Cornell Bowers C·IS
College of Computing and Information Science

Deep Learning

Week 05: ViTs and CLIP

Thanks to

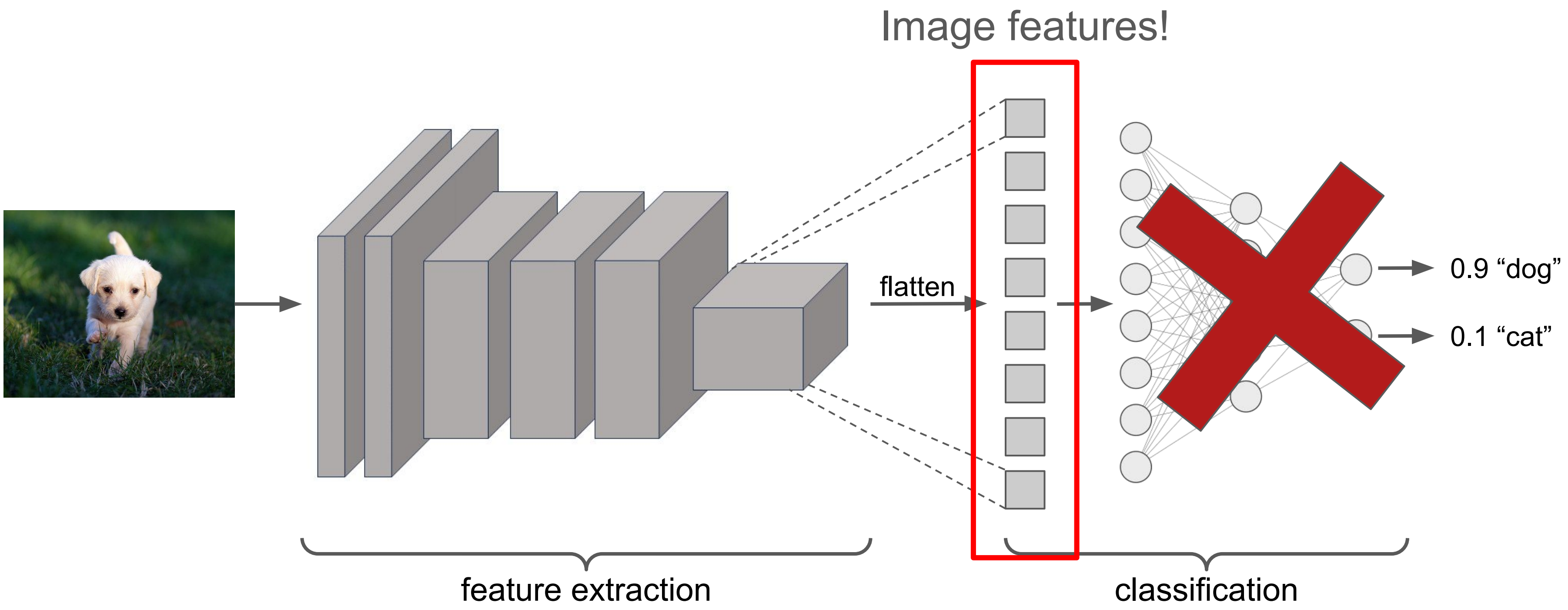
Varsha Kishore
Justin Lovelace
Anissa Dallmann
Dylan Van Bramer

Logistics

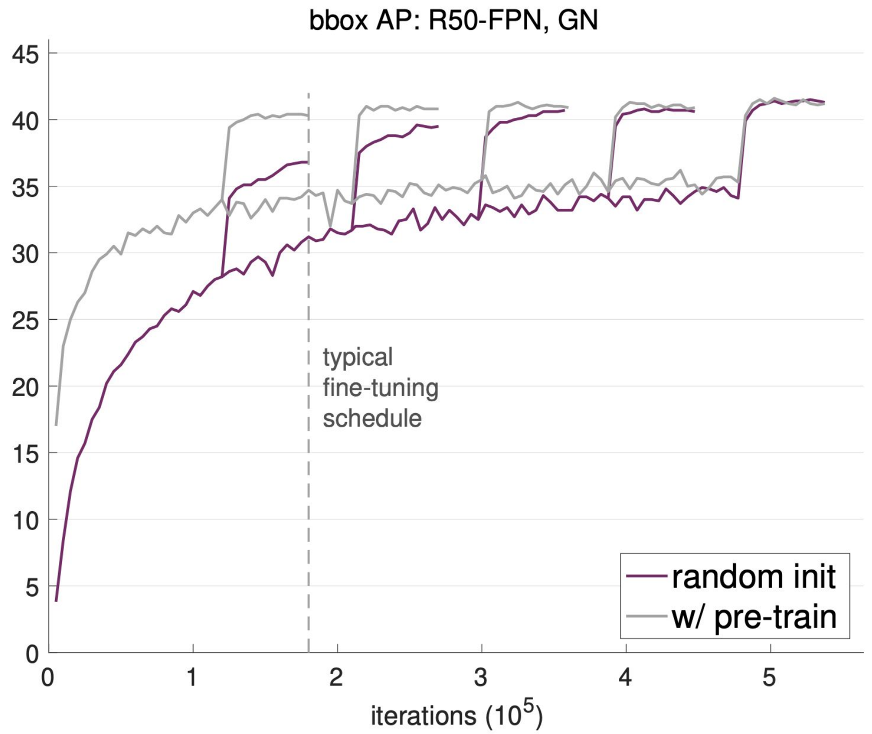
- **HW2+P2** is due today!
- **HW3+P3** will be released today. due 3/13/25 **(only 1 week!)**
- **Quiz 3** Paper released. Paper: *CLIP*
- **Final Project** signups released today. Look out for Ed announcement.

Cornell Bowers C·IS

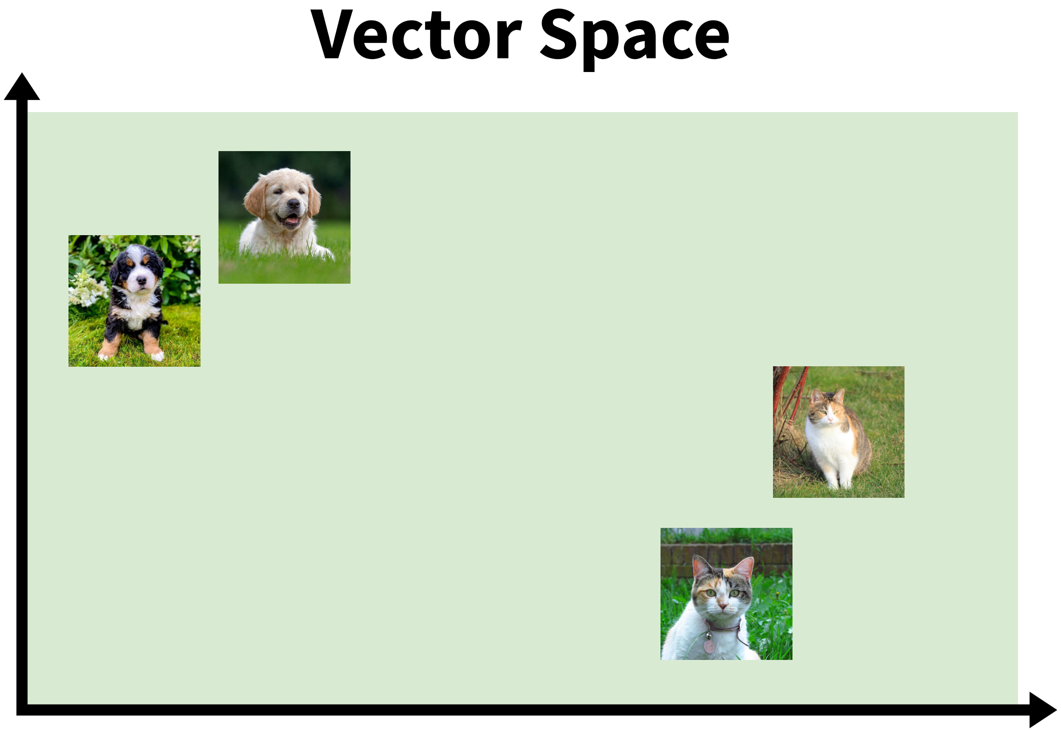
Image Classification



Cornell Bowers C·IS



Use self-supervised learning to learn embeddings for images



Cornell Bowers C·IS

Discussion: Comparison of Loss Functions

Triplet:

$$l = \max(0, \mathrm{sim}(\mathbf{x}, \mathbf{x}^-) - \mathrm{sim}(\mathbf{x}, \mathbf{x}^+) + c)$$

SimCLR loss:

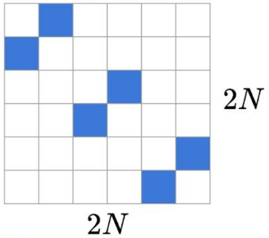
$$l = -\log\left(\frac{\exp(\mathrm{sim}(\mathbf{x}, \mathbf{x}^+)/\tau)}{\exp(\mathrm{sim}(\mathbf{x}, \mathbf{x}^+)/\tau) + \exp(\mathrm{sim}(\mathbf{x}, \mathbf{x}^-)/\tau)}\right)$$

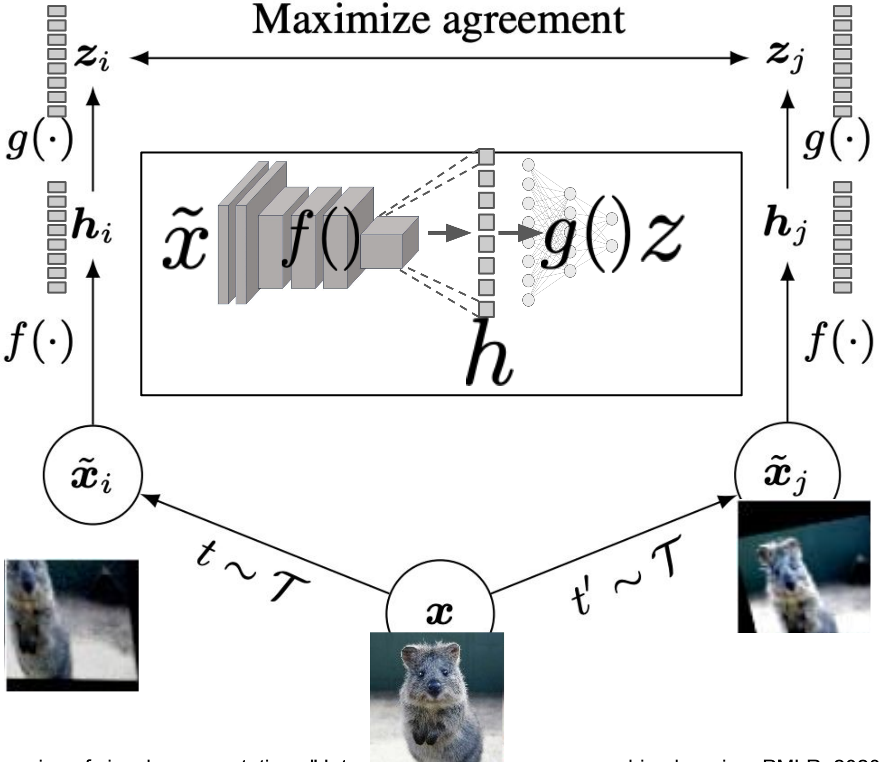
Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.

SimCLR

Algorithm 1 SimCLR's main learning algorithm.

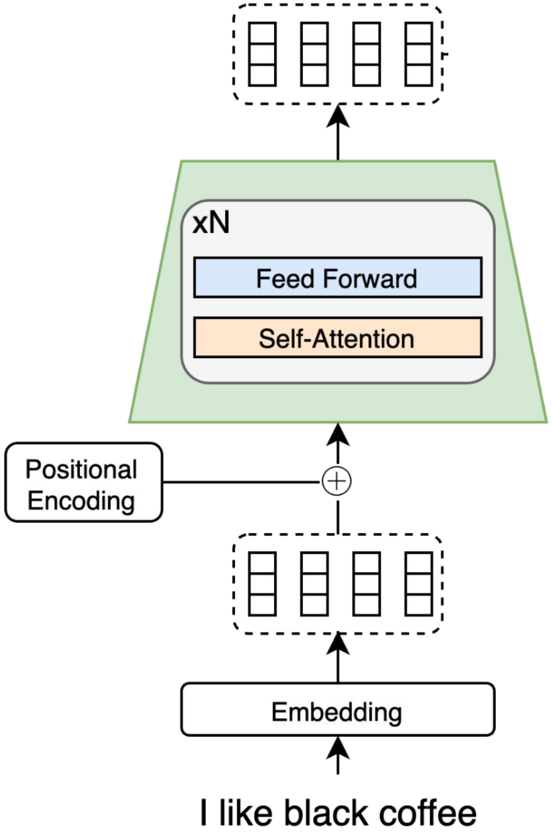
input: batch size N, constant τ, structure of f, g, $\mathcal{T}$.
for sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^{N}$ **do**
  for all $k \in \{1, \ldots, N\}$ **do**
    draw two augmentation functions $t \sim \mathcal{T}$, $t' \sim \mathcal{T}$
    # the first augmentation
    $\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$ # representation
    $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$ # projection
    # the second augmentation
    $\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$ # representation
    $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$ # projection
  end for
  for all $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ **do**
    $s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$ # pairwise similarity
  end for
  define $\ell(i,j)$ **as** $\ell(i,j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$
  $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
  update networks f and g to minimize $\mathcal{L}$
end for
return encoder network $f(\cdot)$, and throw away $g(\cdot)$





Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.

Remember Transformers?



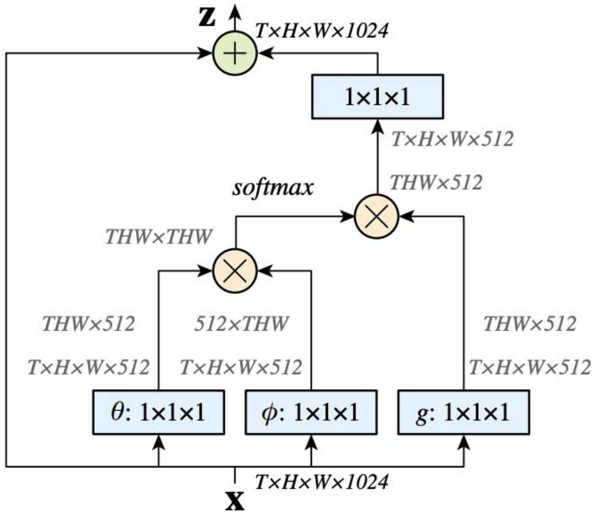
Cornell Bowers C·IS

How can we use Transformers on Imagers?

Cornell Bowers C·IS

How to use Attention for Vision Tasks?

Attempt #1: Add attention to existing CNNs

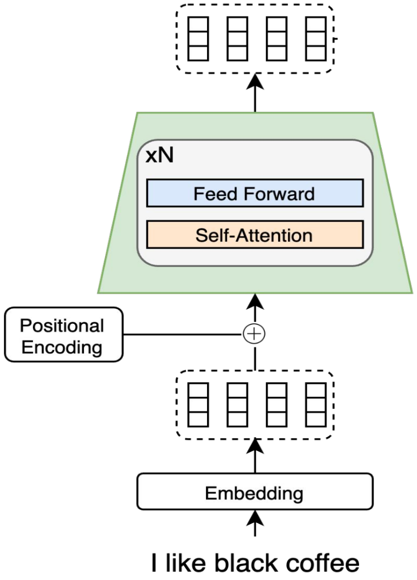


Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7794-7803).

Cornell Bowers C·IS

How to use Attention for Vision Tasks?

Attempt #2: Adapt standard transformers to image data



Cornell Bowers CIS

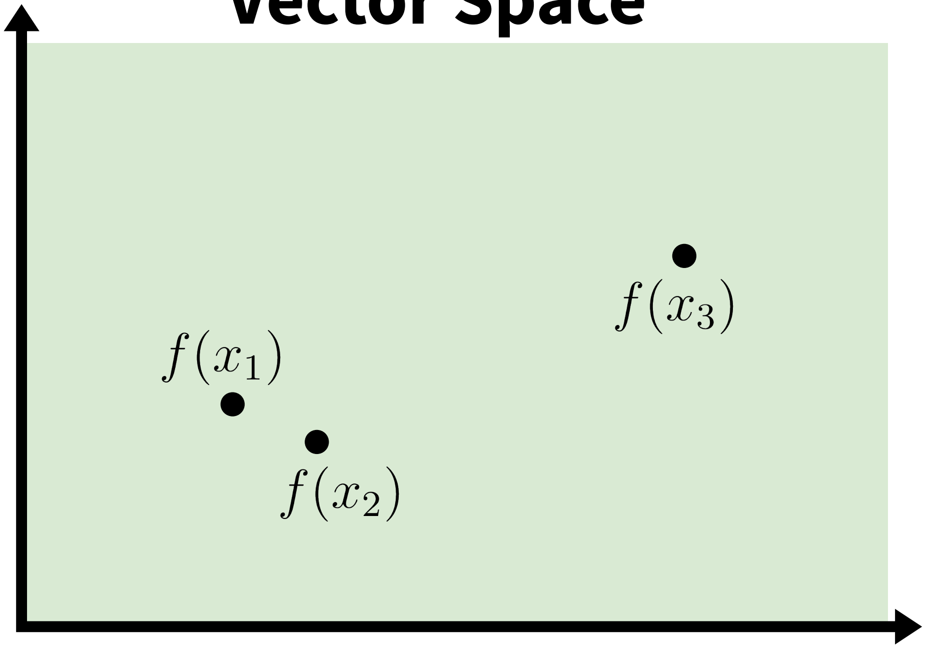
$f(x)$ = word embedding

"machine" x_1

"learning" x_2

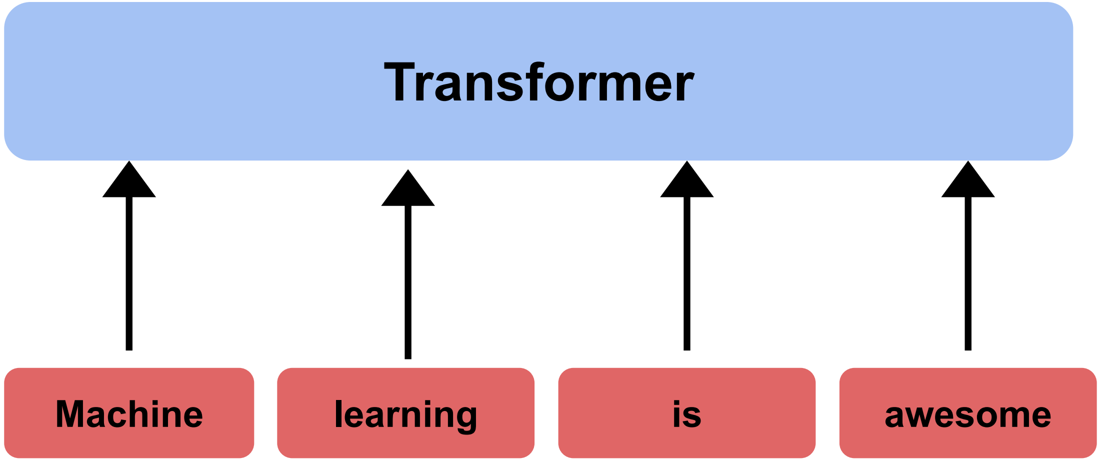
"awesome" x_3

Vector Space



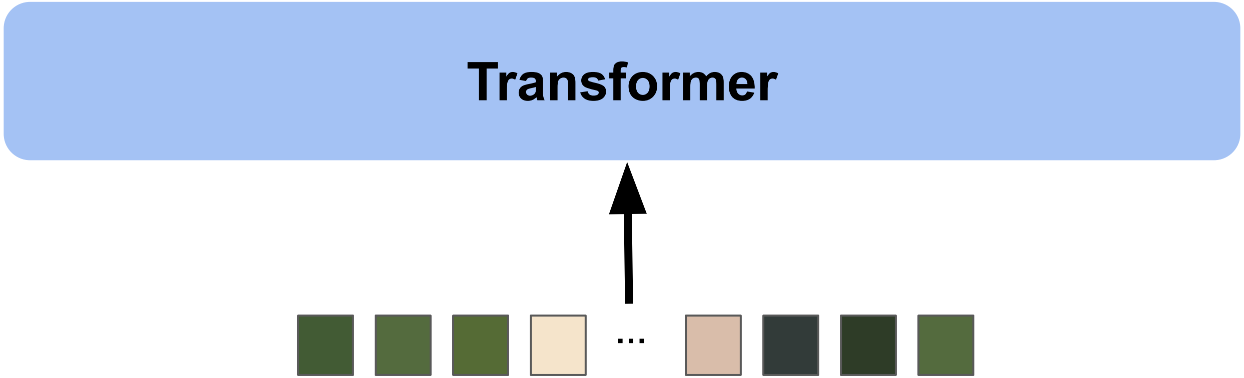
Cornell Bowers C·IS

Can we extend this idea to images?



Machine learning is awesome

Idea1: Use pixels as input tokens

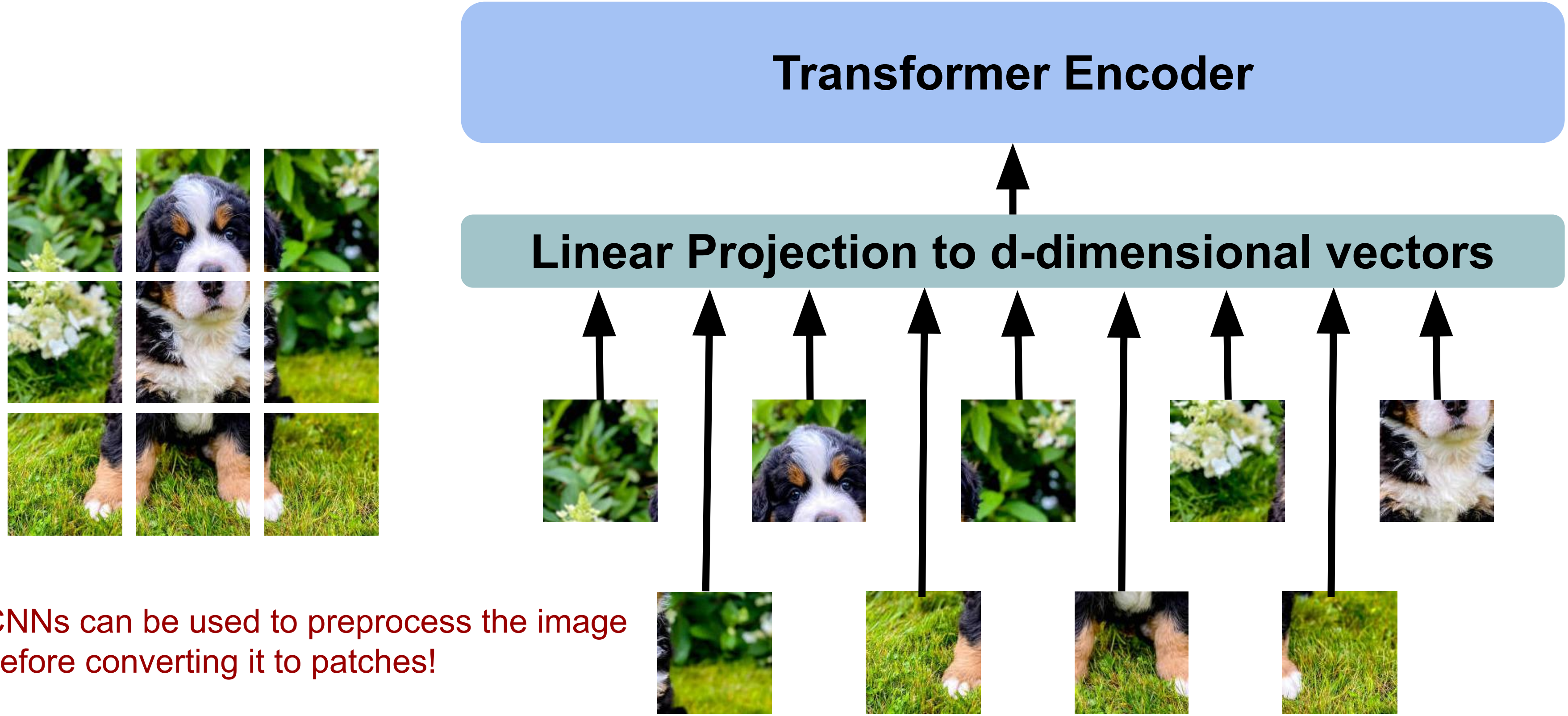


Discussion:

- How big would the attention matrix be for an RxR image?
- How much memory do we need for a 512x512 image, with 16 attention heads, and 48 layers - for attention alone?
- What if the image was 32x32 pixels?

https://web.eecs.umich.edu/~justincj/slides/eecs498/WI2022/598_WI2022_lecture18.pdf

Idea2: use image patches as input tokens



CNNs can be used to preprocess the image before converting it to patches!

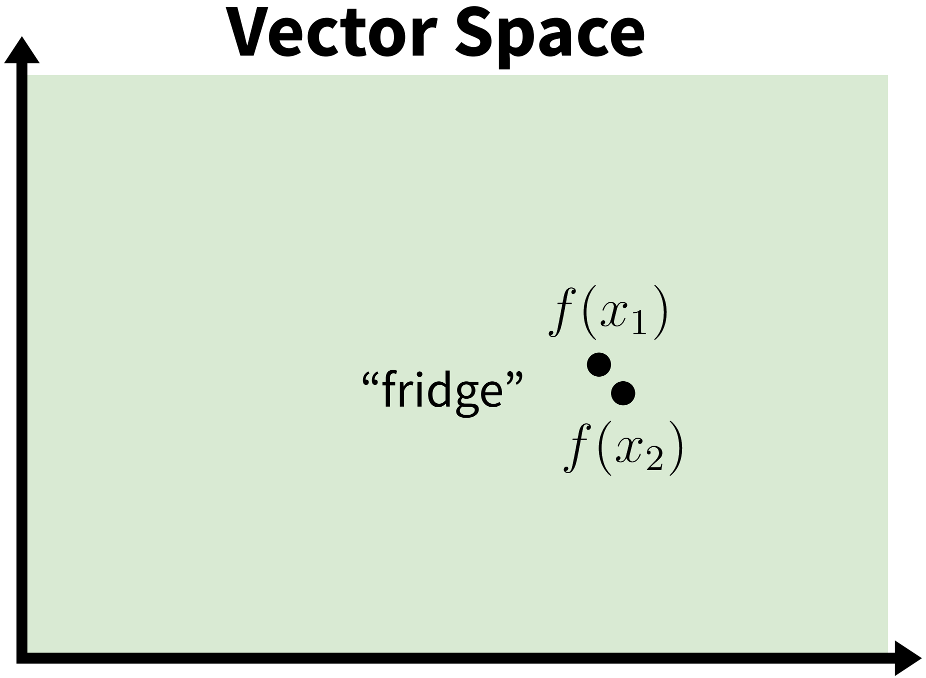
Cornell Bowers C·IS

$f(x)$ = convolution

x_1

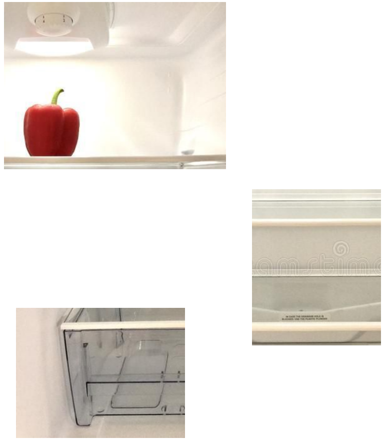
x_2

At a high level, these are both just fridges



Cornell Bowers C·IS

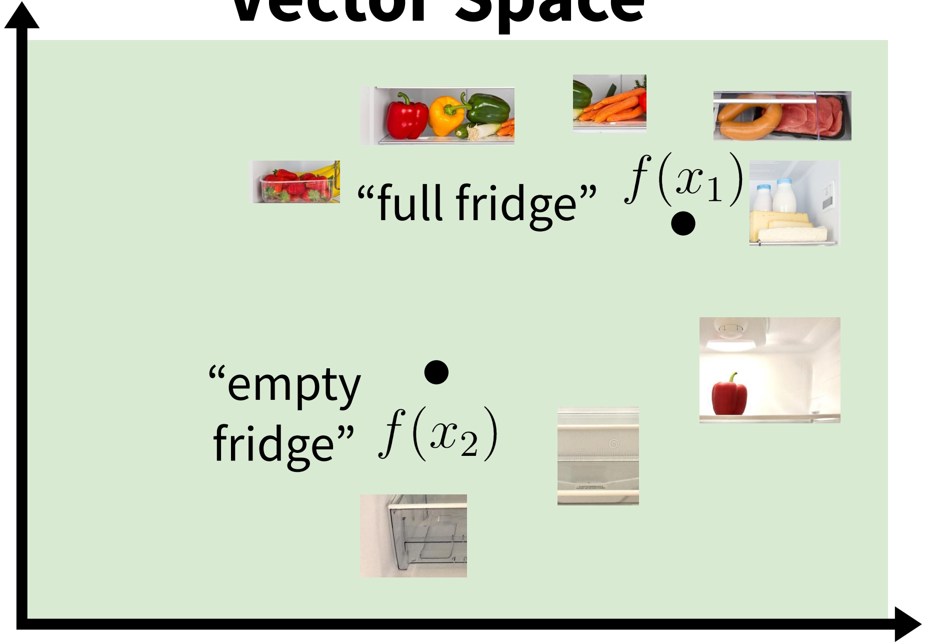
x_1



x_2

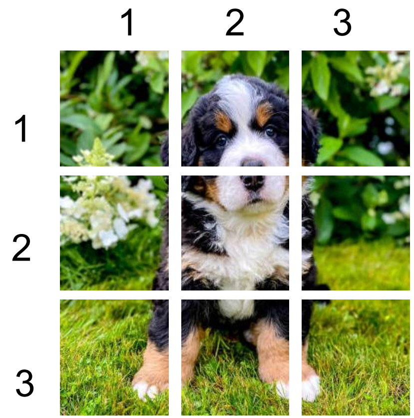
$f(x)$ = Vision Transformers

Vector Space

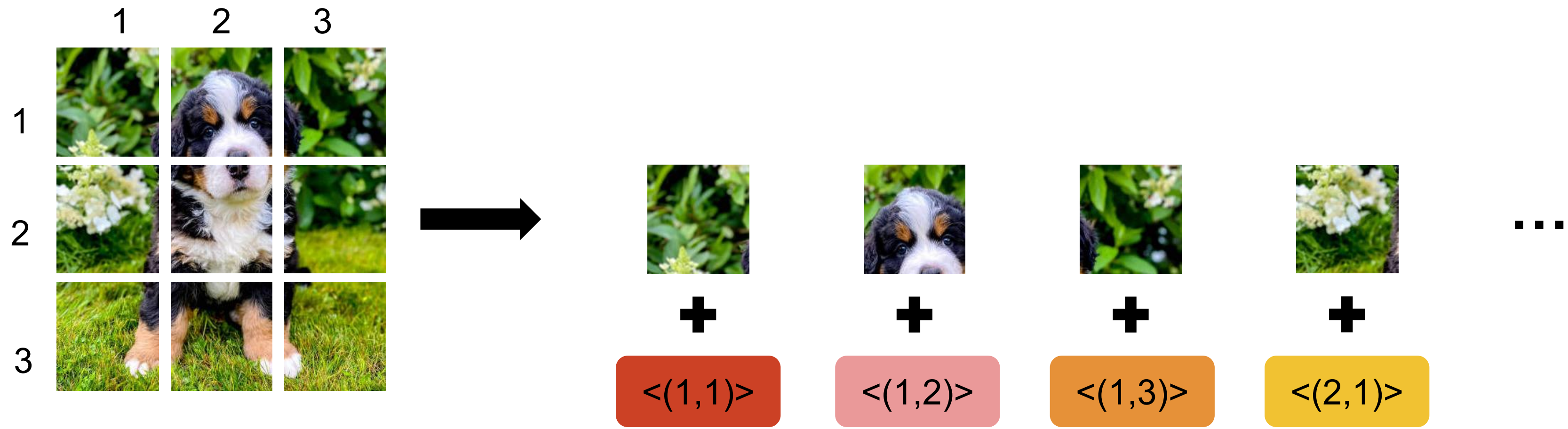


Use image patches like words in a sentence!

Adding positional embeddings

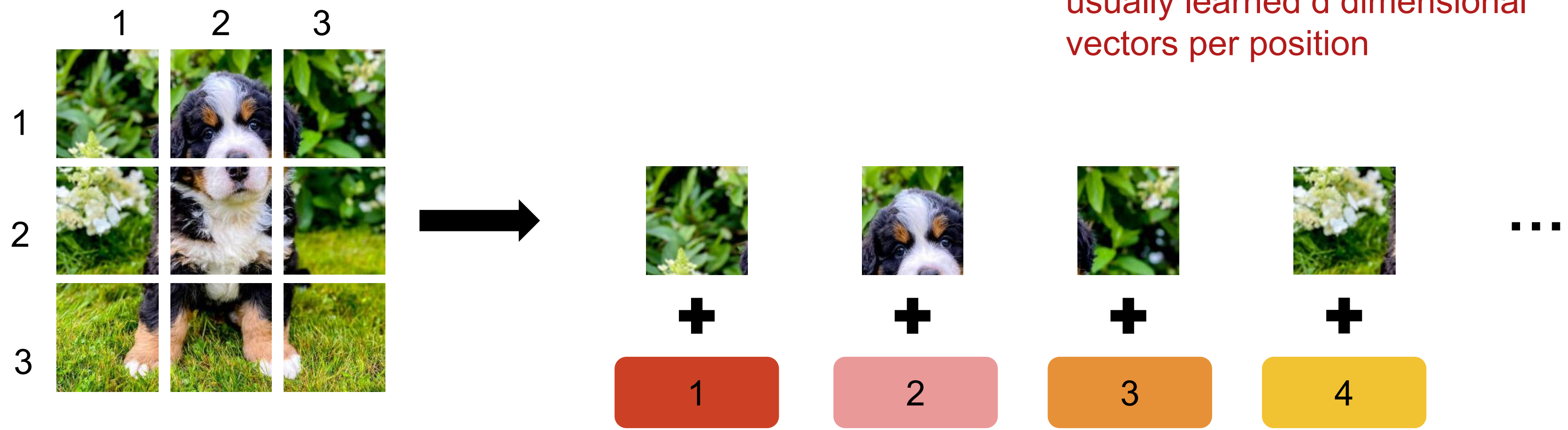


Adding positional embeddings

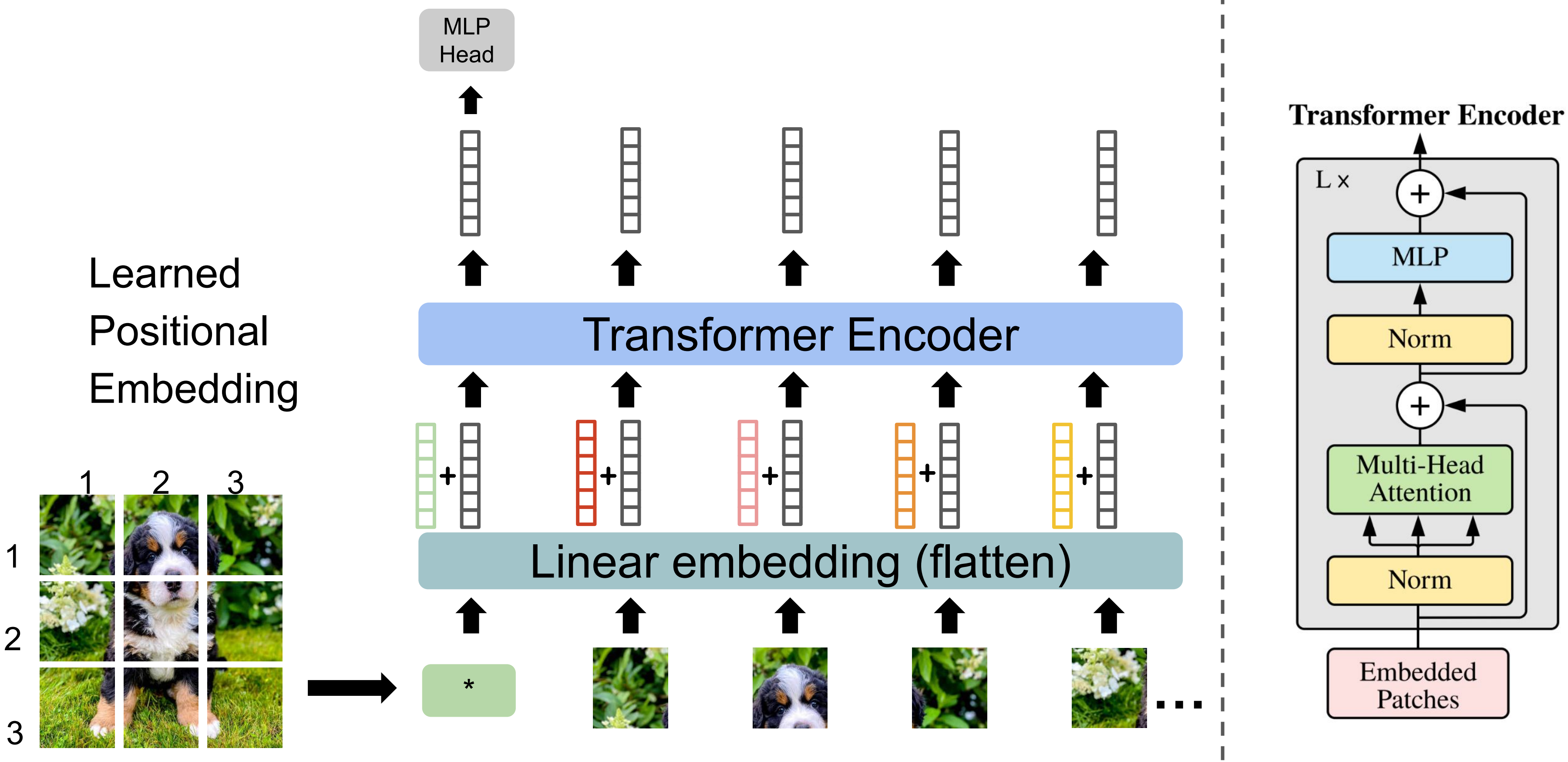


Adding positional embeddings

Positional embeddings for ViTs are usually learned d dimensional vectors per position

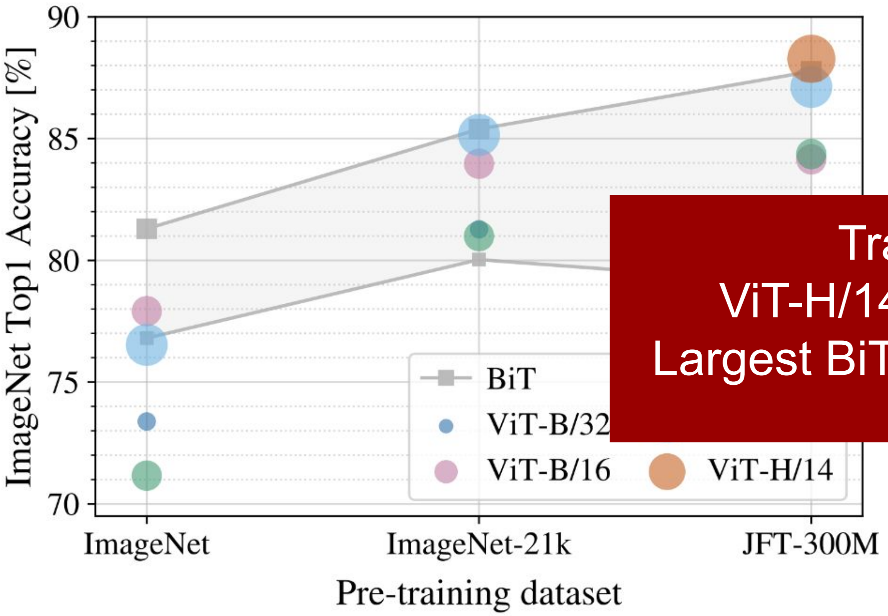


Vision Transformer

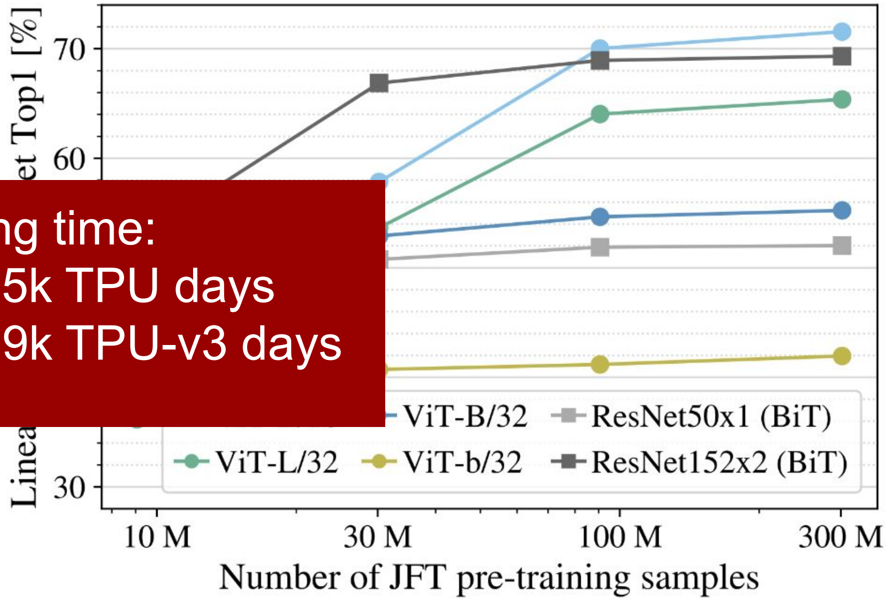


Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

ViT Results



Training time:
ViT-H/14: 2.5k TPU days
Largest BiT: 9.9k TPU-v3 days



Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

ViT Summary

Model:

- Model is almost identical to Transormers for text sequences
- Replace words with PxP pixel image patches, P ∈ {14, 16, 32} (no overlap)
- Each patch is embedded linearly into a vector of size 1024
- 1D **learned** positional embeddings

Training:

- For pre-training, optimize for image classification on large supervised dataset (e.g. ImageNet 21K, JFT -300M)
- For fine-tuning, learn a new classification head on a small dataset (e.g. CIFAR-100)

ACTIVITY: When do ViTs outperform CNNs, and vice versa?

Think about what you know about transformers - what are some of their drawbacks?

When is it “worth it” to use transformers instead of just CNNs?

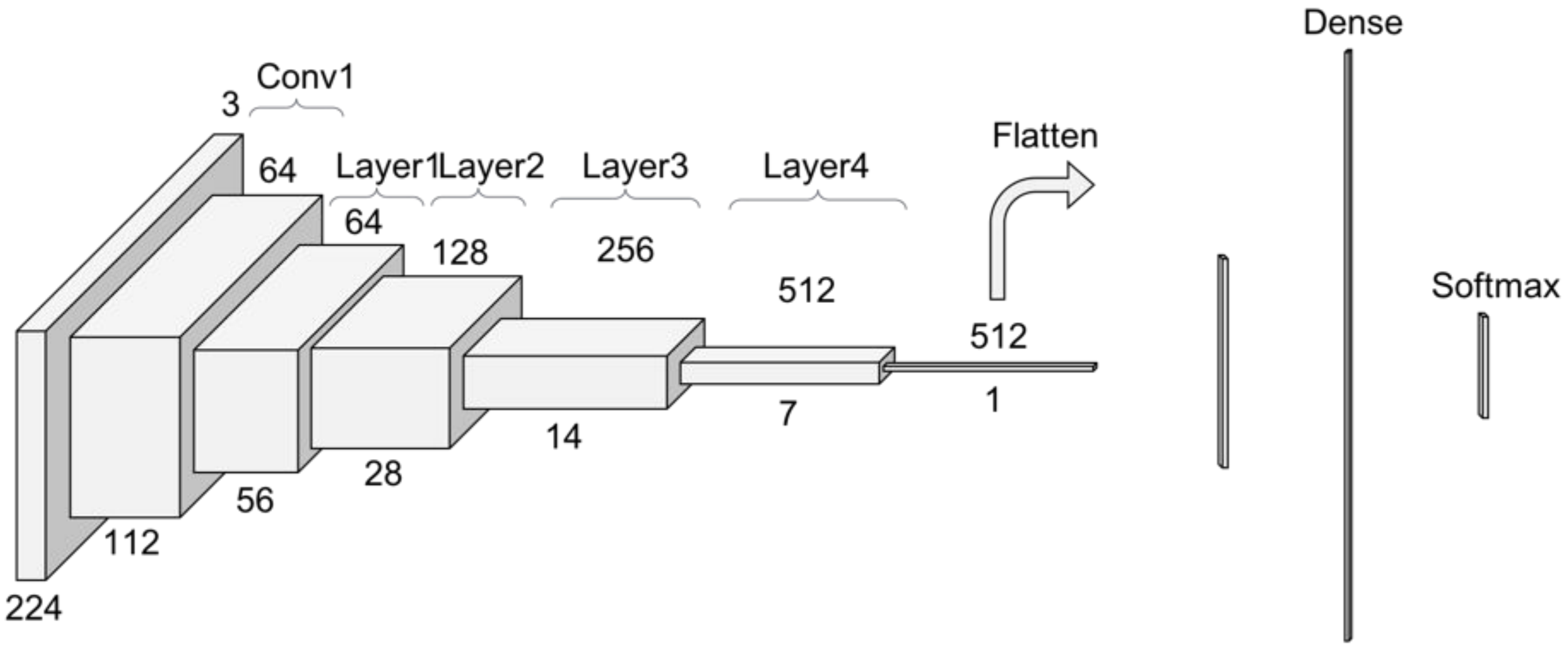
CNNs

- Translational invariance
- Simple, proven architecture
- Capture features at different scales
- Required less data

ViTs

- Attention mechanisms
- Allow for multimodal data integration

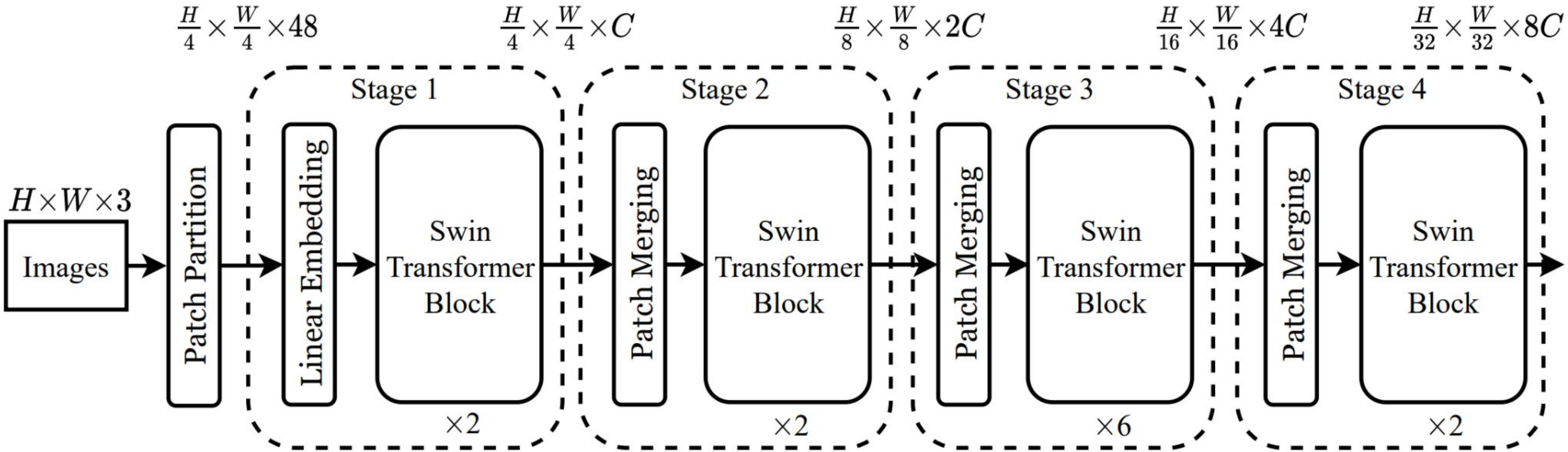
Cornell Bowers C·IS



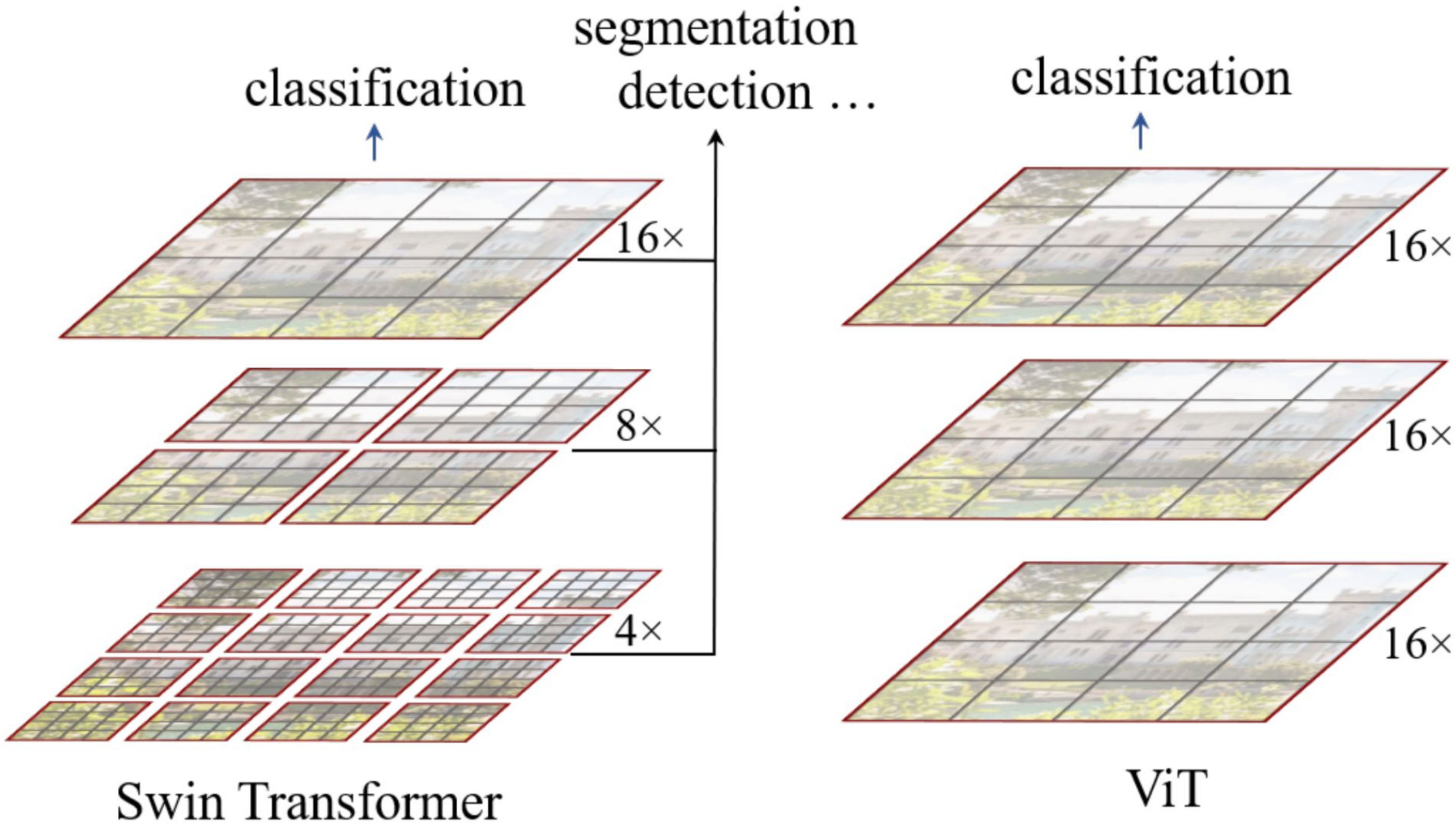
Cornell Bowers C·IS

Idea 3: Swim Transformers

Hierarchical architecture that has the flexibility to model at various scales

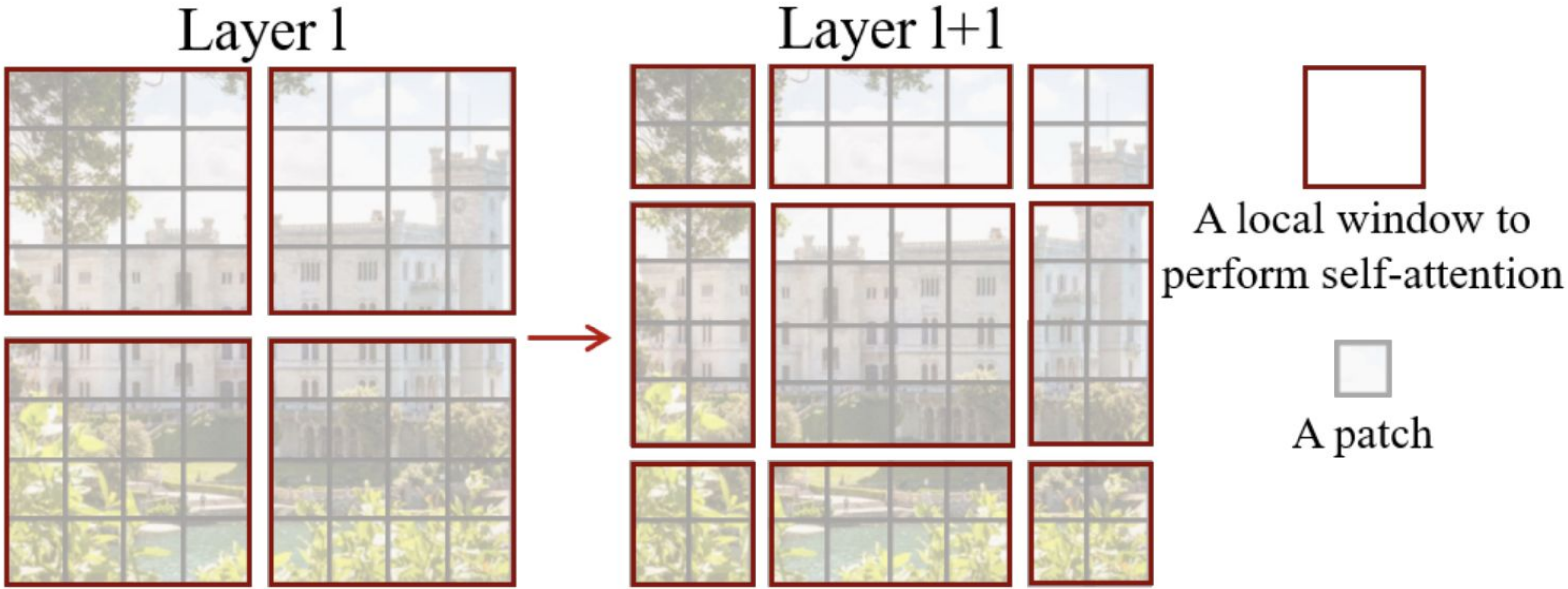


Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10012-10022).


segmentation
detection …
classification
16×
8×
4×
Swin Transformer
classification
16×
16×
16×
ViT

Shifted Window attention

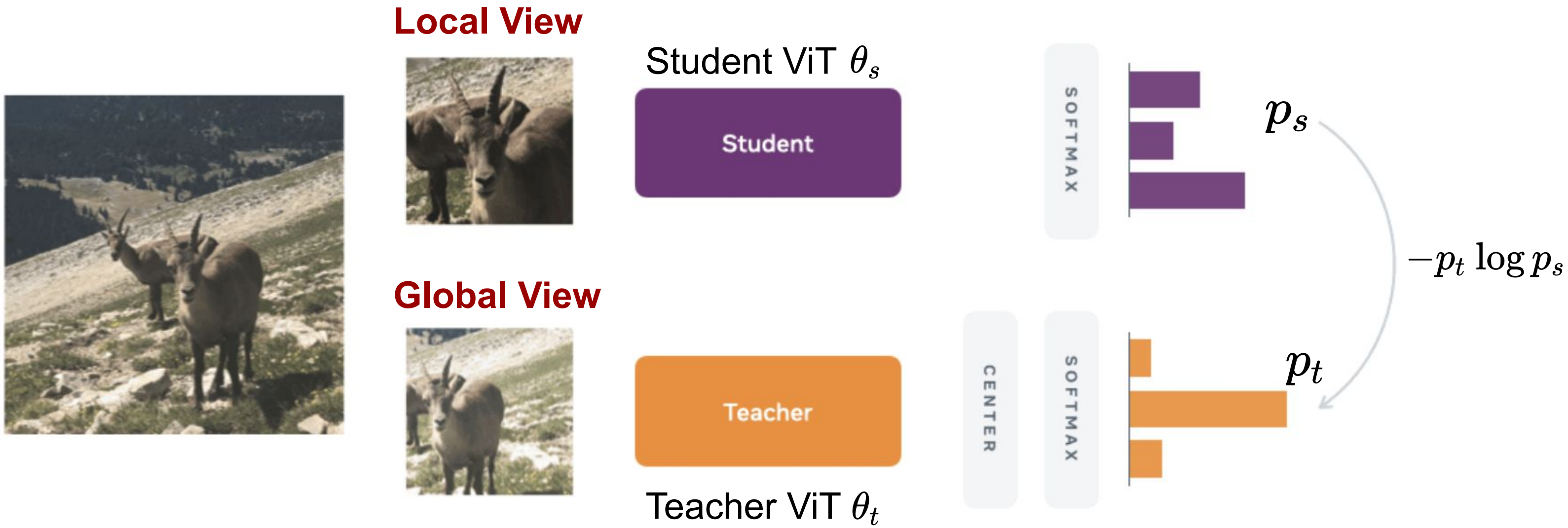
Linear computational complexity with respect to image size



Performance

method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
R-101x3 [38]	384^2	388M	204.6G	-	84.4
R-152x4 [38]	480^2	937M	840.5G	-	85.4
ViT-B/16 [20]	384^2	86M	55.4G	85.9	84.0
ViT-L/16 [20]	384^2	307M	190.7G	27.3	85.2
Swin-B	224^2	88M	15.4G	278.1	85.2
Swin-B	384^2	88M	47.0G	84.7	86.4
Swin-L	384^2	197M	103.9G	42.1	87.3

Self-Supervised Vision Transformers (DiNO)



Momentum encoder: $\theta_t \leftarrow \lambda\theta_t + (1-\lambda)\theta_s$

Cornell Bowers C·IS

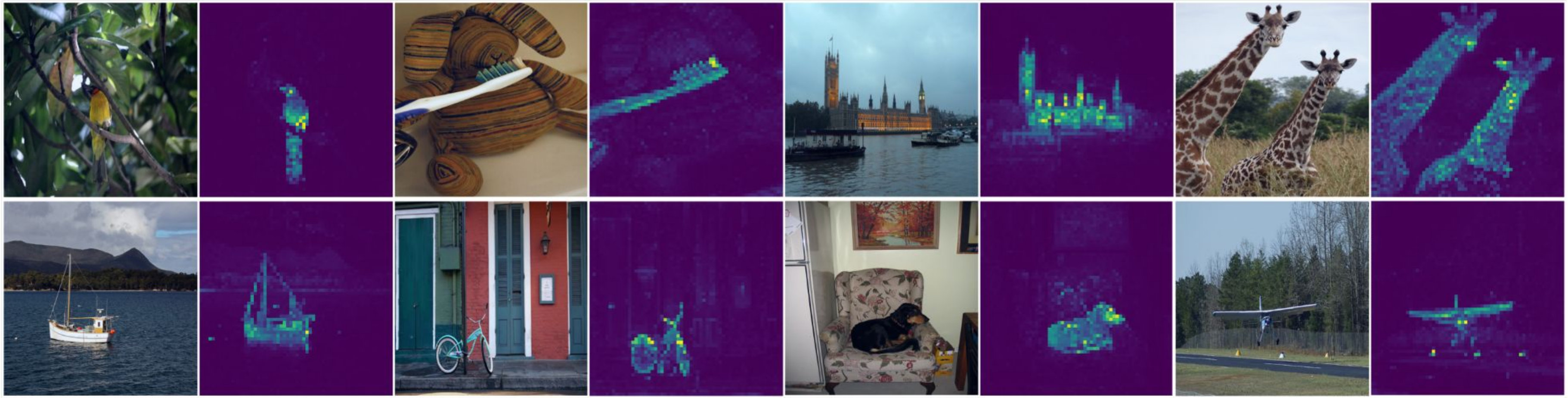


Figure 1: **Self-attention from a Vision Transformer with 8×8 patches trained with no supervision.** We look at the self-attention of the `[CLS]` token on the heads of the last layer. This token is not attached to any label nor supervision. These maps show that the model automatically learns class-specific features leading to unsupervised object segmentations.

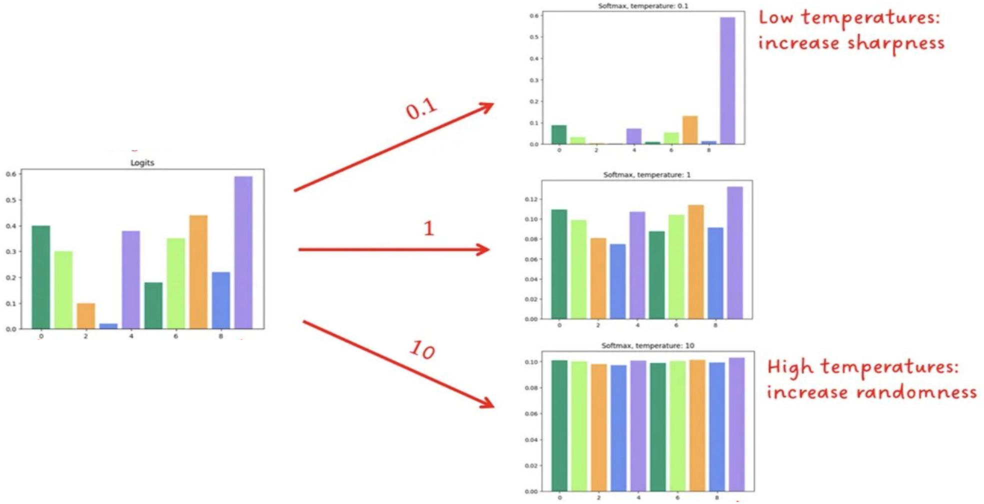
Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9650-9660).

Cornell Bowers C·IS

Centering and sharpening

- Centering prevents one dimension from dominating
- Sharpening prevents learning a uniform distribution

$$\mathrm{softmax}(x)_i = \frac{e^{\frac{y_i}{T}}}{\Sigma_j^N e^{\frac{y_j}{T}}}$$

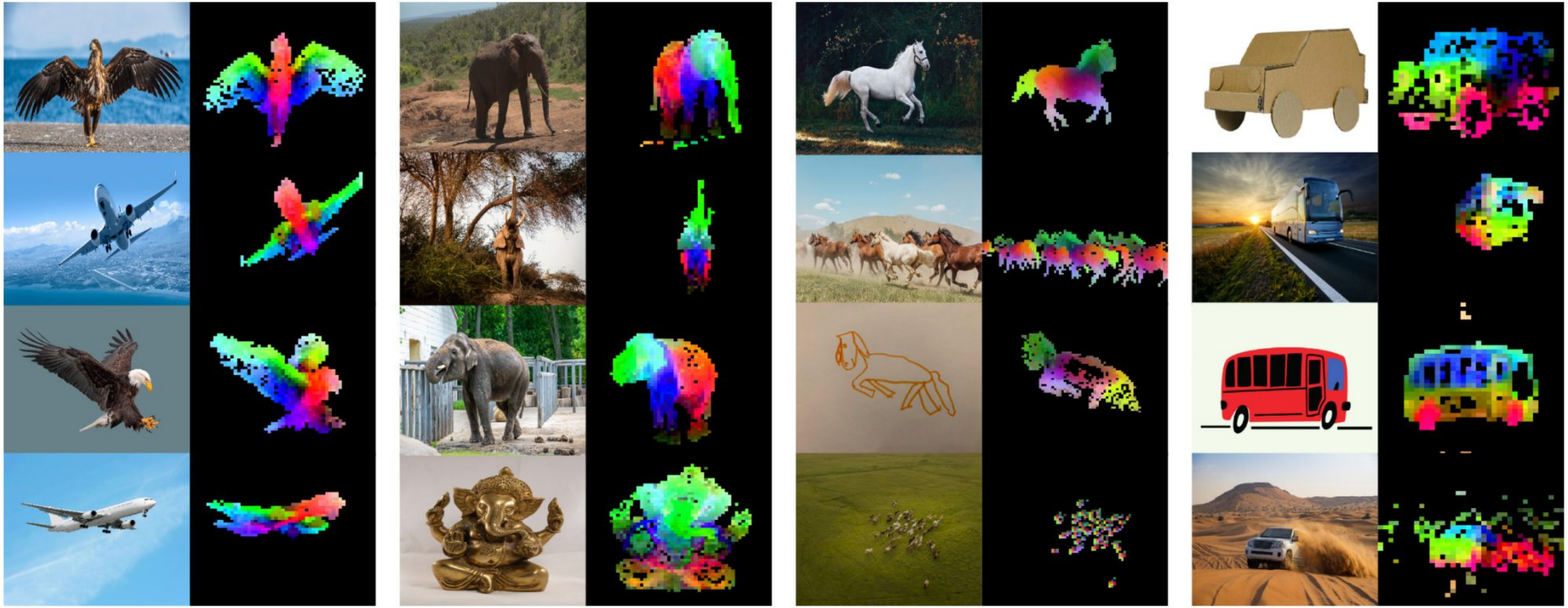


DINO v2

	INet-1k k-NN	INet-1k linear
iBOT	72.9	82.3
+(our reproduction)	74.5 $\uparrow$ 1.6	83.2 $\uparrow$ 0.9
+LayerScale, Stochastic Depth	75.4 $\uparrow$ 0.9	82.0 $\downarrow$ 1.2
+128k prototypes	76.6 $\uparrow$ 1.2	81.9 $\downarrow$ 0.1
+KoLeo	78.9 $\uparrow$ 2.3	82.5 $\uparrow$ 0.6
+SwiGLU FFN	78.7 $\downarrow$ 0.2	83.1 $\uparrow$ 0.6
+Patch size 14	78.9 $\uparrow$ 0.2	83.5 $\uparrow$ 0.4
+Teacher momentum 0.994	79.4 $\uparrow$ 0.5	83.6 $\uparrow$ 0.1
+Tweak warmup schedules	80.5 $\uparrow$ 1.1	83.8 $\uparrow$ 0.2
+Batch size 3k	81.7 $\uparrow$ 1.2	84.7 $\uparrow$ 0.9
+Sinkhorn-Knopp	81.7 =	84.7 =
+Untying heads = DINOv2	82.0 $\uparrow$ 0.3	84.5 $\downarrow$ 0.2

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., ... & Bojanowski, P. (2023). Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

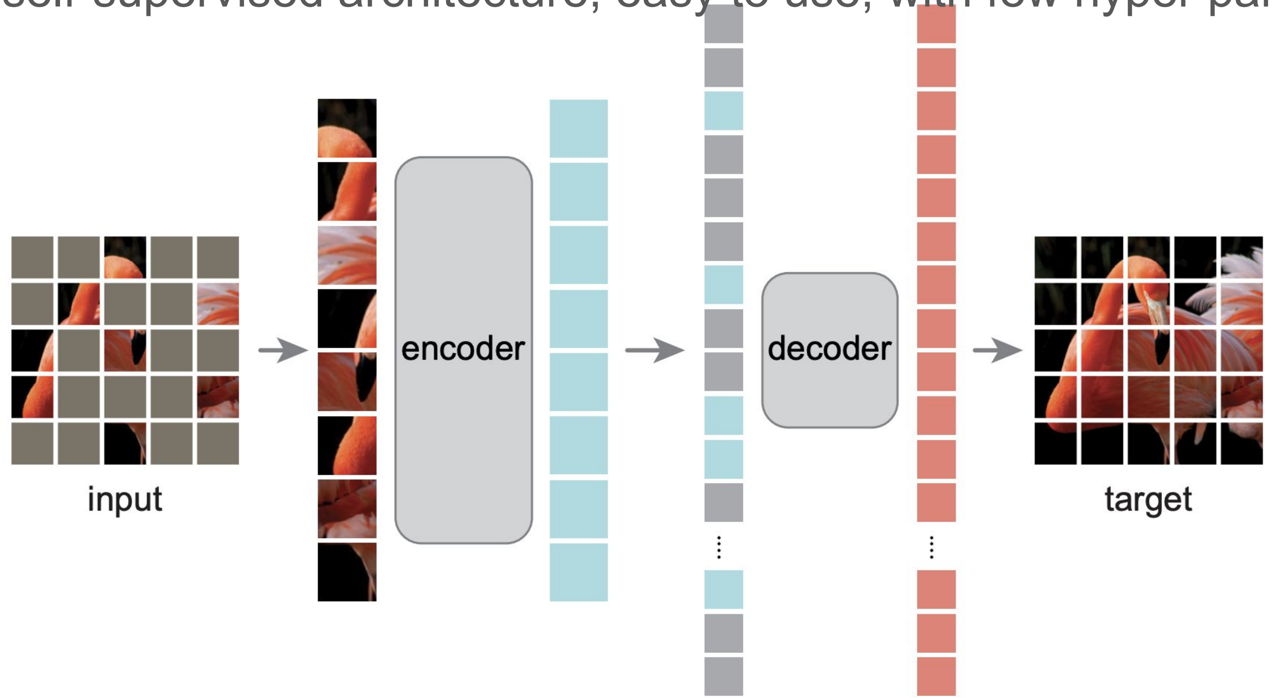
Cornell Bowers C·IS



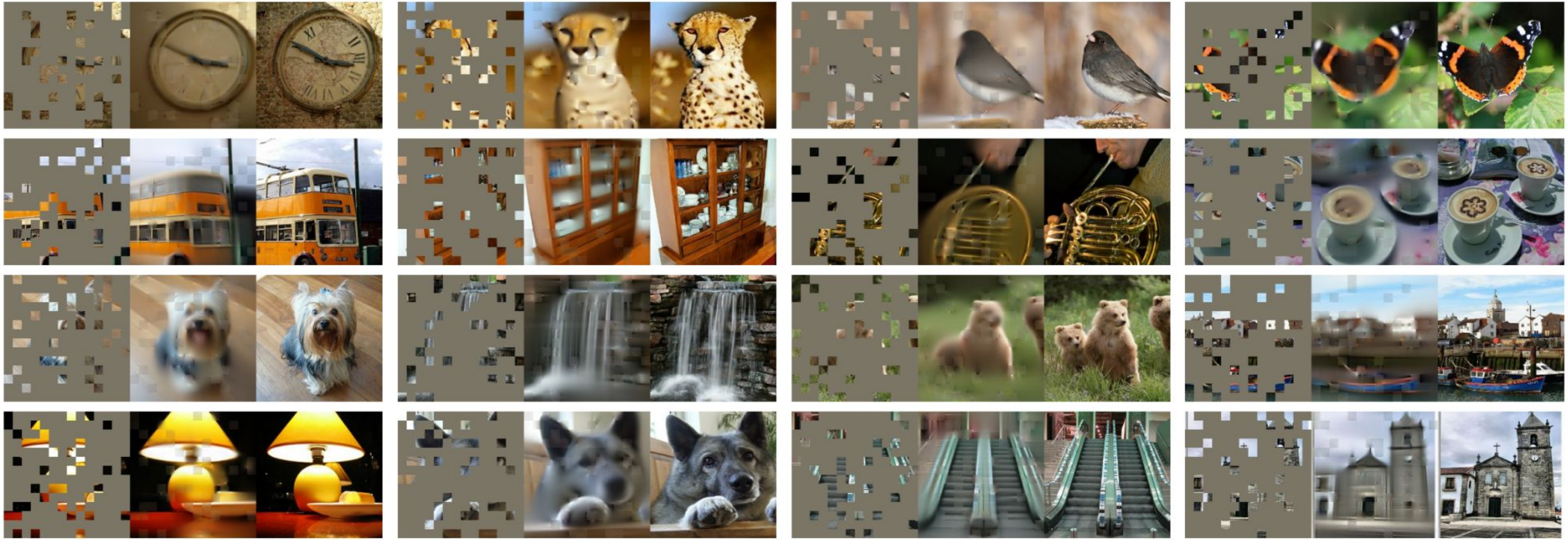
Cornell Bowers C·IS

Masked Autoencoders (MaE)

A simple self-supervised architecture, easy to use, with few hyper parameters.



He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 16000-16009).



Discuss: BERT is trained with cross entropy loss. What loss function should we use for MaE?

Cornell Bowers C·IS

MaE Results

- Compared to supervised ViTs
  - Requires minimal data augmentation
  - Transfers better to downstream vision tasks
    - Object detection, segmentation

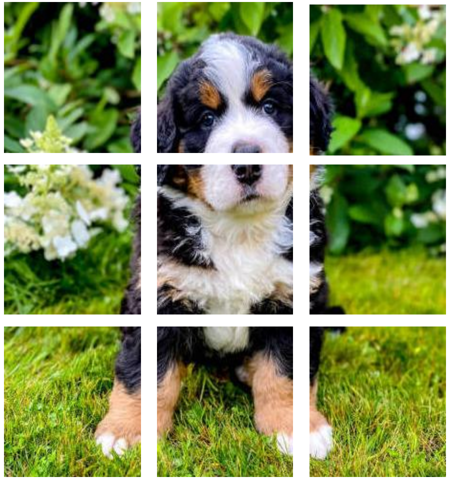
case	ft	lin
none	84.0	65.7
crop, fixed size	84.7	73.1
crop, rand size	**84.9**	**73.5**
crop + color jit	84.3	71.9

(e) **Data augmentation**. Our MAE works with minimal or no augmentation.

		AP^{box}		AP^{mask}	
method	pre-train data	ViT-B	ViT-L	ViT-B	ViT-L
supervised	IN1K w/ labels	47.9	49.3	42.9	43.9
MoCo v3	IN1K	47.9	49.3	42.7	44.0
BEiT	IN1K+DALLE	49.8	**53.3**	44.4	47.1
MAE	IN1K	**50.3**	**53.3**	**44.9**	**47.2**

Table 4. **COCO object detection and segmentation** using a ViT Mask R-CNN baseline. All entries are based on our implementation. Self-supervised entries use IN1K data *without* labels. Mask AP follows a similar trend as box AP.

$f(x)$ = transformer rep.

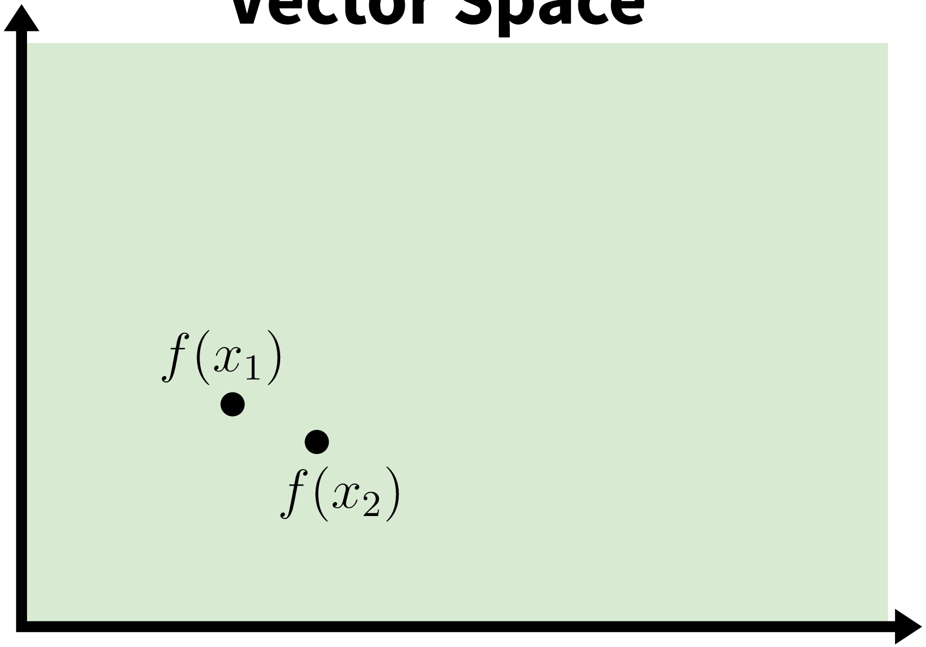


x_1

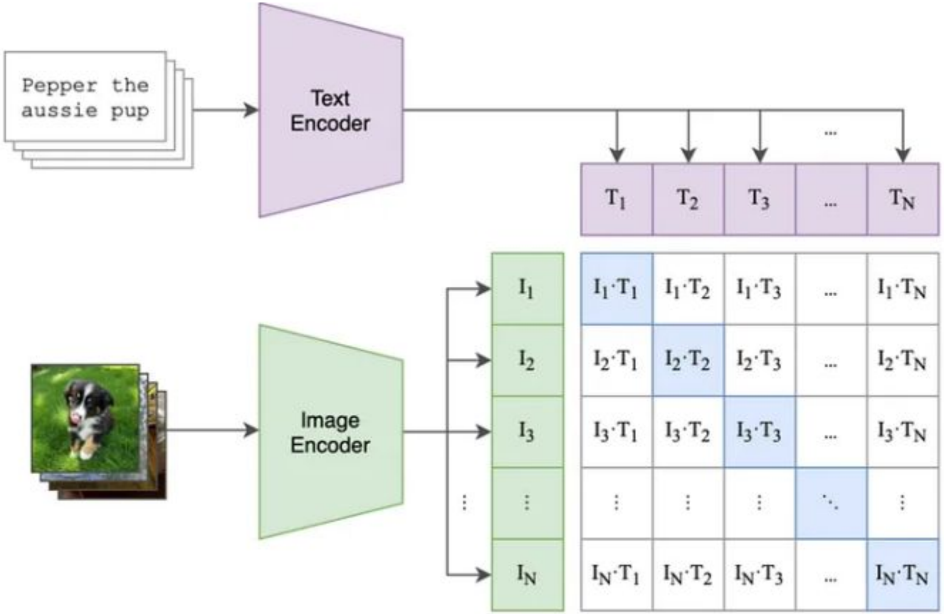
A puppy sits with flowers

x_2

Vector Space



CLIP (Contrastive Language–Image Pre-training)



Conde, M. V., & Turgutlu, K. (2021). CLIP-Art: Contrastive pre-training for fine-grained art classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3956-3960).

Discuss: How can you train this model?

```
# 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]
```

- Trained on 256 V100 GPUs for two weeks on 400 million (image, text pairs)
- On AWS, this would cost at least 200k dollars

```
# 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
```

Cornell Bowers C·IS

Ranking using CLIP

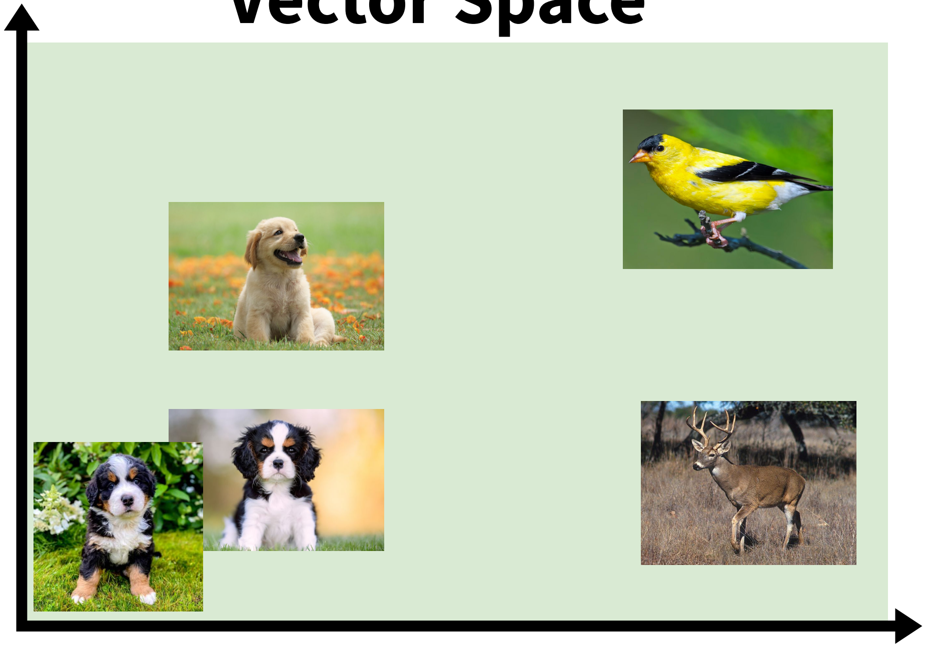
sim score - 0.9

sim score - 0.3

sim score - 0.6

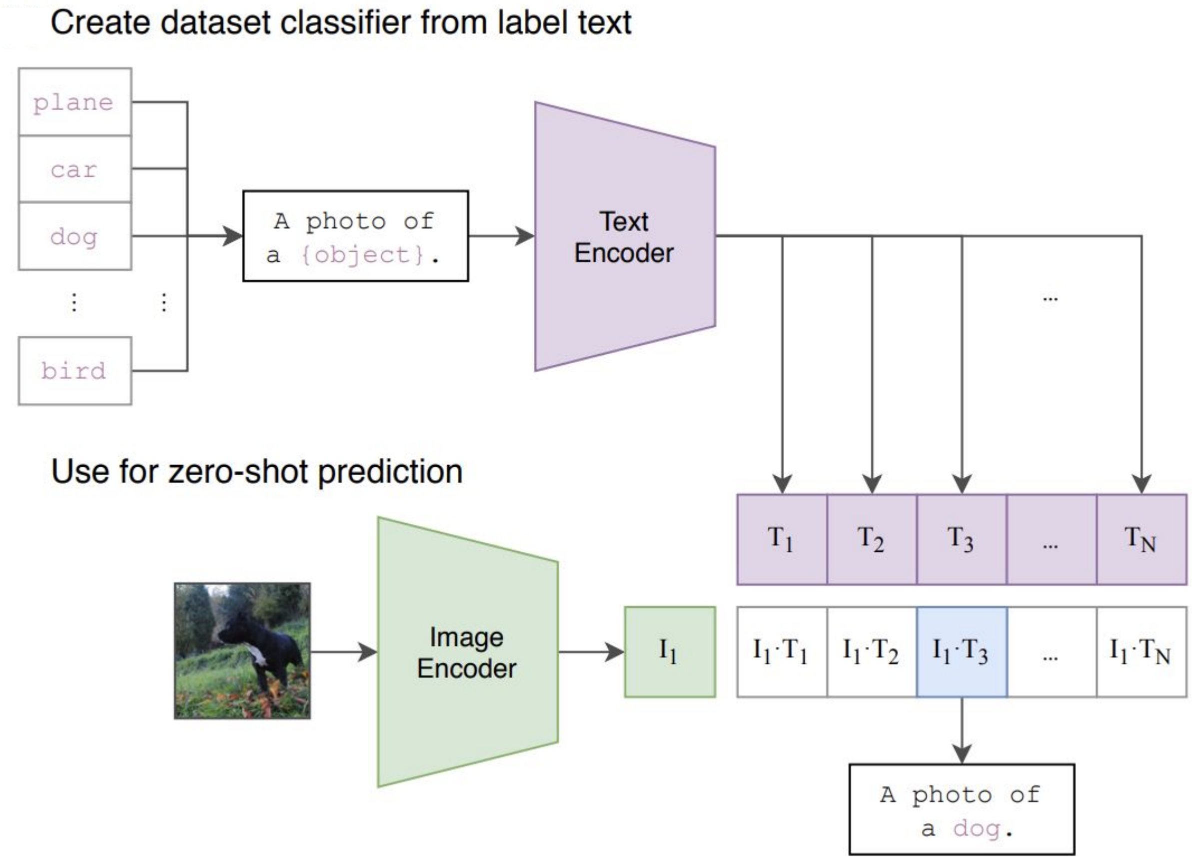
sim score - 0.1

Vector Space



Clip demo

https://huggingface.co/spaces/vivien/clip


Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
Use for zero-shot prediction
Image Encoder
I_1
T_1
T_2
T_3
...
T_N
$I_1 \cdot T_1$
$I_1 \cdot T_2$
$I_1 \cdot T_3$
...
$I_1 \cdot T_N$
A photo of a dog.

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

- ✓ 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

- ✓ 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**.

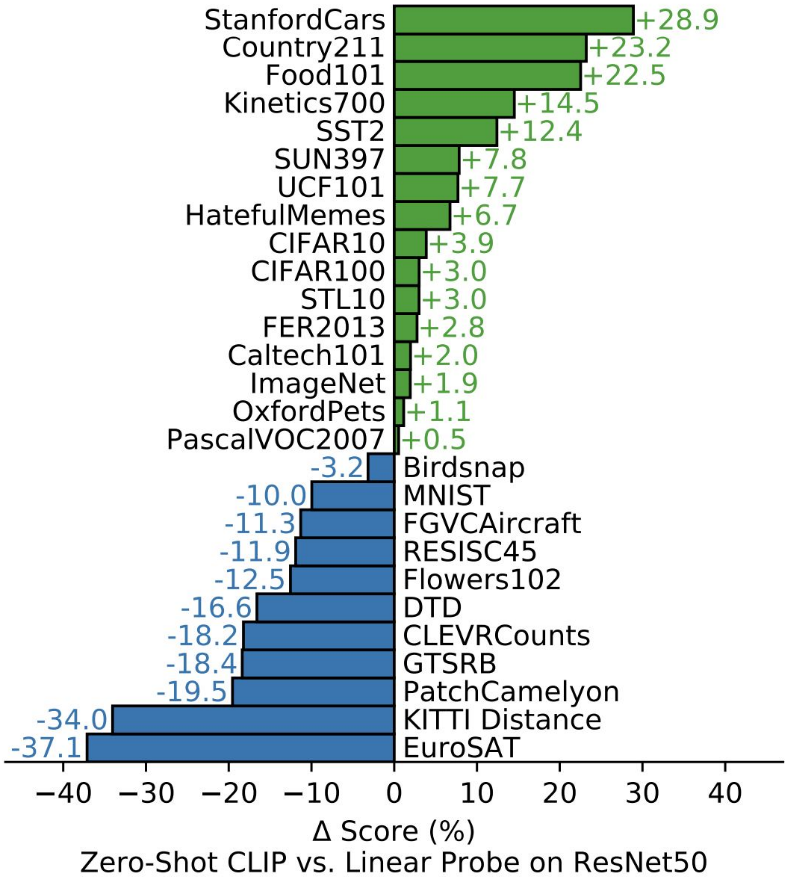
EUROSAT

annual crop land (12.9%) Ranked 4 out of 10

- × 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**.

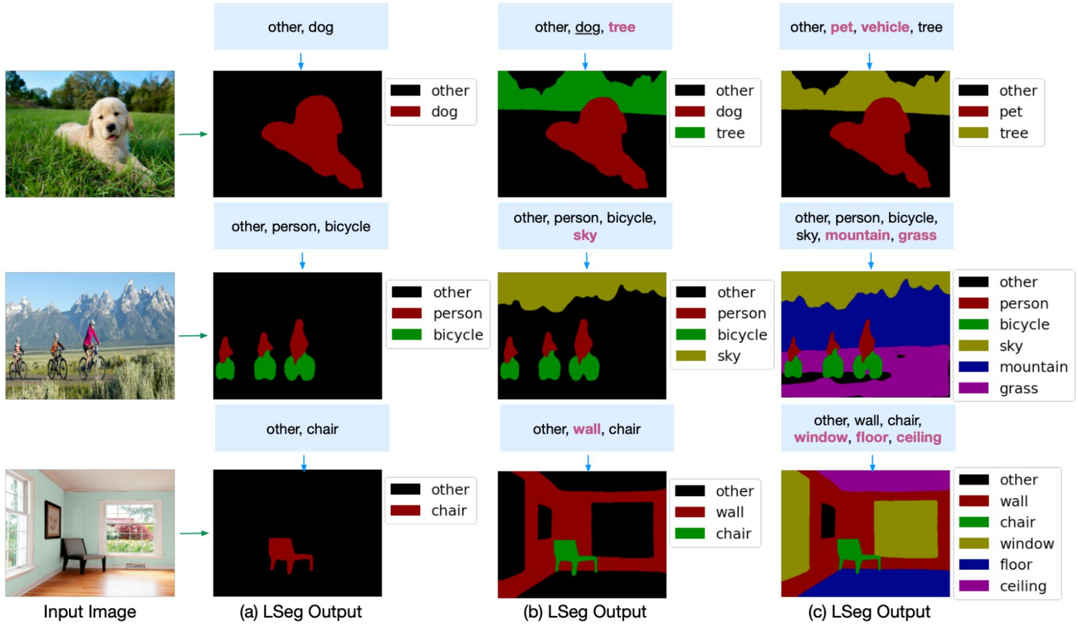
https://openai.com/research/clip

Cornell Bowers C·IS



Cornell Bowers C·IS

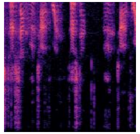
Application of CLIP



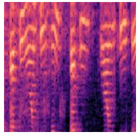
https://arxiv.org/pdf/2201.03546.pdf

Cornell Bowers C·IS

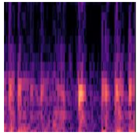
$f(x)$ = audio spectrogram transformer



x_1

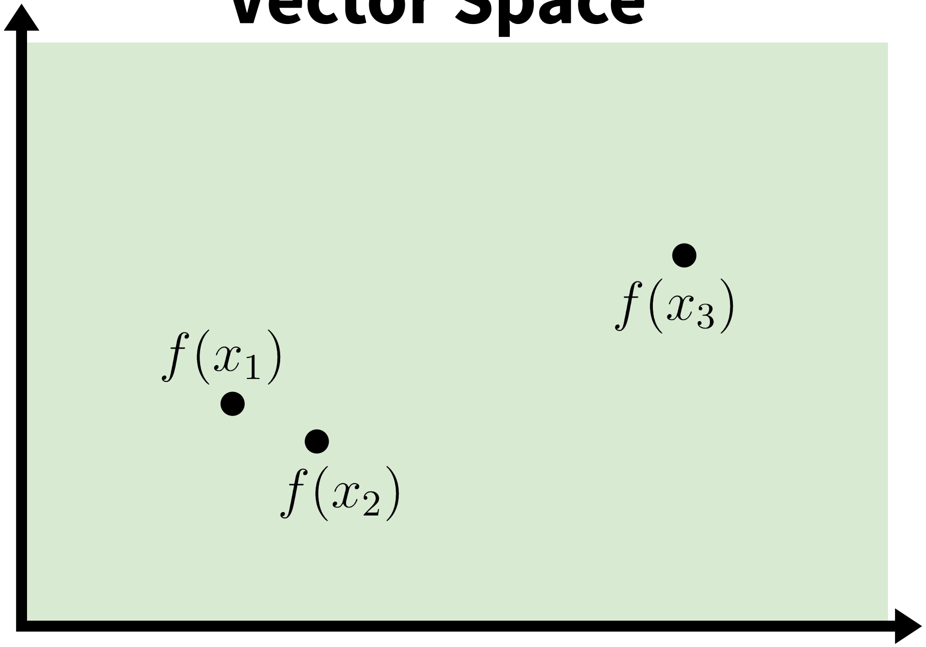


x_2

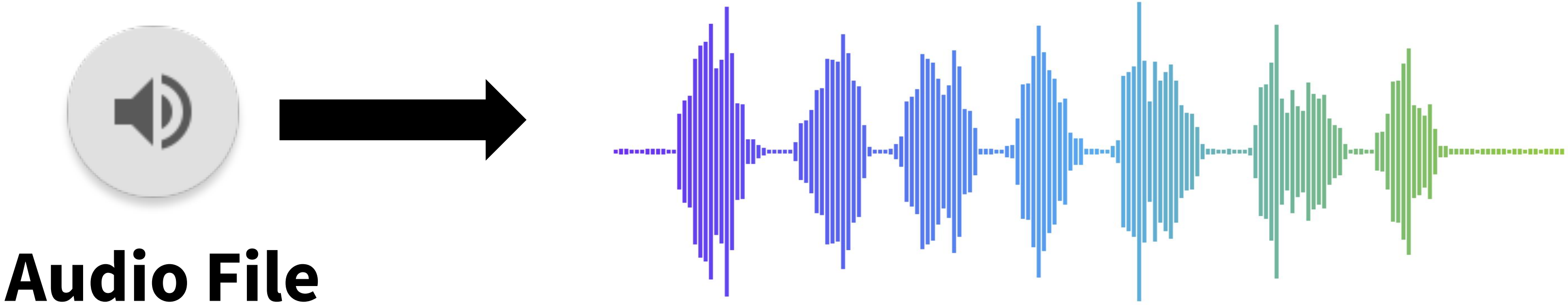


x_3

Vector Space



Audio Processing

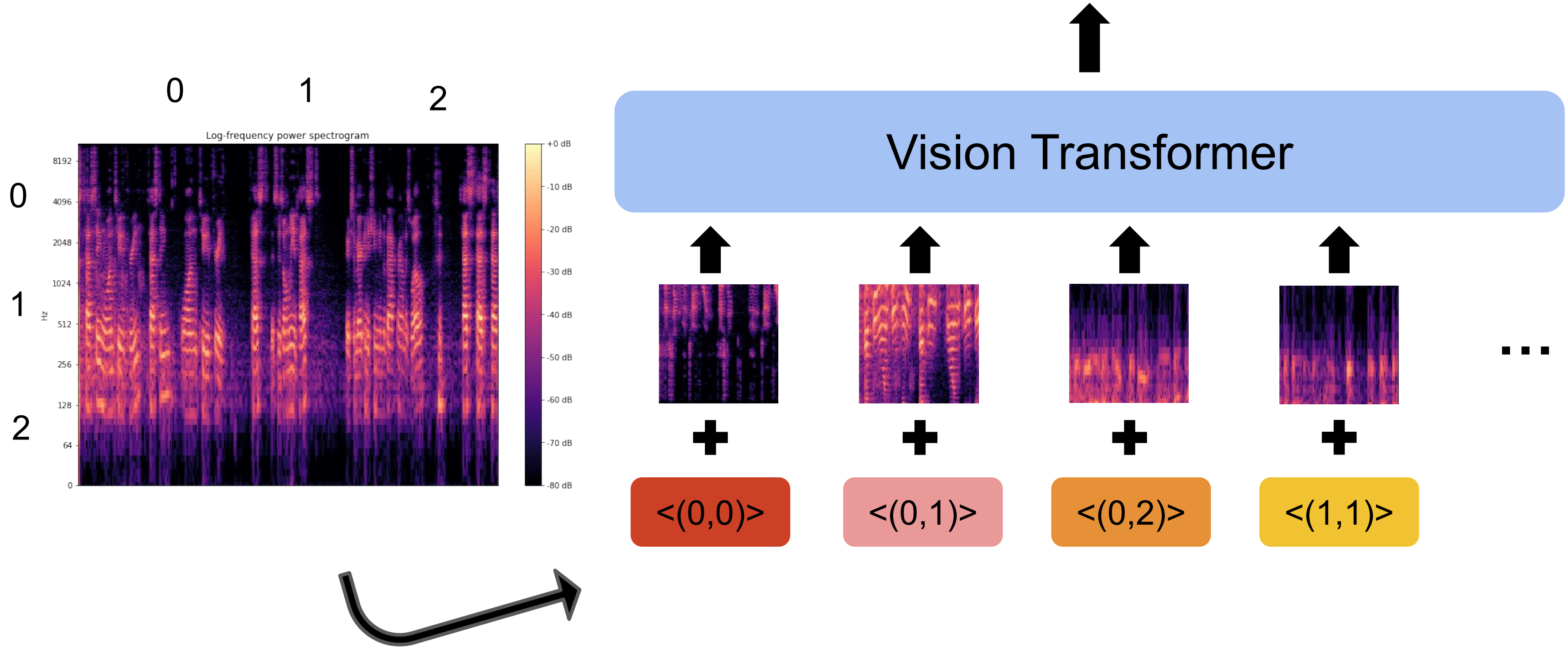


Spectrogram:

- Energy, pitch, fundamental frequency
- Decomposes signal into frequencies and their corresponding amplitudes

Spectrogram source: [Dumpala 2017]

Audio as a vision problem



Cornell Bowers CIS

Review

- Transformers can be used for vision tasks
  - Typically do not use convolution or other image specific architectures - just flattened image patches
  - Consequently they require more data
  - However they are very flexible and can be used with other data modalities
  - Learn positional embedding
- Swin transformers
  - Learn features at different scales
  - Scale linearly with the image size
  - Are less popular because of their complexity
- Self-supervised learning
  - Provides Transformer based models with the amount of data they need
  - Often used as backbone pre-trained models
  - Dino and MAE both learn very good embeddings
- Multi-modality
  - CLIP learns a common embedding for images and text
  - Can be used to retrieve images or parts of images from plain text