastive Learning

$$-\sum_{i=1}^{2n}\frac{1}{2|N_i|-1}\sum_{j\in N(y_i),j\neq i}\log\frac{\exp(\mathbf{z}_i\cdot\mathbf{z}_j/\tau)}{\sum_{k\in I,k\neq i}\exp(\mathbf{z}_i\cdot\mathbf{z}_k/\tau)}$$

...from the same instance

query

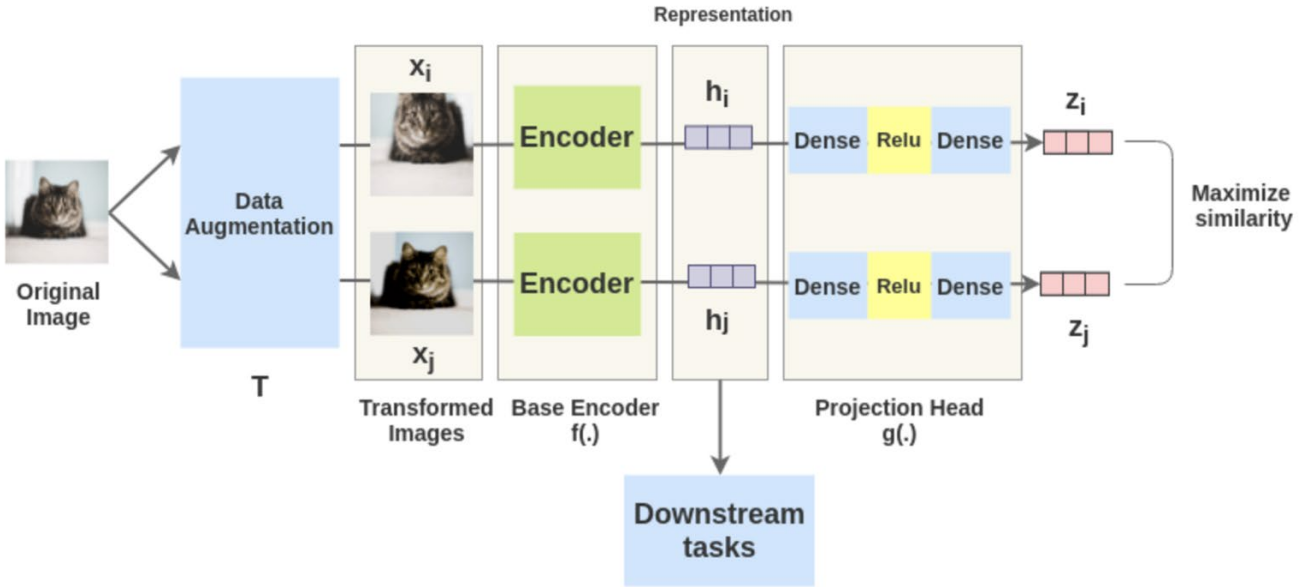
The z's are neural network features.
They are functions of the image and network weights, which are learnt by minimizing the loss function.
z_i is the query, z_j are positive keys
z_k are negative keys.
The learning selects network weights to make the query features similar to the positive keys and different to negative keys

1 0 0 0 0 0

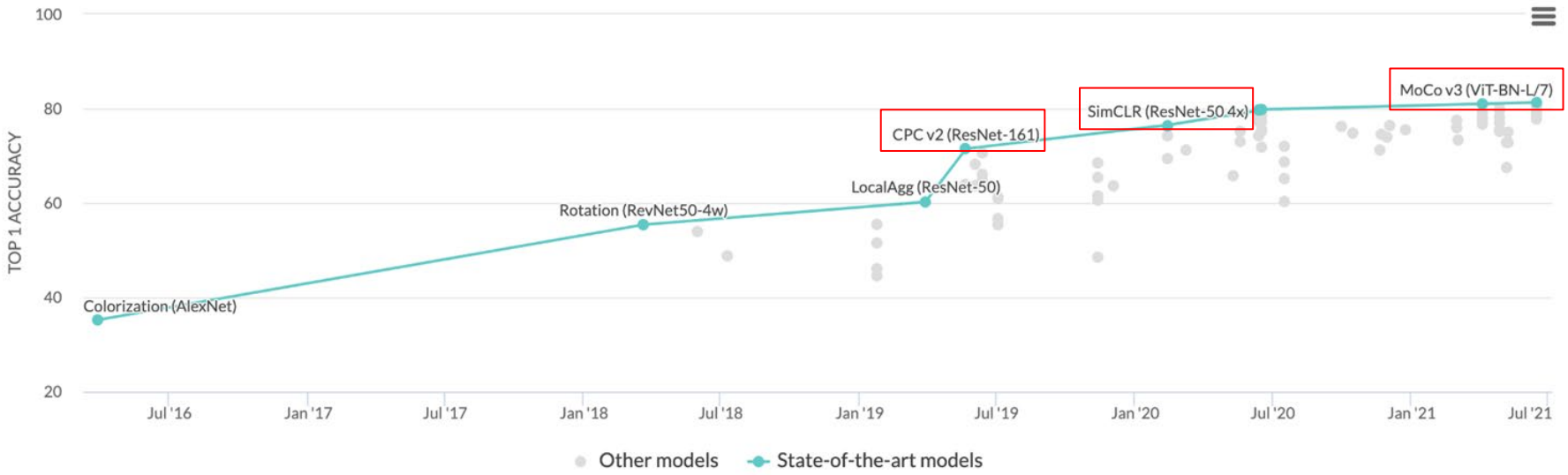
positive key

negative keys (randomly sampled)

Contrastive Learning: maximize similarity queries & +ve keys

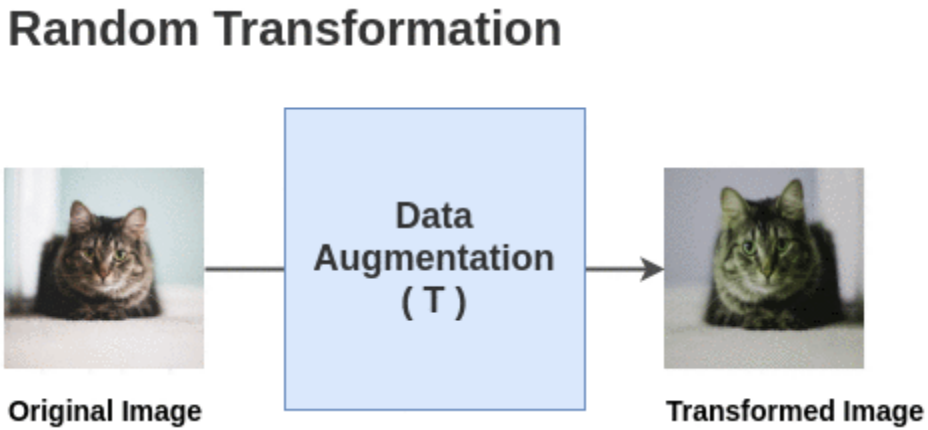


Contrastive Learning was very popular before 2022



ImageNet-1K Linear Evaluation accuracy

Key: The +ve keys are obtained by Augmentation



Self-Supervised helps with attention

Supervised Transformer is trained for object classification. The attention maps give an approximate segmentation of the target object.
A self-supervised transformer, using contrastive learning, gives tighter attention maps.

Attention Map from Vision Transformer

Mathilde Caron, et al., Emerging Properties in Self-Supervised Vision Transformers, arXiv 2021

Self-Supervised Learning by Contrastive

- The basic finding is that simple heads based on neural network features learnt by unsupervised (contrastive learning) have roughly the same performance as fully supervised methods, but require less supervised data.
- The self-supervised features have attractive properties which make them better than supervised methods for image reconstruction and for grouping image regions by attention.

Why do we need self-supervised learning?

- Because labels are expensive.

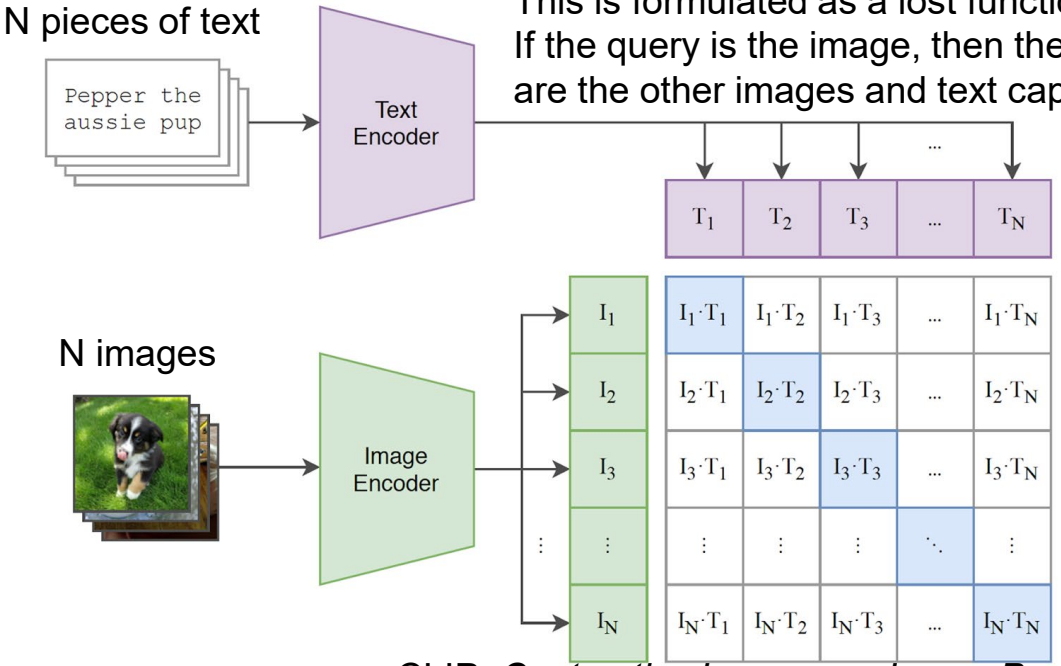
- *Because labels are limited.*

How are labels limited?

- The information in one image is *richer* than its label.
- Label could be *ambiguous, biased, incomplete*...

Vision & Language: Example of Contrastive Learning

The trained data consists of paired images and text captions.
The goal is to find text and image features which are similar for each pair and differ for the image and text features of the other pairs.
This is formulated as a lost function as before.
If the query is the image, then the +key is the text and the –ve keys are the other images and text captions.



CLIP is very popular for vision-language tasks.
It is very low tech and good for Initializing more advanced methods.

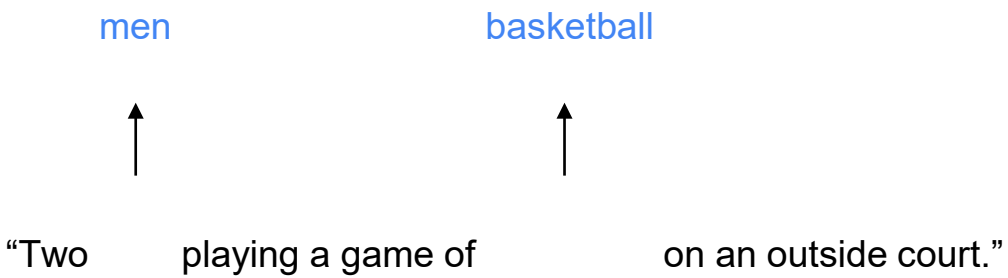
CLIP: *Contrastive Language–Image Pre-training*

Alec Radford, et al., Learning Transferable Visual Models From Natural Language Supervision, arXiv 2021

Masked Image Modeling.

This was inspired by NLP where it was used for learning Large Language Models (LLMs) but is being replaced by auto-regressive methods.

Masked Language Modeling



Devlin *et al.*, BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding

Masked Image Modeling

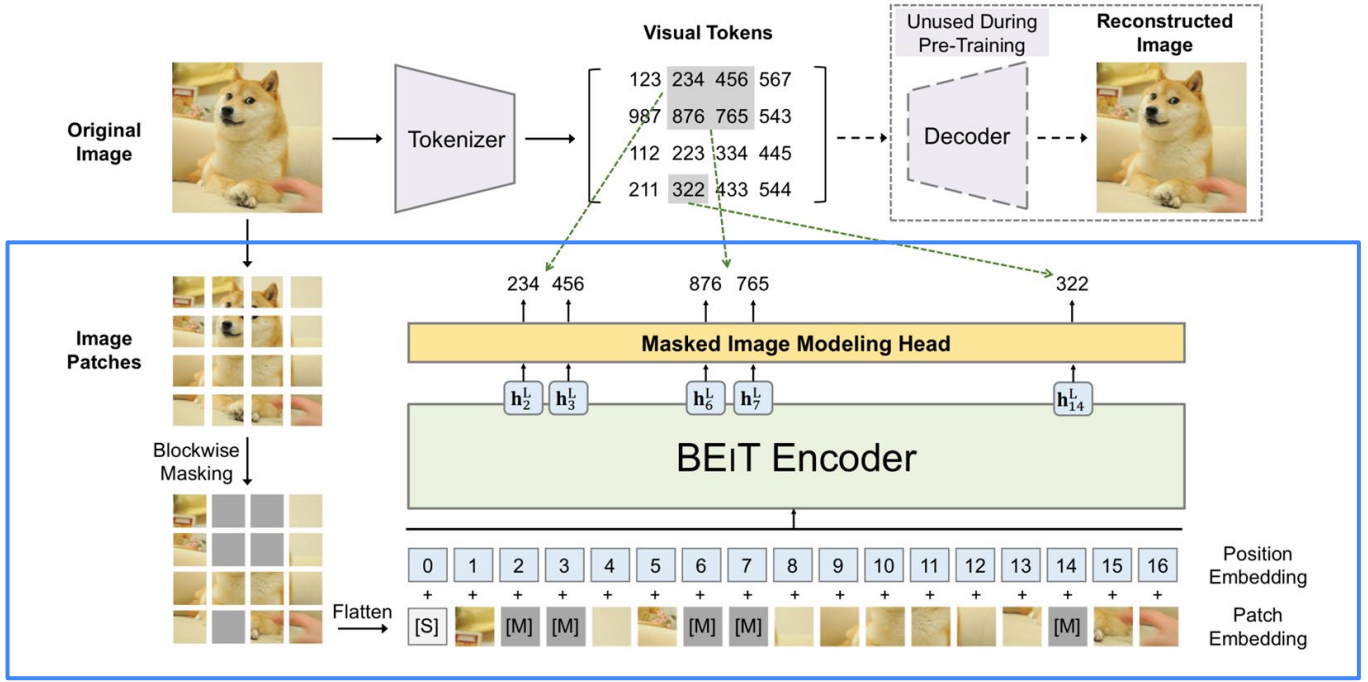
We humans have the ability to first recognize the masked object and then infill it.

Language vs. Vision

- Language
  - sparse, discrete, semantic-rich
  - natural word tokens

- Vision
  - dense, continuous, high-dimensional
  - mimicking language: visual words/codebook?

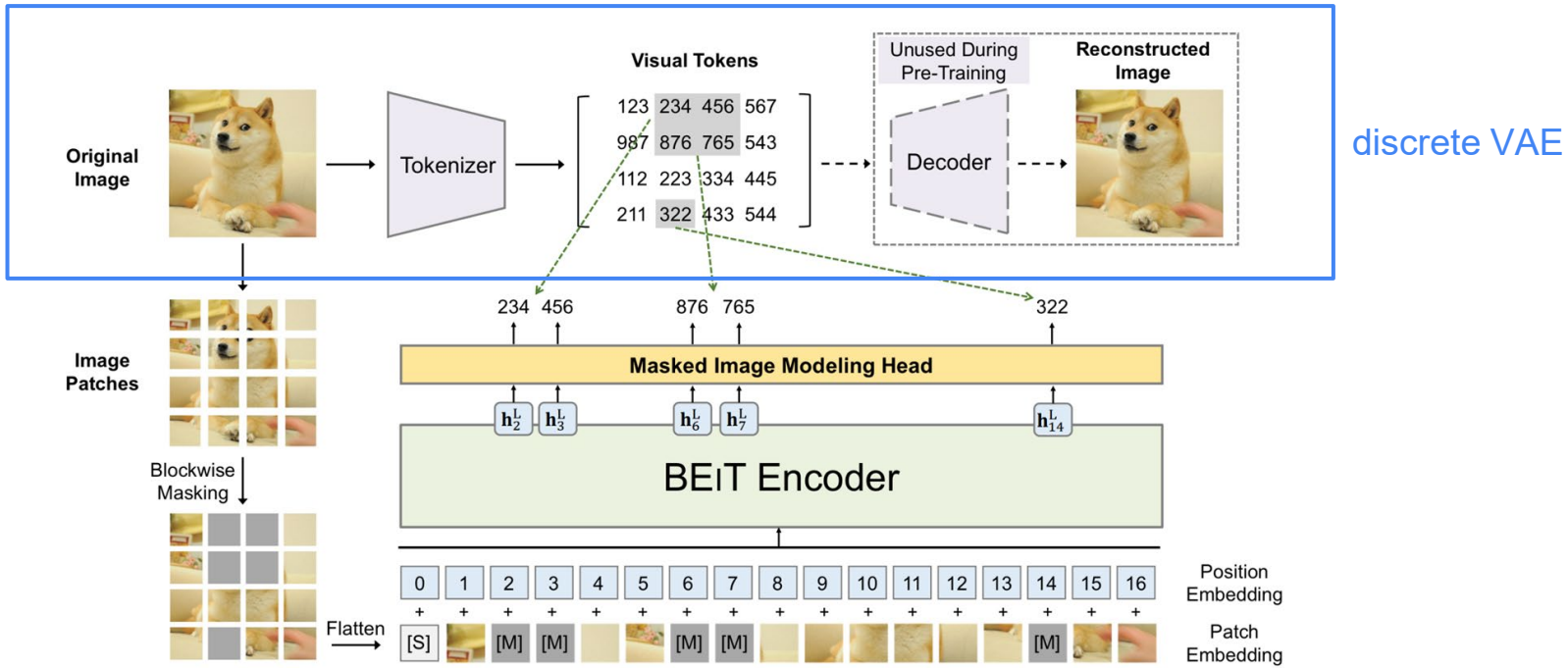
BEiT: A Transformer with Masked Input



Bao *et al.*, BEIT: BERT Pre-Training of Image Transformers, NeurIPS 2022

BEiT: Predicting Visual Tokens

The visual tokens – discrete – are learnt by variational auto-encoders separately from the MIM. The visual tokens are a discrete dictionary of image patches.



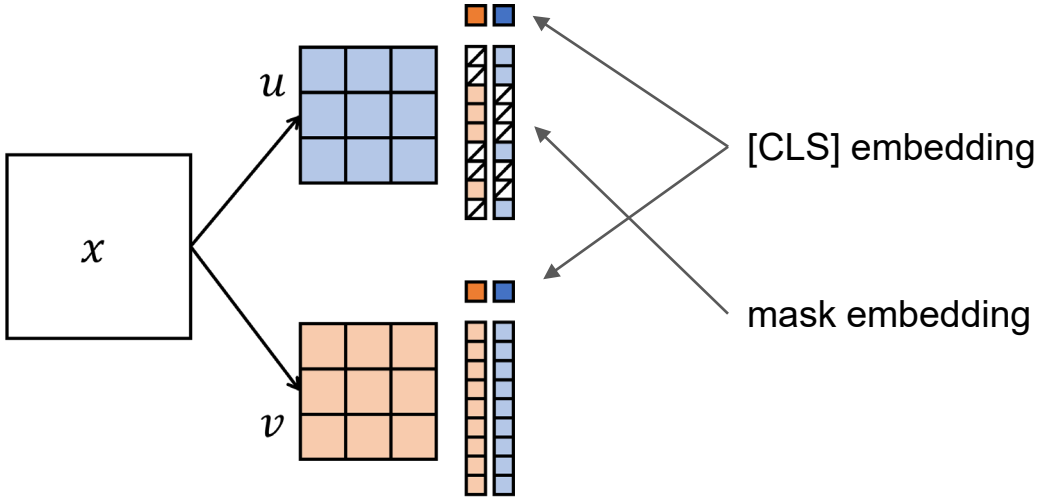
Bao *et al.*, BEIT: BERT Pre-Training of Image Transformers, NeurIPS 2022

iBOT: Image BERT Pre-Training with Online Tokenizer

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, Tao Kong

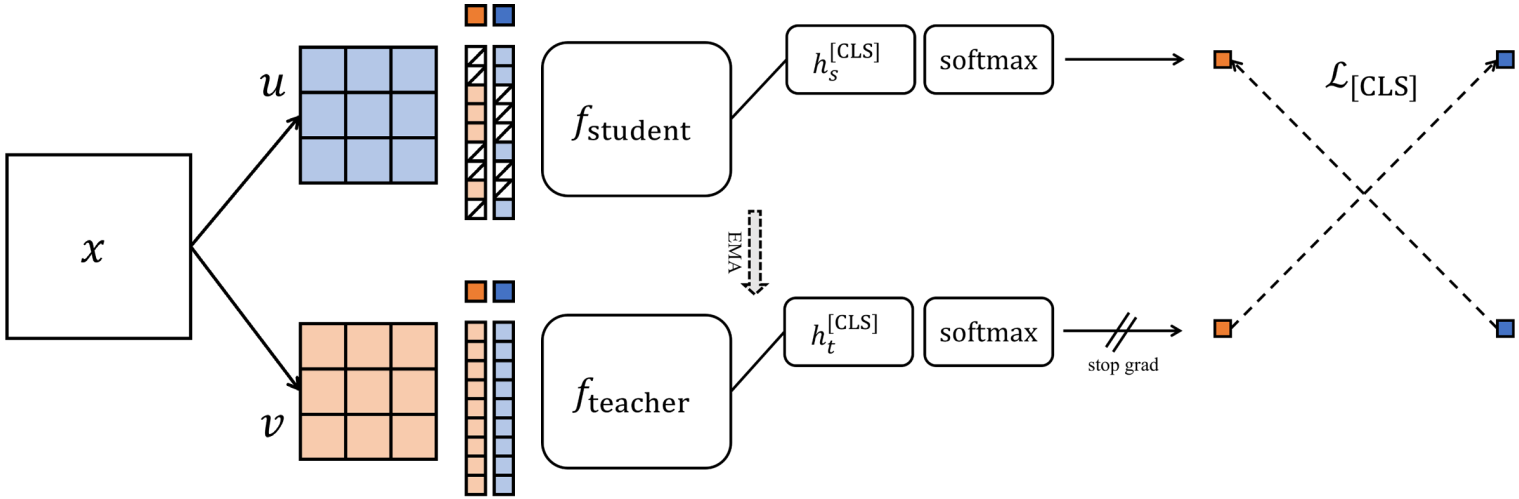
ByteDance, Johns Hopkins University
Shanghai Jiao Tong University, UC Santa Cruz
ICLR 2022

Masked and Multi-Viewed Input



Each image x is first augmented to two views, each of which has a masked version.

Image Tokenizer

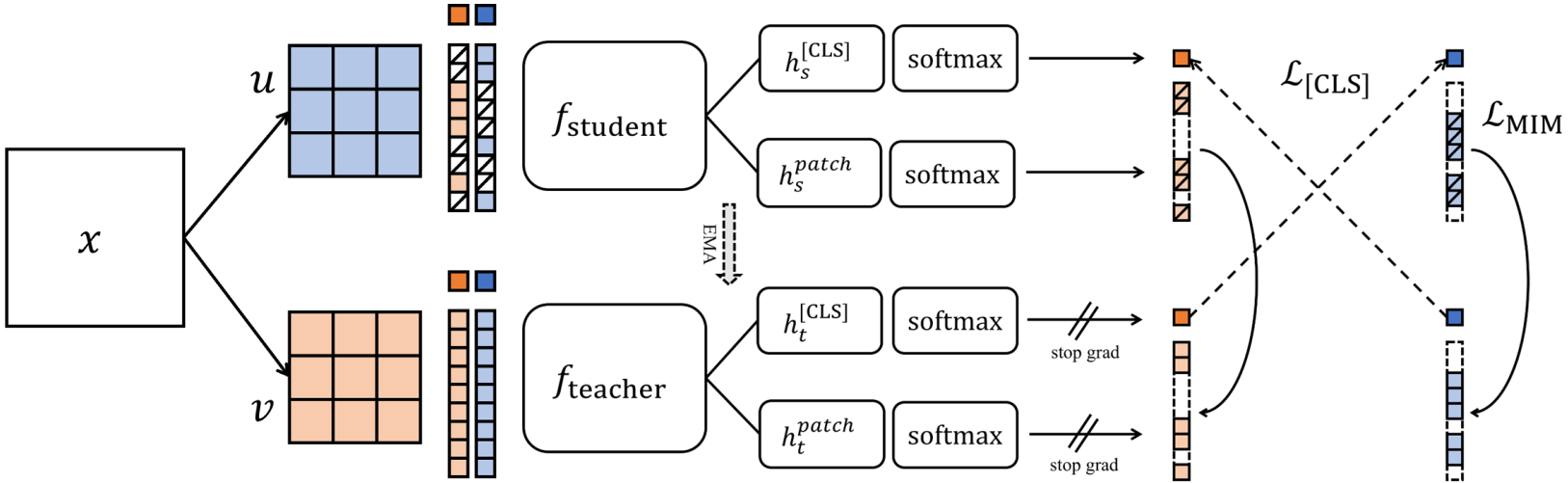


[CLS] embeddings are transformed to visual tokens by $h^{[\text{CLS}]}$ followed by softmax.

Caron *et al.*, Emerging Properties in Self-Supervised Vision Transformers

Masked Image Modeling

Although this model is effective it has black box aspects and it is hard to give intuitive reasons why they are effective.



Patch embeddings of masked areas predict their corresponding unmasked version.

Linear Probing: iBOT is very effective with simple heads.

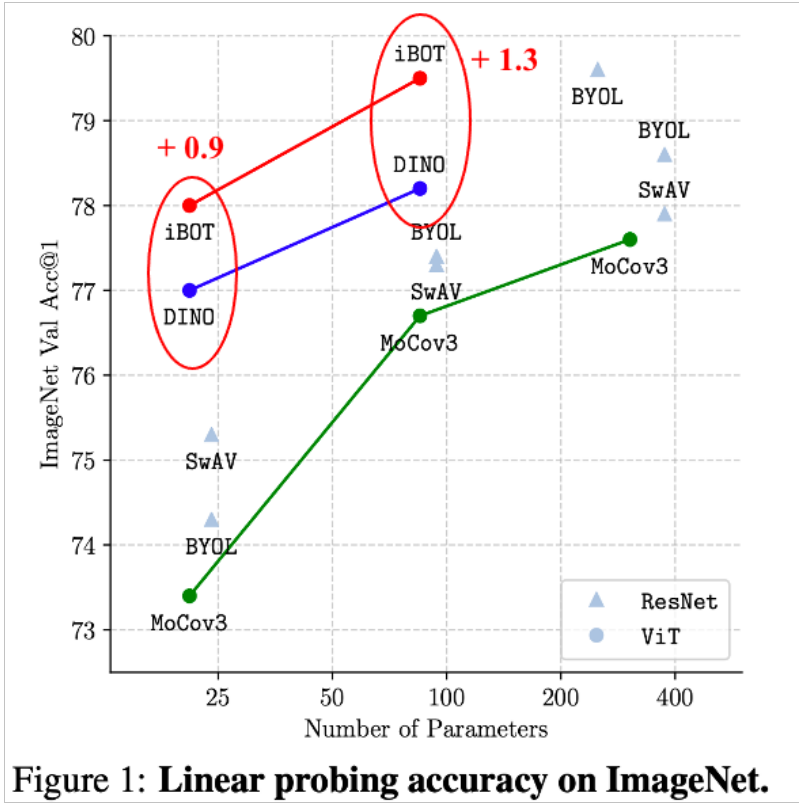


Figure 1: **Linear probing accuracy on ImageNet.**

Fine-Tuning

Method	Arch.	Epo.[1]	Acc.
Rand.	ViT-S/16	-	79.9
MoCov3	ViT-S/16	600	81.4
BEiT	ViT-S/16	800	81.4
DINO	ViT-S/16	3200	82.0
iBOT	ViT-S/16	3200	**82.3**
Rand.	ViT-B/16	-	81.8
MoCov3	ViT-B/16	600	83.2
BEiT	ViT-B/16	800	83.4
MAE	ViT-B/16	1600	83.6
DINO	ViT-B/16	1600	83.6
iBOT	ViT-B/16	1600	**83.8**

ImageNet-1K pre-training and fine-tuning

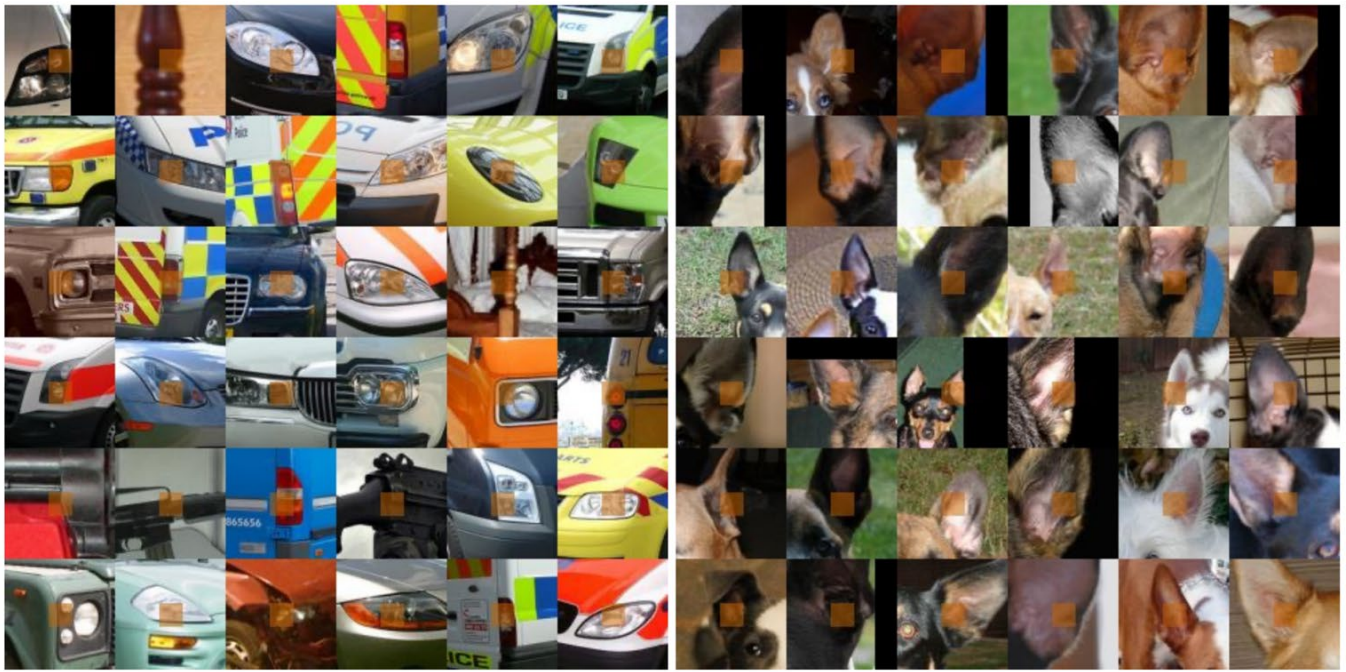
Transferring to Downstream Tasks

Method	Det.	ISeg.	Seg.†	Seg.
	AP^b	AP^m	mIoU	mIoU
Sup.	49.8	43.2	35.4	46.6
BEiT	50.1	43.5	27.4	45.8
DINO	50.1	43.4	34.5	46.8
MAE	-	-	-	48.1
iBOT	**51.2**	**44.2**	**38.3**	**50.0**

ViT-B, pre-trained by IN-1K
COCO Detection and Instance Segmentation,
ADE20K Semantic Segmentation

Image-Level Visual Tokens: Qualitatively Interpretable

Patch-Level Visual Tokens: Qualitatively Interpretable



Ablations: Sharing Heads

Sharing [CLS] and Patch Heads

Method	$\mathcal{L}_{\text{MIM}}$	$\mathcal{L}_{[\text{CLS}]}$	SH	k-NN	Lin.	Fin.
iBOT	✓	✓	✓	69.1	74.2	81.5
	✓	✓	✗	69.0	73.8	81.5
	✓	✗	-	9.5	29.8	79.4
	○	✗	-	44.3	60.0	81.7
BEiT	△	✗	-	6.9	23.5	81.4
DINO	✗	✓	-	67.9	72.5	80.6
BEiT + DINO	△	✓	-	48.0	62.7	81.2

○: standalone DINO (w/o mcrop, 300-epoch)
△: pre-trained DALL-E encoder

ViT-S/16, ImageNet-1K

Ablations: w/o [CLS] loss

Method	$\mathcal{L}_{\text{MIM}}$	$\mathcal{L}_{[\text{CLS}]}$	SH	k-NN	Lin.	Fin.
iBOT	✓	✓	✓	69.1	74.2	81.5
	✓	✓	✗	69.0	73.8	81.5
	✓	✗	-	9.5	29.8	79.4
	○	✗	-	44.3	60.0	81.7
BEiT	△	✗	-	6.9	23.5	81.4
DINO	✗	✓	-	67.9	72.5	80.6
BEiT + DINO	△	✓	-	48.0	62.7	81.2

○: standalone DINO (w/o mcrop, 300-epoch)
△: pre-trained DALL-E encoder

ViT-S/16, ImageNet-1K

Ablations: standalone DINO

Method	$\mathcal{L}_{\text{MIM}}$	$\mathcal{L}_{\texttt{[CLS]}}$	SH	k-NN	Lin.	Fin.
iBOT	✓	✓	✓	69.1	74.2	81.5
	✓	✓	✗	69.0	73.8	81.5
	✓	✗	-	9.5	29.8	79.4
	○	✗	-	44.3	60.0	81.7
BEiT	△	✗	-	6.9	23.5	81.4
DINO	✗	✓	-	67.9	72.5	80.6
BEiT + DINO	△	✓	-	48.0	62.7	81.2

○: standalone DINO (w/o mcrop, 300-epoch)
△: pre-trained DALL-E encoder

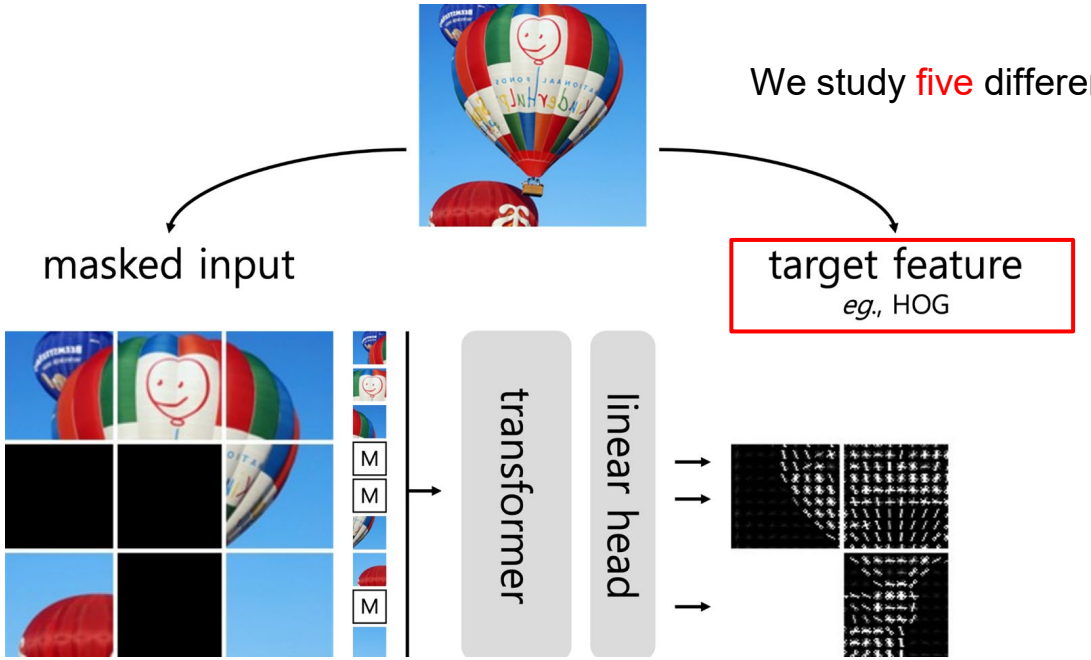
ViT-S/16, ImageNet-1K

Masked Feature Prediction for Self-Supervised Visual Pre-Training

Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, Christoph Feichtenhofer

Facebook AI Research, Johns Hopkins University
CVPR 2023

Masked Feature Prediction



regress the masked patches

Setting: Pre-Training + Fine-Tuning

feature type	one-stage	variant	arch.	param.	epoch†	top-1
scratch	-	DeiT [84]	-	-	-	81.8
pixel colors	✓	RGB	-	-	-	82.5
image descriptor	✓	HOG [22]	-	-	-	**83.6**
dVAE token	✗	DALL-E [73]	dVAE	54	1199	82.8
unsupervised feature	✗	MoCo v2 [16]	ResNet50	23	800	83.6
unsupervised feature	✗	MoCo v3 [18]	ViT-B	85	600	83.9
unsupervised feature	✗	DINO [9]	ViT-B	85	1535	**84.0**
supervised feature	✗	pytorch [67]	ResNet50	23	90	82.6
supervised feature	✗	DeiT [84]	ViT-B	85	300	81.9
pseudo-label	✗	Token Labeling [50]	NFNet-F6	438	360	78.8

ViT-B, ImageNet val accuracy

Feature #1: pixel colors

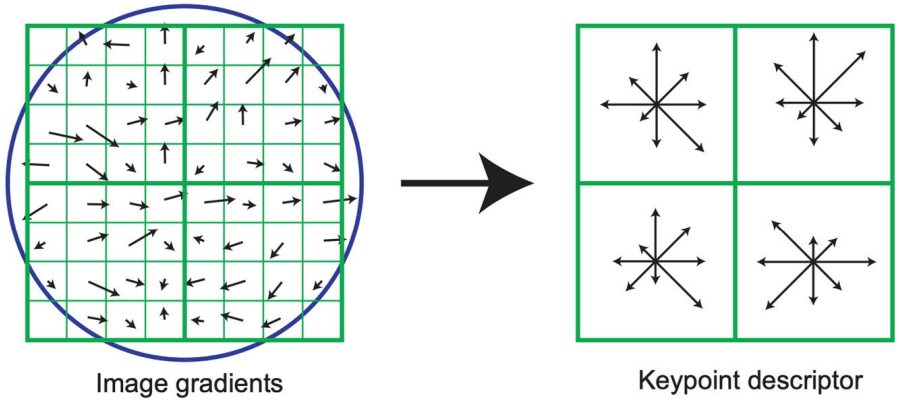
- RGB raw pixels
  - A small gain
  - trivial local statistics and high-frequency details

feature type	one-stage	variant	arch.	param.	epoch†	top-1
scratch	-	DeiT [84]	-	-	-	81.8
pixel colors	✓	RGB	-	-	-	82.5

+0.7

Feature #2: HOG

- Histogram of Oriented Gradients
  - popular in 2000s
  - invariance to geometry and photometric change (to some extent)
  - fast to compute with pytorch and GPU



from SIFT paper

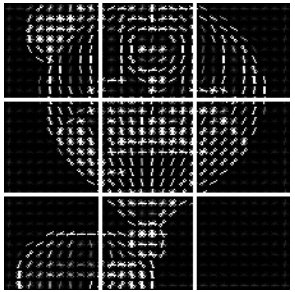
Feature #2: HOG

Input image

Histogram of Oriented Gradients

from scikit-image

Feature #2: HOG

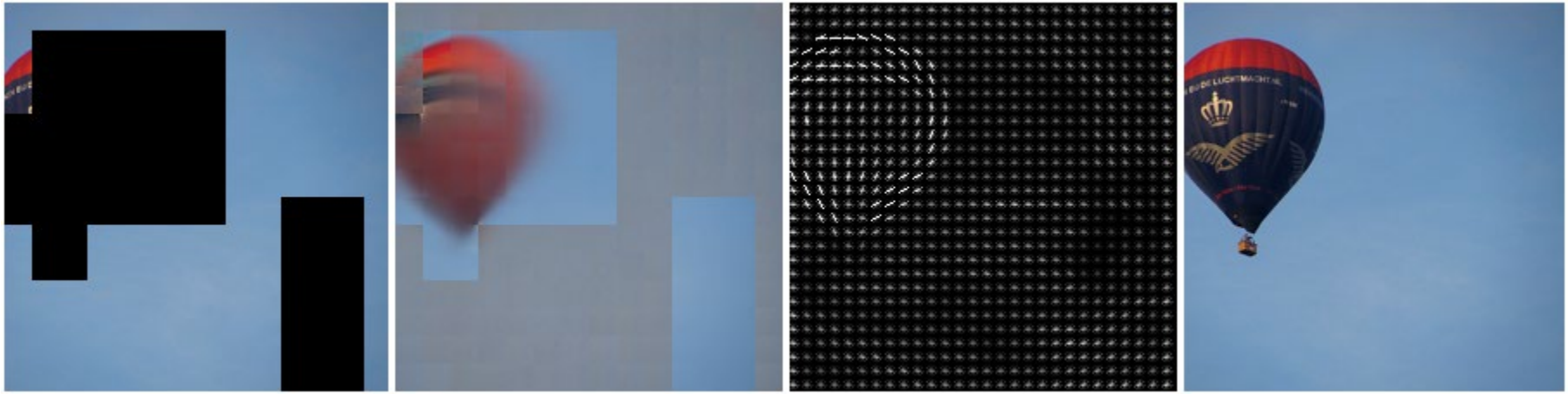


- Histogram of Oriented Gradients
  - invariance helps!

MFP feature type	one-stage	variant	arch.	param.	epoch†	top-1
scratch	-	DeiT [81]	-	-	-	81.8
pixel colors	✓	RGB	-	-	-	82.5
image descriptor	✓	HOG [22]	-	-	-	**83.6**

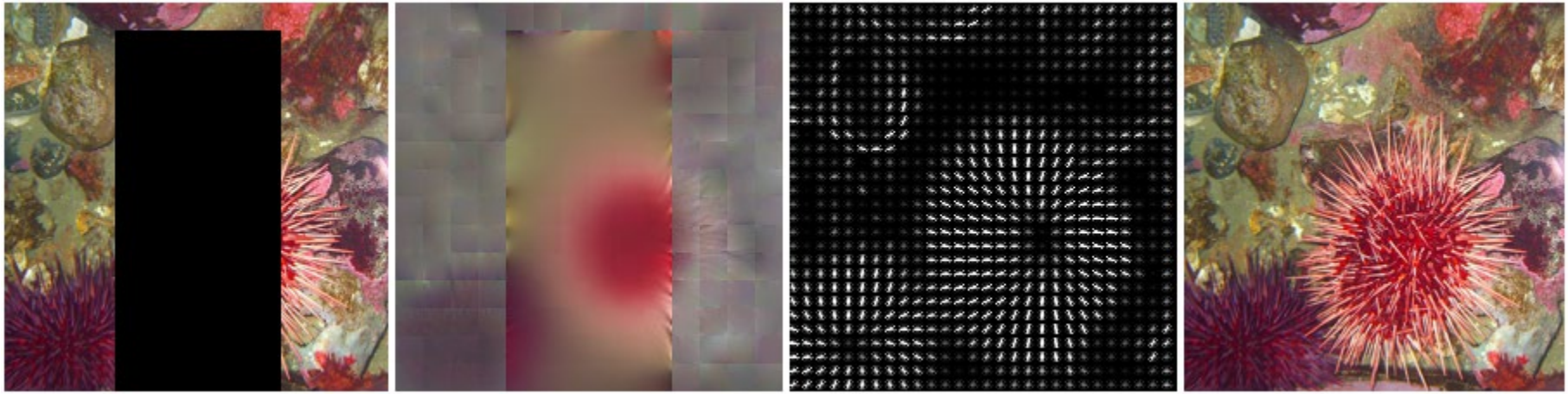
+1.8

Pixel *vs.* HOG: Color Ambiguity



pixel: large loss penalty because of unmatched color

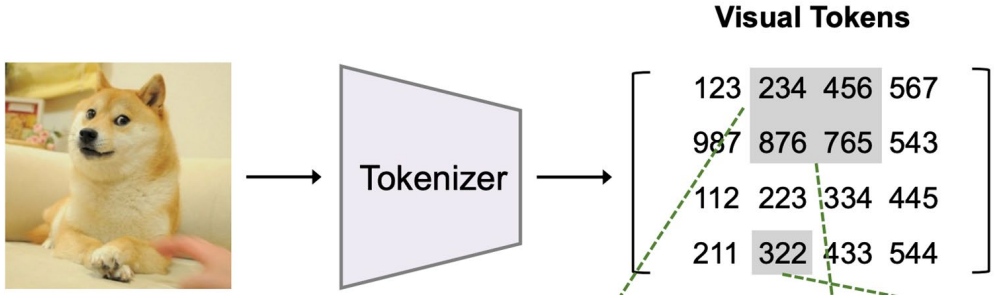
Pixel *vs.* HOG: Texture Ambiguity



HOG: captures major edge directions

Feature #3: token

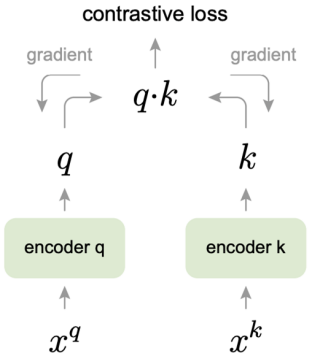
- discrete VAE token
  - patch clustering
  - BEiT



feature type	one-stage	variant	arch.	param.	epoch†	top-1
scratch	-	DeiT [84]	-	-	-	81.8
pixel colors	✓	RGB	-	-	-	82.5
image descriptor	✓	HOG [22]	-	-	-	**83.6**
dVAE token	✗	DALL-E [73]	dVAE	54	1199	82.8

Feature #4: deep features

- unsupervised deep features
  - contrastive unsupervised methods
  - work better than others



feature type	one-stage	variant	arch.	param.	epoch†	top-1
scratch	-	DeiT [84]	-	-	-	81.8
pixel colors	✓	RGB	-	-	-	82.5
image descriptor	✓	HOG [22]	-	-	-	**83.6**
dVAE token	✗	DALL-E [73]	dVAE	54	1199	82.8
unsupervised feature	✗	MoCo v2 [16]	ResNet50	23	800	83.6
unsupervised feature	✗	MoCo v3 [18]	ViT-B	85	600	83.9
unsupervised feature	✗	DINO [9]	ViT-B	85	1535	**84.0**

+2.2

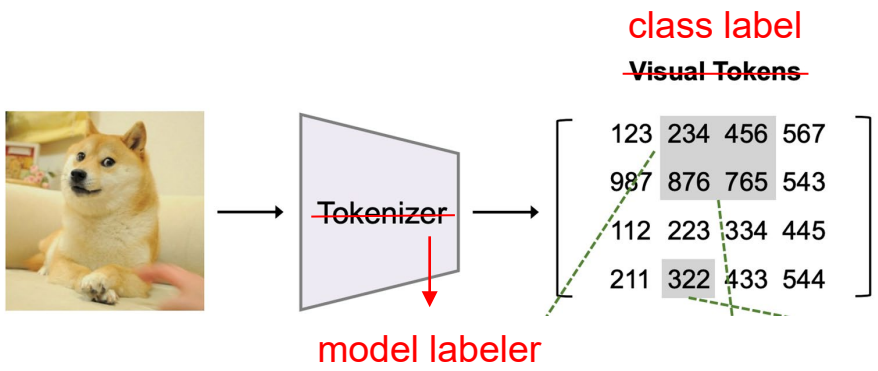
Feature #4: deep features

- supervised deep features
  - more labels, lower top-1
  - ResNet50 helps, ViT-B does not

feature type	one-stage	variant	arch.	param.	epoch†	top-1
scratch	-	DeiT [84]	-	-	-	81.8
pixel colors	✓	RGB	-	-	-	82.5
image descriptor	✓	HOG [22]	-	-	-	**83.6**
dVAE token	✗	DALL-E [73]	dVAE	54	1199	82.8
unsupervised feature	✗	MoCo v2 [16]	ResNet50	23	800	83.6
unsupervised feature	✗	MoCo v3 [18]	ViT-B	85	600	83.9
unsupervised feature	✗	DINO [9]	ViT-B	85	1535	**84.0**
supervised feature	✗	pytorch [67]	ResNet50	23	90	82.6
supervised feature	✗	DeiT [84]	ViT-B	85	300	81.9

Feature #5: pseudo label

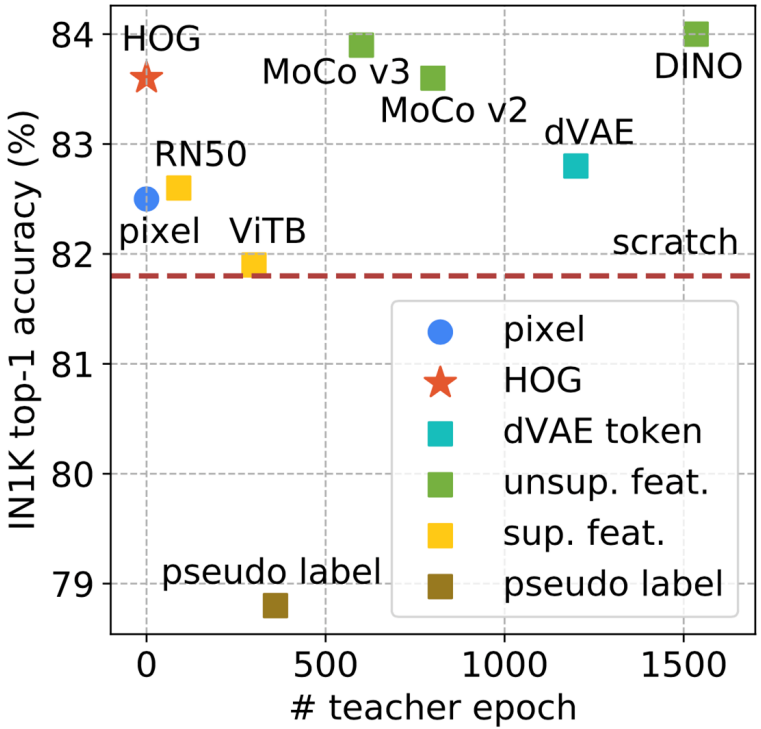
- pseudo class label for each patch
  - labeled by a 86.5% supervised model
  - but results in a huge drop



feature type	one-stage	variant	arch.	param.	epoch†	top-1
scratch	-	DeiT [84]	-	-	-	81.8
pixel colors	✓	RGB	-	-	-	82.5
image descriptor	✓	HOG [22]	-	-	-	**83.6**
dVAE token	✗	DALL-E [73]	dVAE	54	1199	82.8
unsupervised feature	✗	MoCo v2 [16]	ResNet50	23	800	83.6
unsupervised feature	✗	MoCo v3 [18]	ViT-B	85	600	83.9
unsupervised feature	✗	DINO [9]	ViT-B	85	1535	**84.0**
supervised feature	✗	pytorch [67]	ResNet50	23	90	82.6
supervised feature	✗	DeiT [84]	ViT-B	85	300	81.9
pseudo-label	✗	Token Labeling [50]	NFNet-F6	438	360	78.8

-3.0

Masked Feature Prediction



ImageNet-1K Fine-Tuning

pre-train	extra data	extra model	ViT-B	ViT-L
scratch [84]	-	-	81.8	81.5
supervised$_{384}$ [27]	IN-21K	-	84.0	85.2
MoCo v3 [18]	-	momentum ViT	83.2	84.1
DINO [9]	-	momentum ViT	82.8	-
BEiT [2]	DALL-E	dVAE	83.2	85.2
MaskFeat (w/ HOG)	-	-	**84.0**	**85.7**

+4.2

ImageNet val accuracy

Action Classification

- Kinetics-400
  - rather small (~0.2 million video clips)
  - commonly using ImageNet pre-training or even larger image dataset

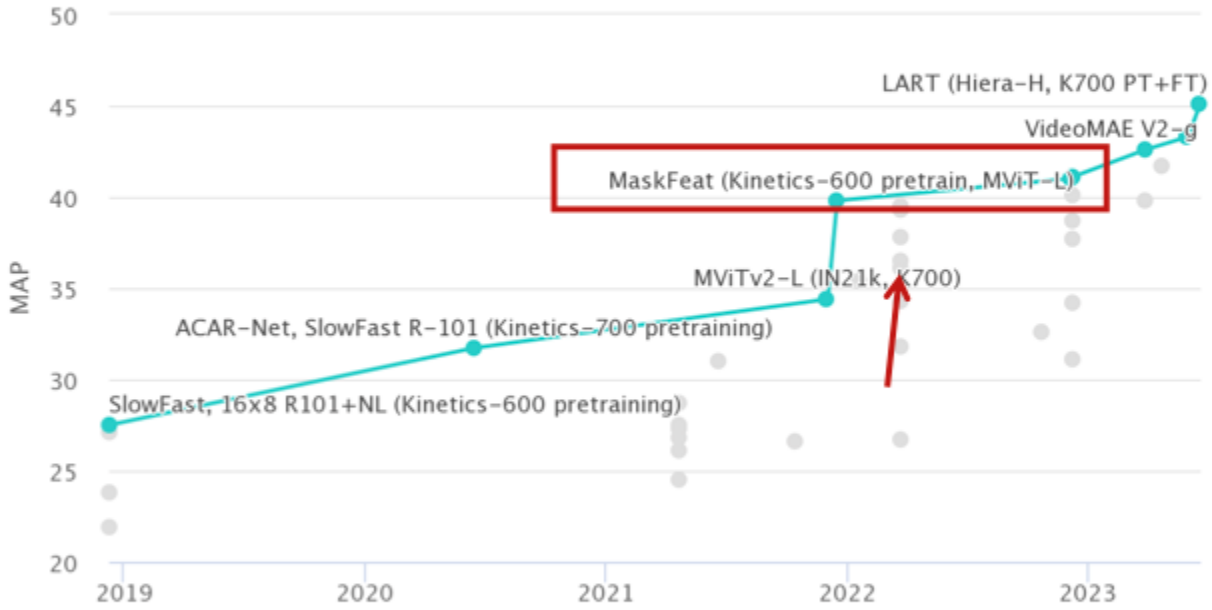
model	pre-train	top-1	top-5	FLOPs×views	Param
Two-Stream I3D [12]	-	71.6	90.0	216 × NA	25
ip-CSN-152 [85]	-	77.8	92.8	109×3×10	33
SlowFast 16×8 +NL [33]	-	79.8	93.9	234×3×10	60
X3D-XL [32]	-	79.1	93.9	48×3×10	11
MoViNet-A6 [53]	-	81.5	95.3	386×1×1	31
MViTv1-B, 64×3 [31]	-	81.2	95.1	455×3×3	37
Swin-B, 32×2 [59]	Sup., IN-21K	82.7	95.5	282×3×4	88
ViT-B-TimeSformer [4]	Sup., IN-21K	80.7	94.7	2380×3×1	121
Swin-L, 32×2 [59]	Sup., IN-21K	83.1	95.9	604×3×4	197
ViViT-L [1]	Sup., JFT-300M	83.5	94.3	3980×3×1	308
Swin-L↑384, 32×2 [59]	Sup., IN-21K	84.9	96.7	2107×5×10	200
ViViT-H [1]	Sup., JFT-300M	84.9	95.8	3981×3×4	654
TokenLearner [75]	Sup., JFT-300M	85.4	N/A	4076×3×4	450
Florence↑384 [95]	Text, FLD-900M	86.5	97.3	N/A×3×4	647
SwinV2-G↑384 [58]	MIM + Sup. IN-21K+Ext-70M	86.8	N/A	N/A×5×4	3000

Action Classification

- MaskFeat
  - largely helps over scratch baseline
  - better than IN-21K pre-training

model	pre-train	top-1	top-5	FLOPs×views	Param
Two-Stream I3D [12]	-	71.6	90.0	216 × NA	25
ip-CSN-152 [85]	-	77.8	92.8	109×3×10	33
SlowFast 16×8 +NL [33]	-	79.8	93.9	234×3×10	60
X3D-XL [32]	-	79.1	93.9	48×3×10	11
MoViNet-A6 [53]	-	81.5	95.3	386×1×1	31
MViTv1-B, 64×3 [31]	-	81.2	95.1	455×3×3	37
Swin-B, 32×2 [59]	Sup., IN-21K	82.7	95.5	282×3×4	88
ViT-B-TimeSformer [4]	Sup., IN-21K	80.7	94.7	2380×3×1	121
Swin-L, 32×2 [59]	Sup., IN-21K	83.1	95.9	604×3×4	197
ViViT-L [1]	Sup., JFT-300M	83.5	94.3	3980×3×1	308
Swin-L↑384, 32×2 [59]	Sup., IN-21K	84.9	96.7	2107×5×10	200
ViViT-H [1]	Sup., JFT-300M	84.9	95.8	3981×3×4	654
TokenLearner [75]	Sup., JFT-300M	85.4	N/A	4076×3×4	450
Florence↑384 [95]	Text, FLD-900M	86.5	97.3	N/A×3×4	647
SwinV2-G↑384 [58]	MIM + Sup. IN-21K+Ext-70M	86.8	N/A	N/A×5×4	3000
MViT-S, 16×4 [56]	-	81.1	94.9	71×1×10	36
MViT-S, 16×4 [56]	Sup., IN-21K	82.6	95.3	71×1×10	36
MViT-S, 16×4 [56]	**MaskFeat**, K400	**82.2**	**95.1**	71×1×10	36
MViT-L, 16×4 [56]	-	80.5	94.1	377×1×10	218
MViT-L, 16×4 [56]	Sup., IN-21K	83.5	95.9	377×1×10	218
MViT-L, 16×4 [56]	**MaskFeat**, K400	**84.3**	**96.3**	377×1×10	218
MViT-L, 16×4 [56]	**MaskFeat**, K600	85.1	96.6	377×1×10	218
MViT-L↑312, 32×3 [56]	-	82.2	94.7	2063×3×5	218
MViT-L↑312, 32×3 [56]	Sup., IN-21K	85.3	96.6	2063×3×5	218
MViT-L↑312, 32×3 [56]	**MaskFeat**, K400	**86.3**	**97.1**	2063×3×5	218
MViT-L↑312, 40×3 [56]	**MaskFeat**, K400	86.4	97.1	2828×3×4	218
MViT-L↑352, 40×3 [56]	**MaskFeat**, K400	**86.7**	**97.3**	3790×3×4	218
MViT-L↑352, 40×3 [56]	**MaskFeat**, K600	**87.0**	**97.4**	3790×3×4	218

Results: Action Detection



Action Detection on AVA2.0

Wei et al., Masked Feature Prediction for Self-Supervised Visual Pre-Training, CVPR 2022

Comparison to Contrastive Methods

- Contrastive methods: Invariance to data augmentation
  - but invariance is not always correct
  - heavily rely on “augmentation engineering”
  - complex because of multi-views

- MaskFeat: Image modeling
  - structure inside one image
  - minimal augmentation
  - drive large-scale models
  - what about large-scale data?

Conclusion

- Self-supervised learning shows that it is possible to learn neural network features without supervision.
- These image features are effective for downstream tasks using simple classifier heads.
- They have often attractive properties such as qualitative interpretability but these are not easy to quantifiable.
- Do they scale? Can we get big improvements over standard supervised approaches using large unsupervised datasets? Debateable.
- Can we develop Large Vision Models which are analogous to Large Language Models? Again this debatable. The input to LLMs are words and tokens which are semantically meaningful. This is not so for Vision.