THE UNIVERSITY OF MELBOURNE

# Representation Learning

Semester 2, 2025

Kris Ehinger

# Outline

- Contrastive learning
- CLIP
- Visualisation and interpretation
- Invariance, generalisation

# Learning outcomes

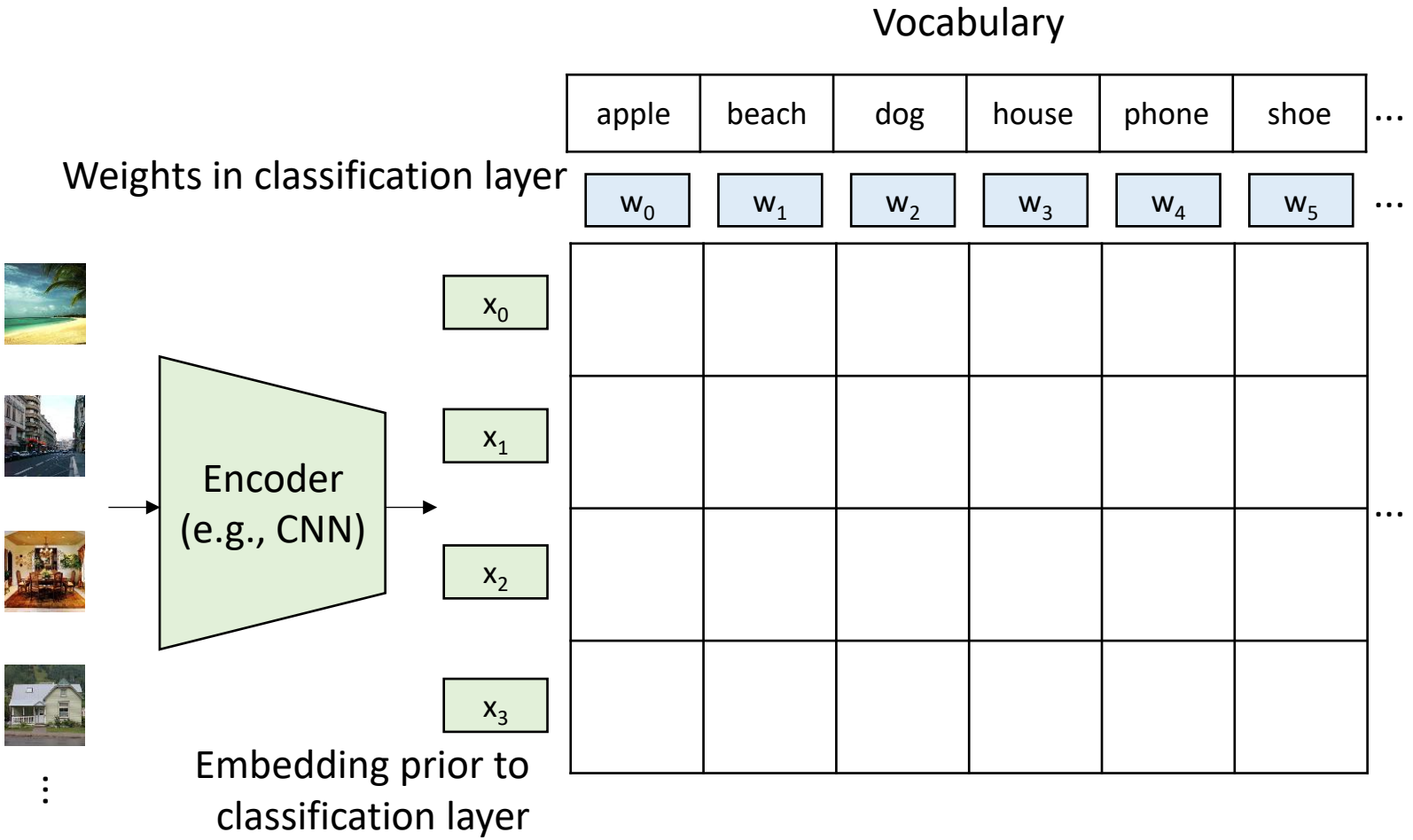
- Explain the concepts behind contrastive learning and CLIP
- Explain methods to visualise or explain the decisions of an image recognition algorithm
- Evaluate computer vision algorithms in terms of their invariance (or tolerance) to image variation

# Contrastive learning

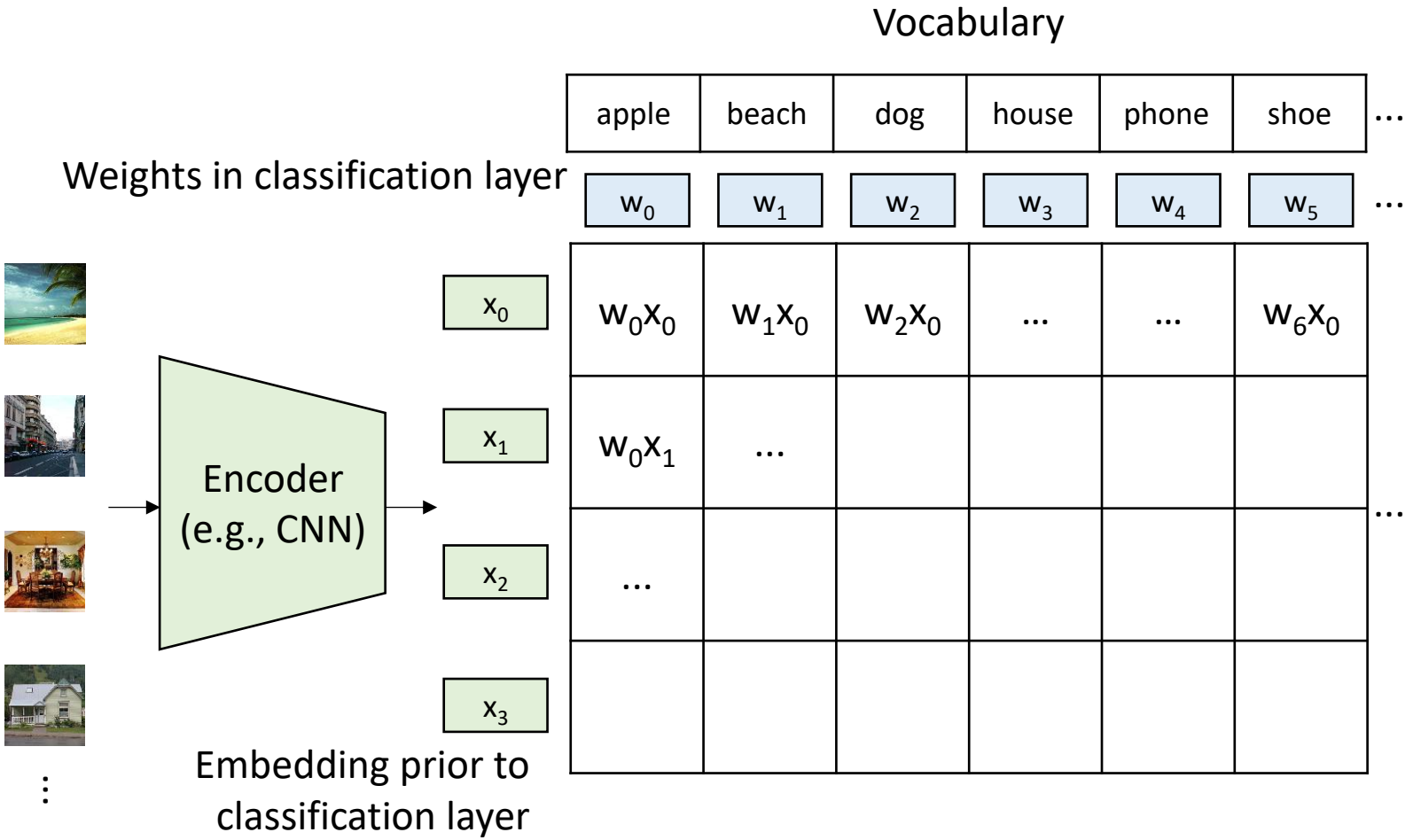
# Contrastive learning

- Self-supervised training method
- Recall:
  - Supervised = train from ground truth labels
  - Unsupervised = train without labels (e.g., clustering)
  - Self-supervised = generate "labels" from data itself
- Example: mask part of the image and train a model to predict the missing part
- Advantage: no need for manual annotation

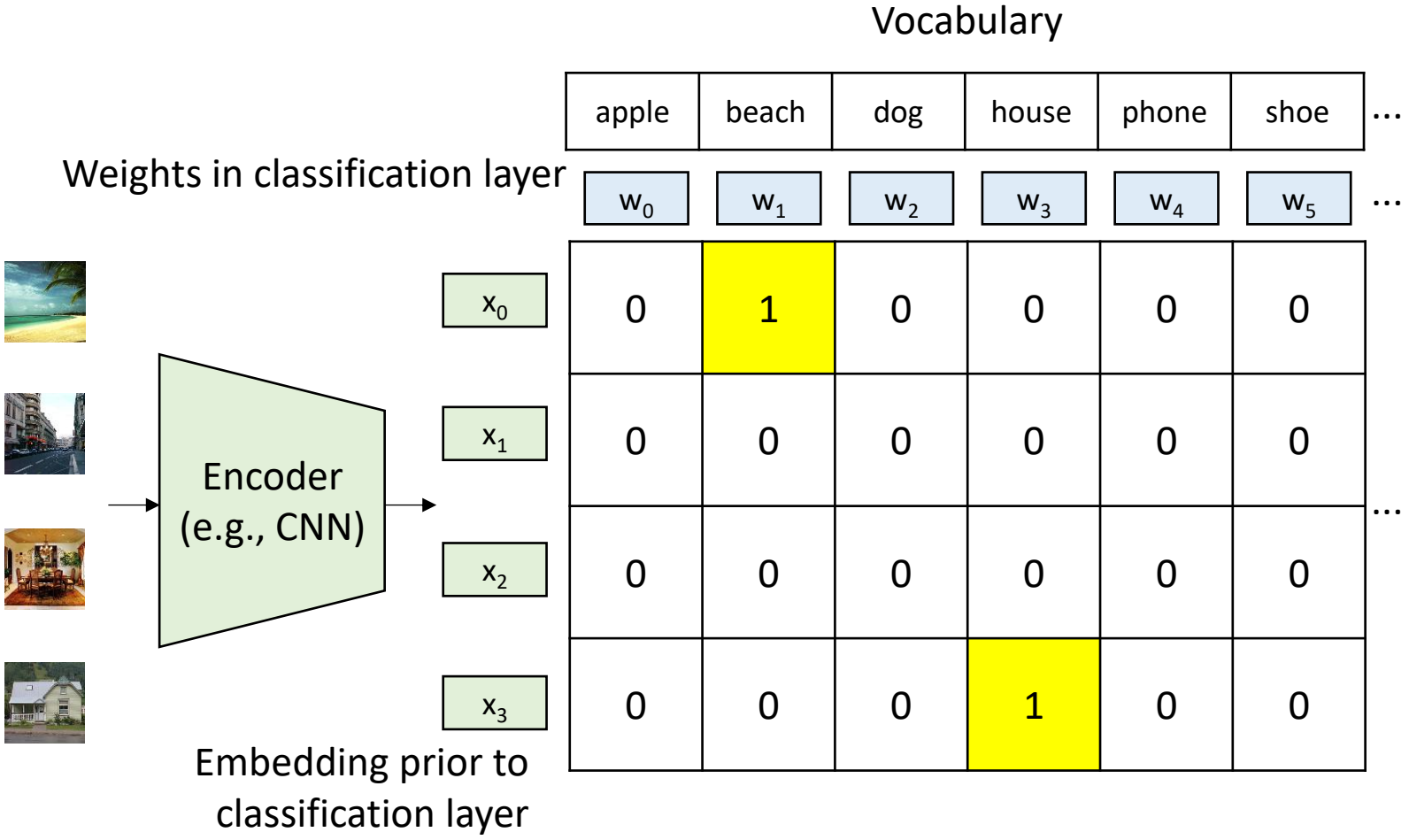
# Contrastive learning (w/ labels)



# Contrastive learning (w/ labels)



# Contrastive learning (w/ labels)

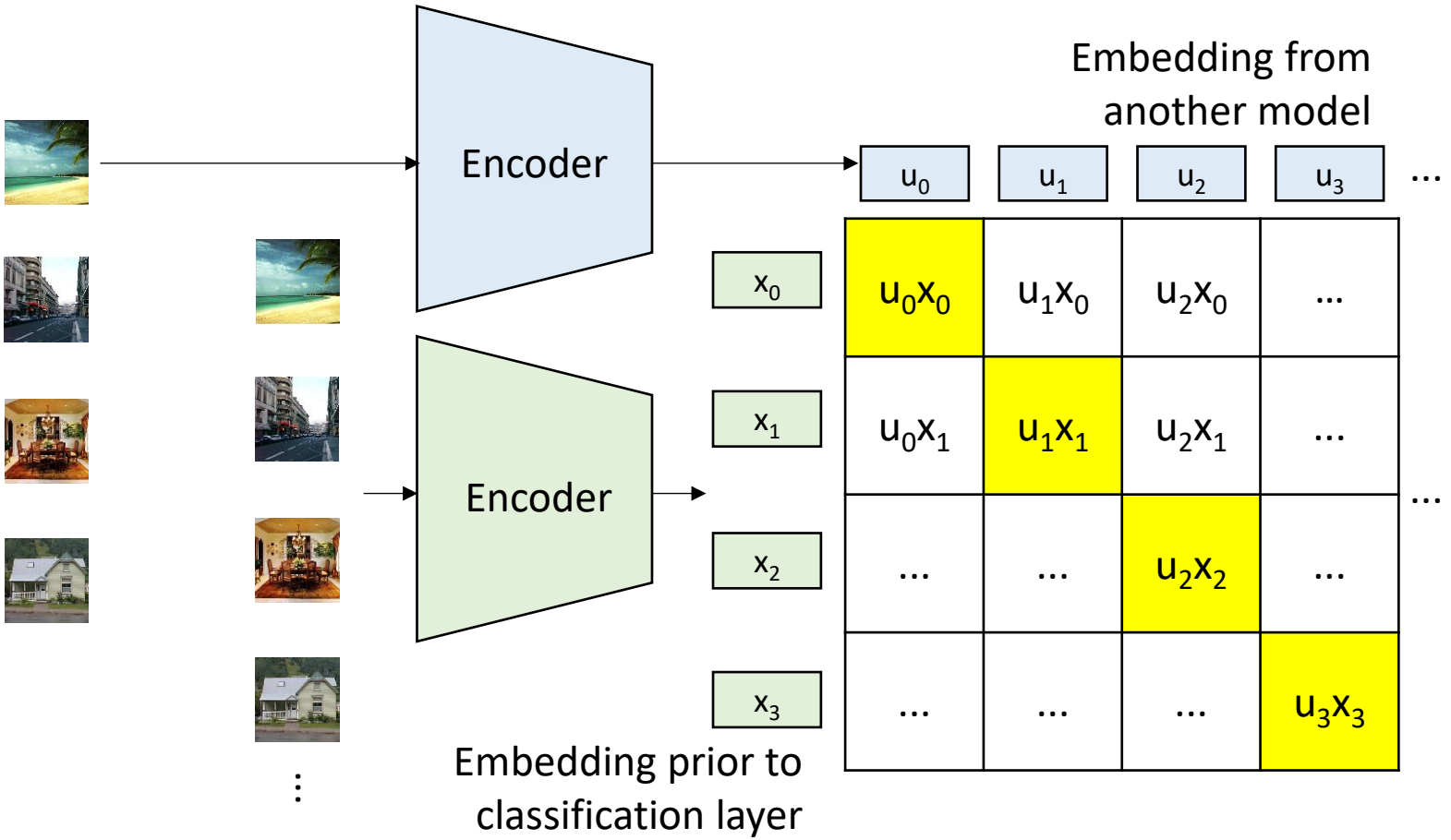


# Contrastive learning (w/ labels)

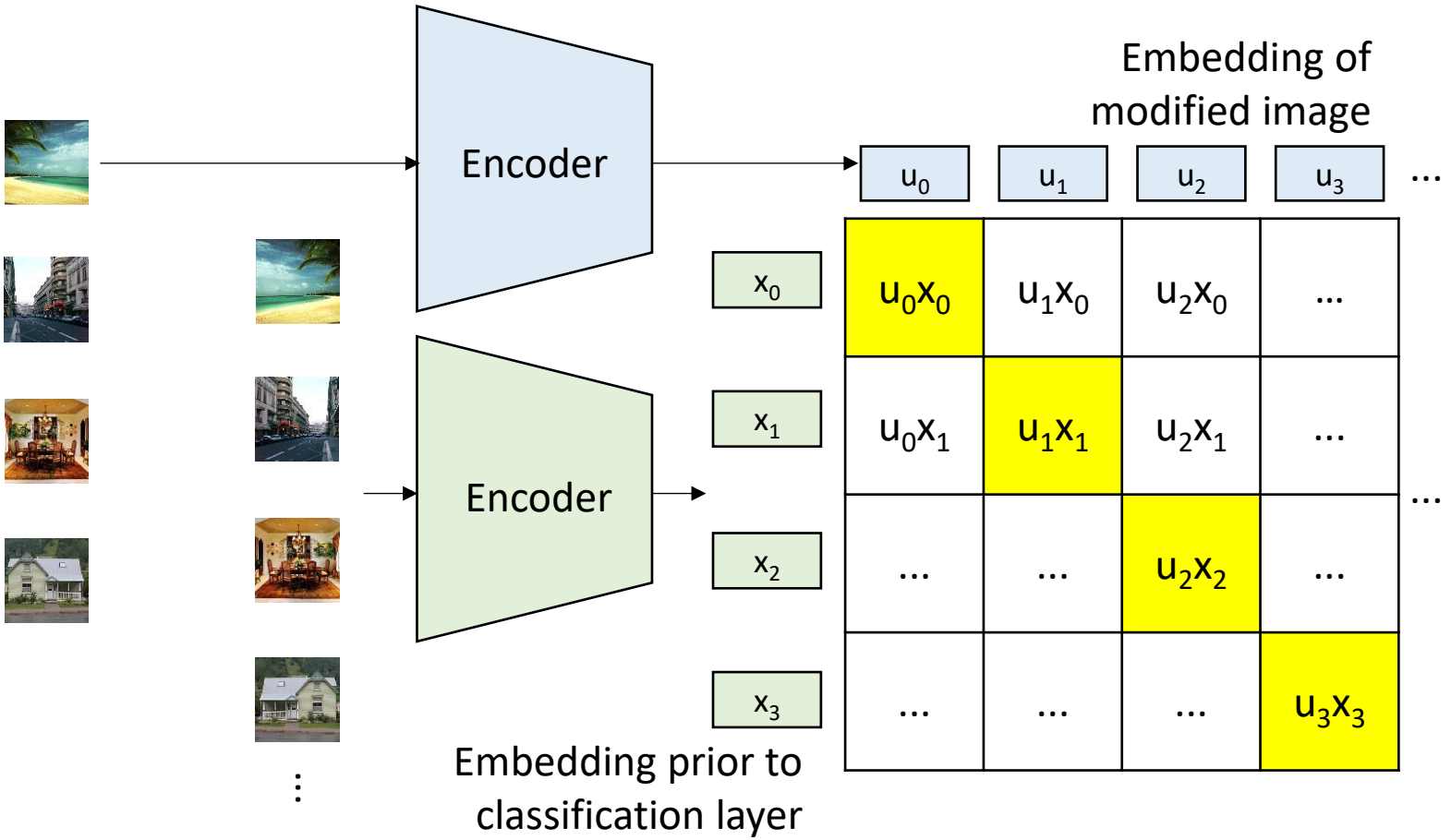
- Goal is to:
  - Maximize similarity of positive pairs
  - Minimize similarity of negative pairs

- What if you have no labels? Make the vocabulary from the data.

# Contrastive learning (no labels)

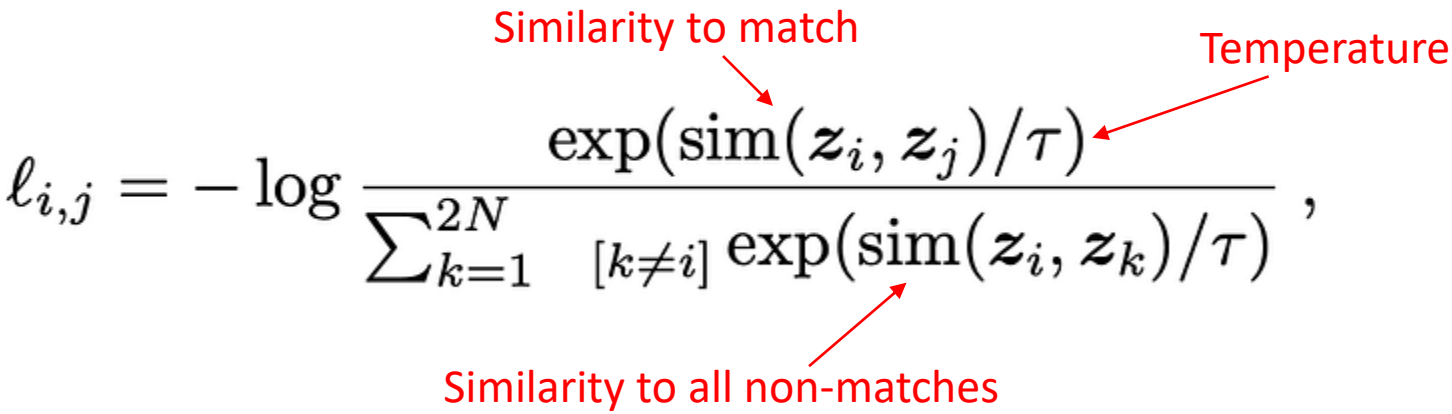


# Contrastive learning (no labels)



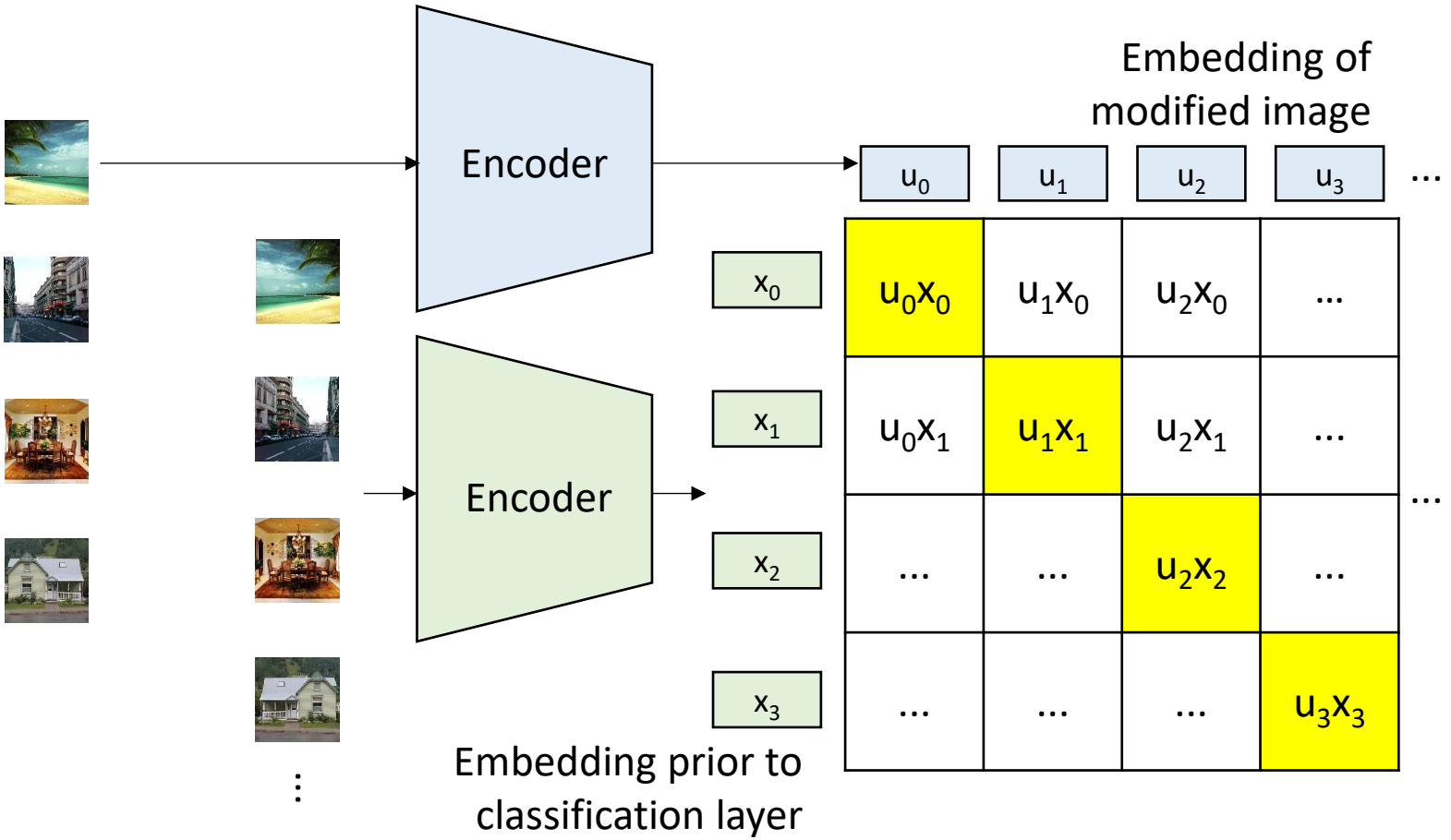
# Contrastive loss

- Contrastive loss function




- To reduce loss, increase similarity of matched pairs and decrease similarity of non-matched pairs
- Temperature = smoothness parameter (low value means similarity drops sharply)

# Contrastive learning (no labels)



# Contrastive learning

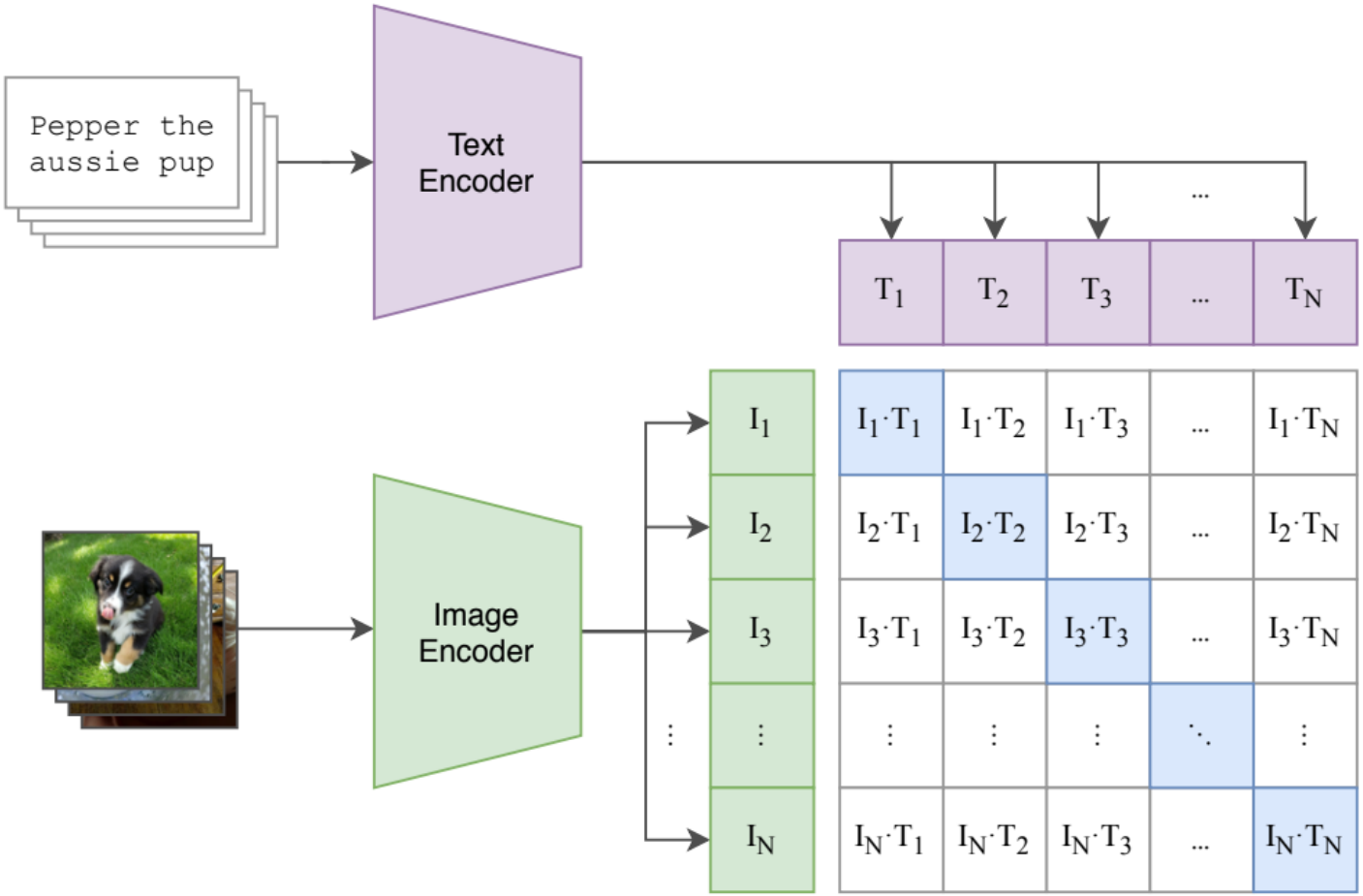
- Self-supervised learning: there is a training signal, but generated from the data itself (e.g., two variants of the same image)
- Many different implementations with different approaches to sampling examples, backpropagating loss

# CLIP

# CLIP

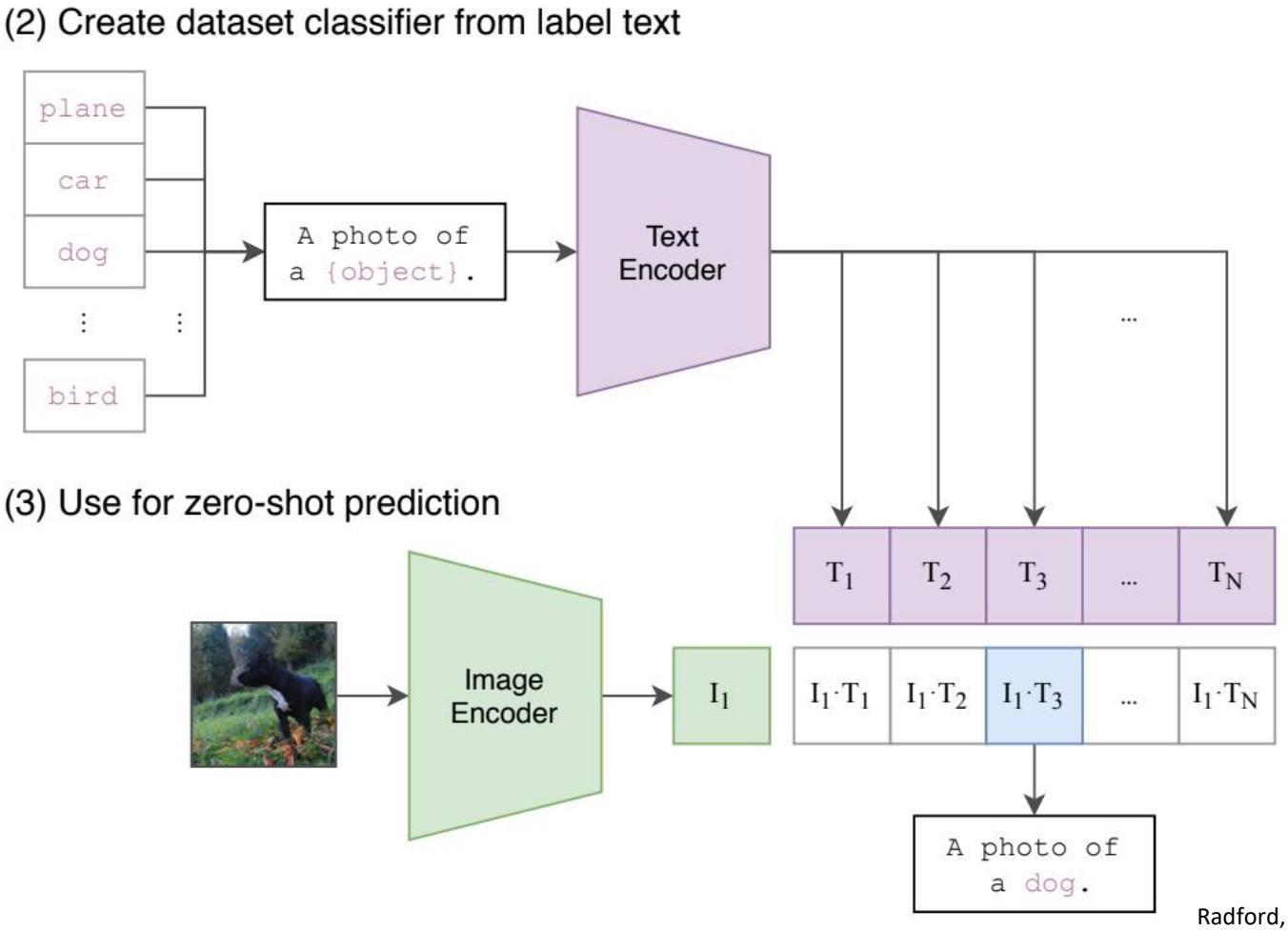
- CLIP = Contrastive Language-Image Pretraining
- Self-supervised learning on a large dataset of images + text captions
  - Reduces requirement for manual labelling of images
  - Wider variety of concepts that could be learned (not just 1000 object classes)

# CLIP




Radford, et al. (2021)

# Using CLIP as a classifier

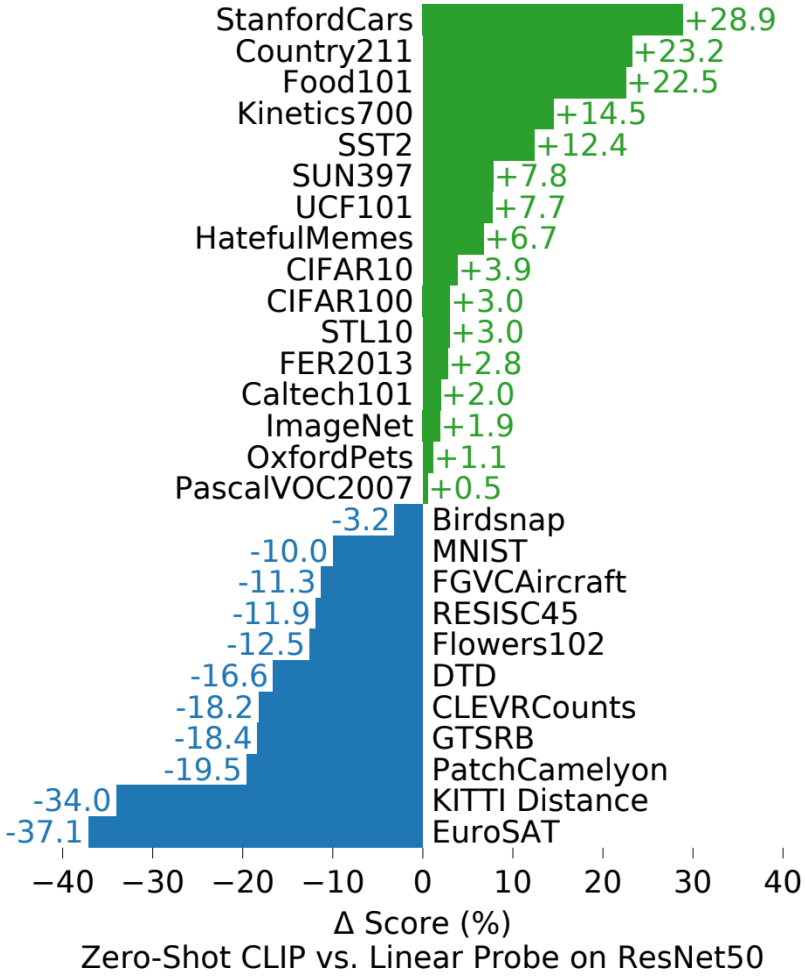



Radford, et al. (2021)

# CLIP

- Trained on WIT, proprietary dataset of 400 million image-text pairs (mostly from internet?)
- Train on a batch of N image-text pairs:
  - $N^2$ possible pairs
  - N are true matches
  - $N^2$-N are false matches
- Measure cosine similarity between image and text embedding
- Contrastive learning with multi-class N-pair loss

# Zero-shot classification

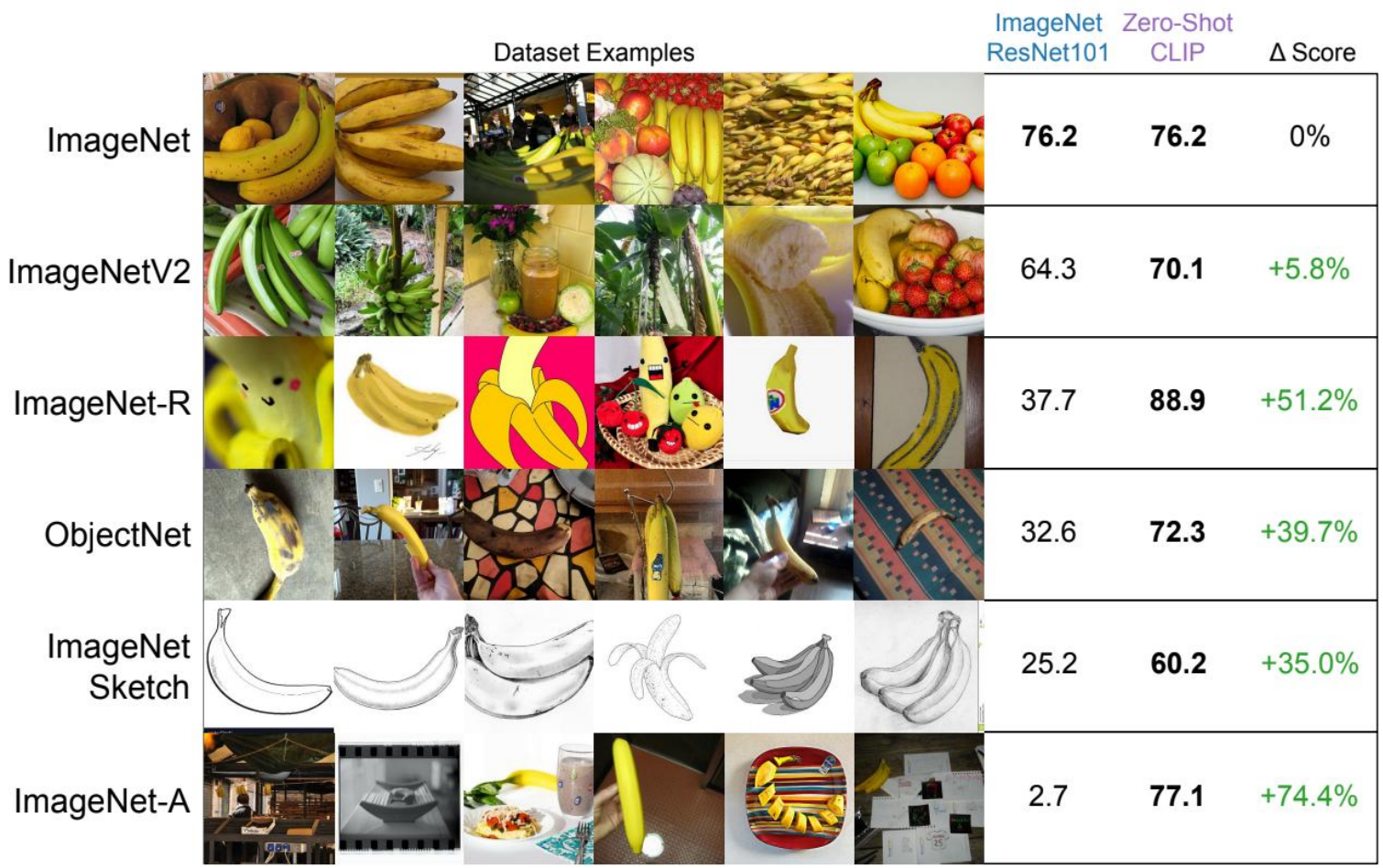



CLIP with no additional training (right) vs. ResNet50 trained on this task/dataset (left)

CLIP outperforms ResNet50

Radford, et al. (2021)

# Robustness of CLIP



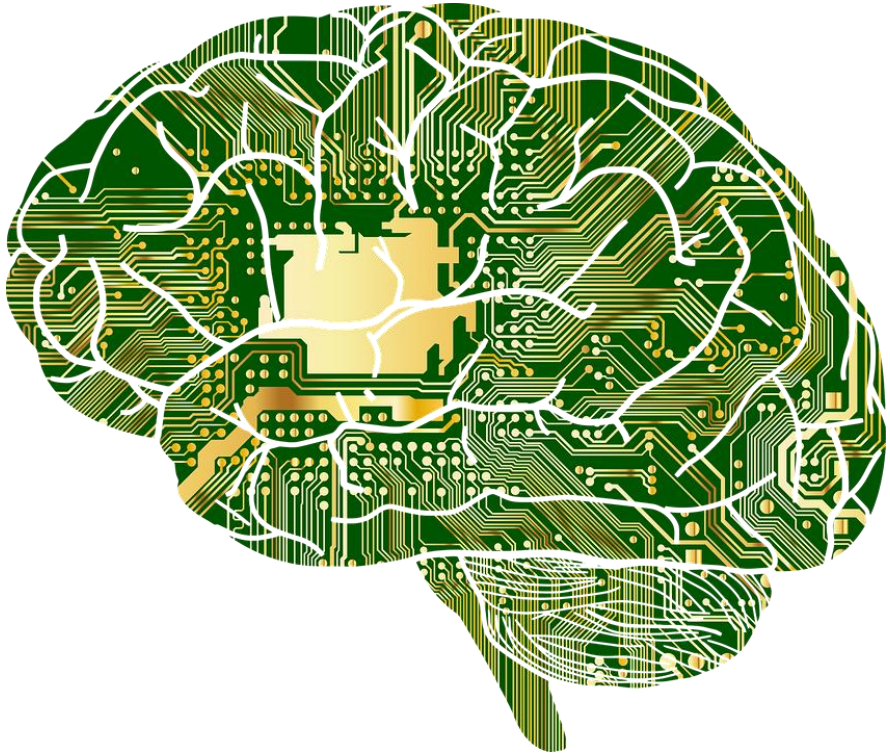

Radford, et al. (2021)

# Summary

- CLIP is a particularly useful model for generating embeddings:
  - Trained on a very large image dataset (400M)
  - Can generate text pseudo-labels due to shared embedding space
- Common uses:
  - Auto-labelling large image datasets (e.g., LAION)
  - Embeddings for LLMs

# Model visualisation

# Understanding vision models

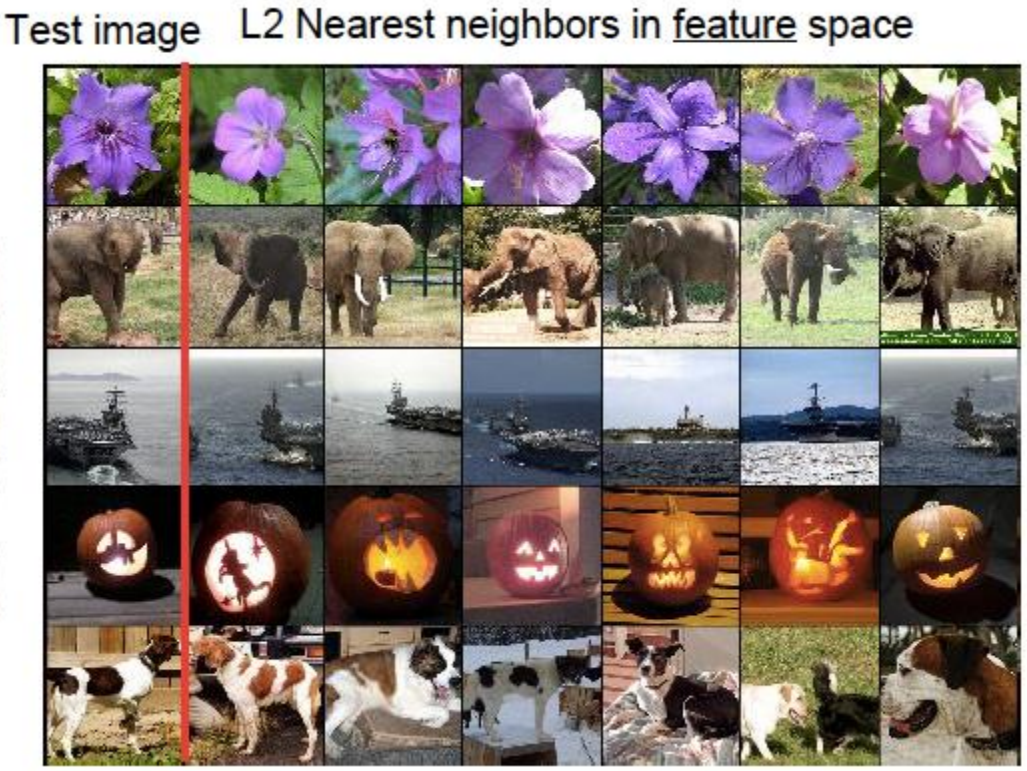


How do we know what a computer vision model is thinking?

# Visualising embeddings

- How are images organised in this feature space?
- It's a high-dimensional space, so we can't just plot all images in this space
  - Can use dimensionality reduction
  - Or look at local regions (what images are near neighbours in this space?)

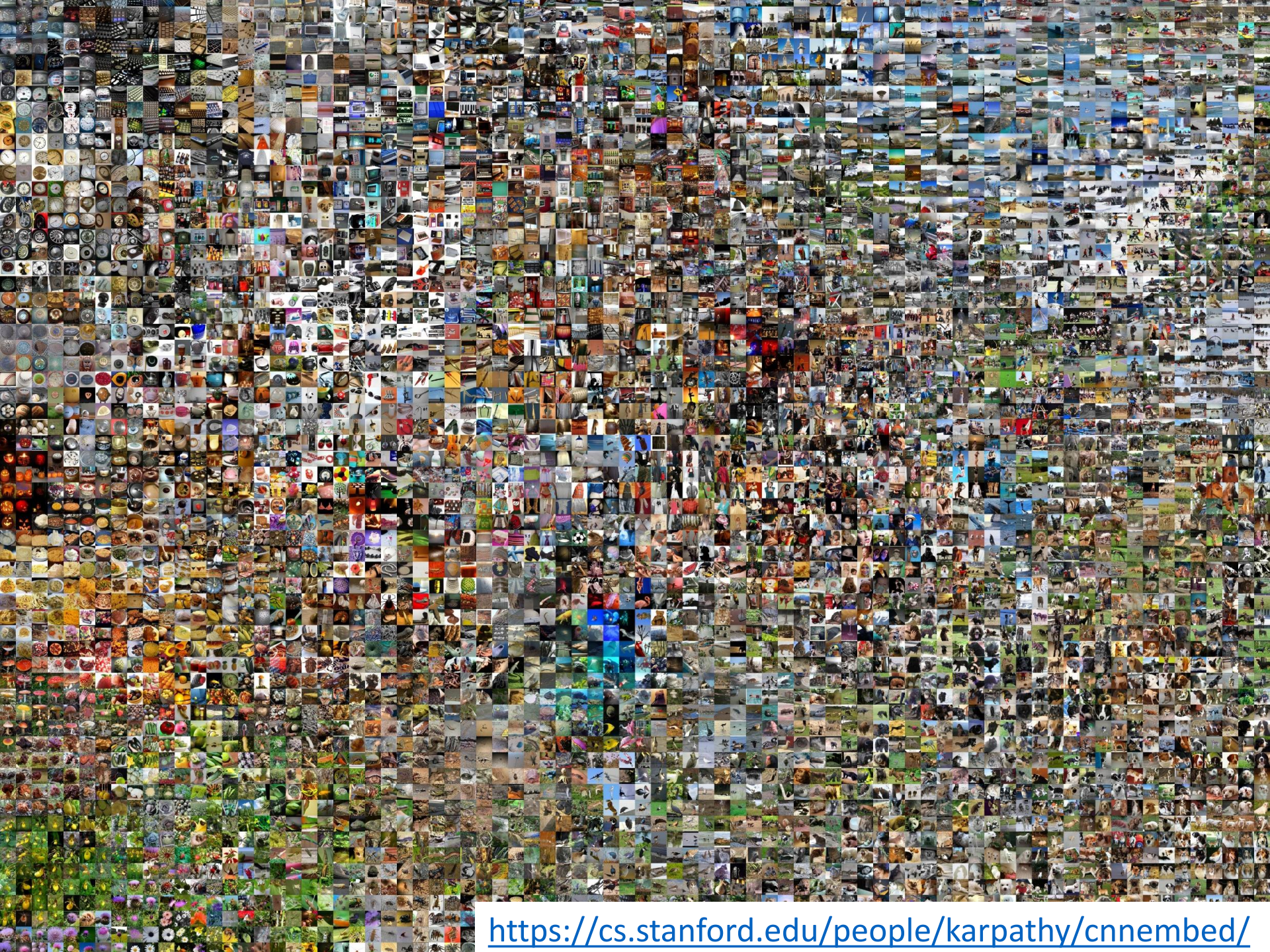
# Nearest neighbours


Krizhevsky, Sutskever, & Hinton (2012)

# Feature space visualisation

- Options for dimensionality reduction
- PCA (principal component analysis)
  - Show the dimensions with the most variance
  - Simple but often hard to interpret since only a few dimensions can be visualised simultaneously
- t-SNE (t-distributed stochastic neighbor embedding)
  - Flatten high-dimensional data into 2D or 3D so that near neighbours stay nearby



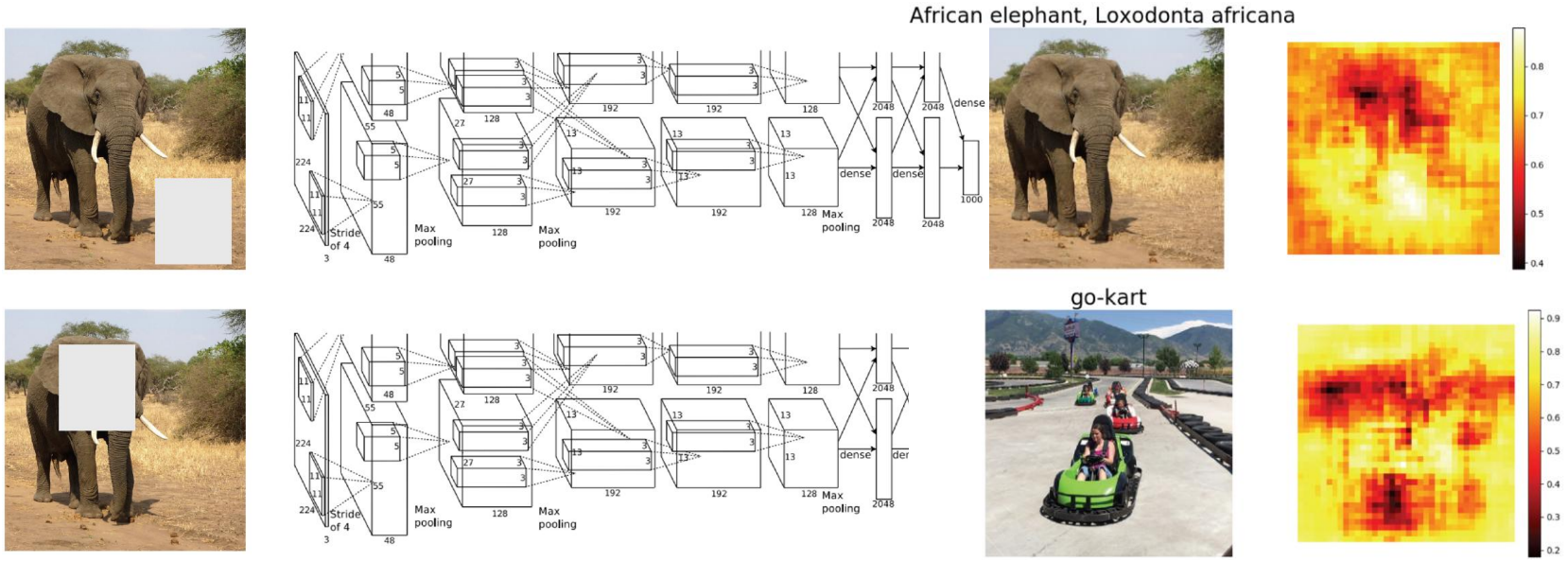
https://cs.stanford.edu/people/karpathy/cnnembed/

# "Where" is the model looking?

- What parts of an image are most important for determining the class label?
  - Can help show what features the model uses to determine class
  - Can help debug problems (e.g., using background to classify object, label confusion when there are multiple objects)

# Visualising image regions

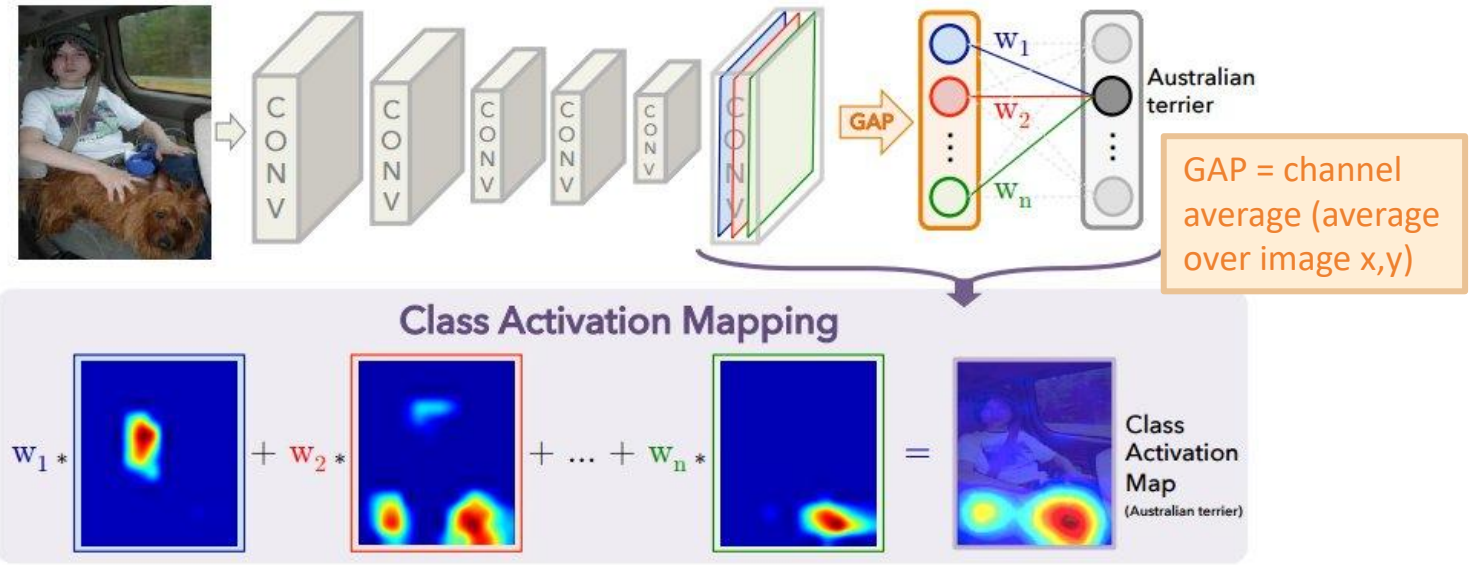
- Occlusion method: mask image and see how much class probability changes




Zeiler & Fergus (2014)

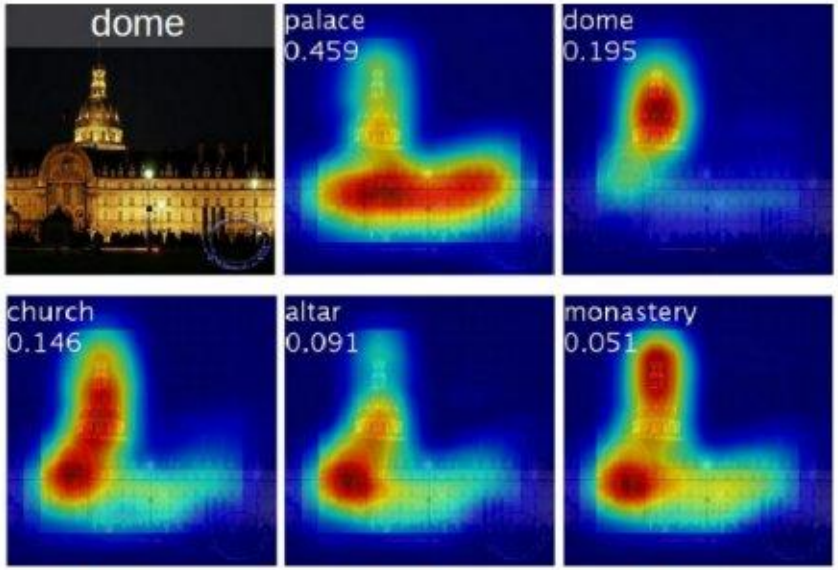
# CAM (Class Activation Mapping)

- Add a Global Average Pooling (GAP) layer before classification layer, use weights of this layer to determine *where* the class-relevant features are




Zhou, et al. (2016)

# CAM (Class Activation Mapping)




Class activation maps of top 5 predictions


Class activation maps for one object class

# Grad-CAM

(a) Original Image (b) Guided Backprop 'Cat' (c) Grad-CAM 'Cat'

(g) Original Image (h) Guided Backprop 'Dog' (i) Grad-CAM 'Dog'

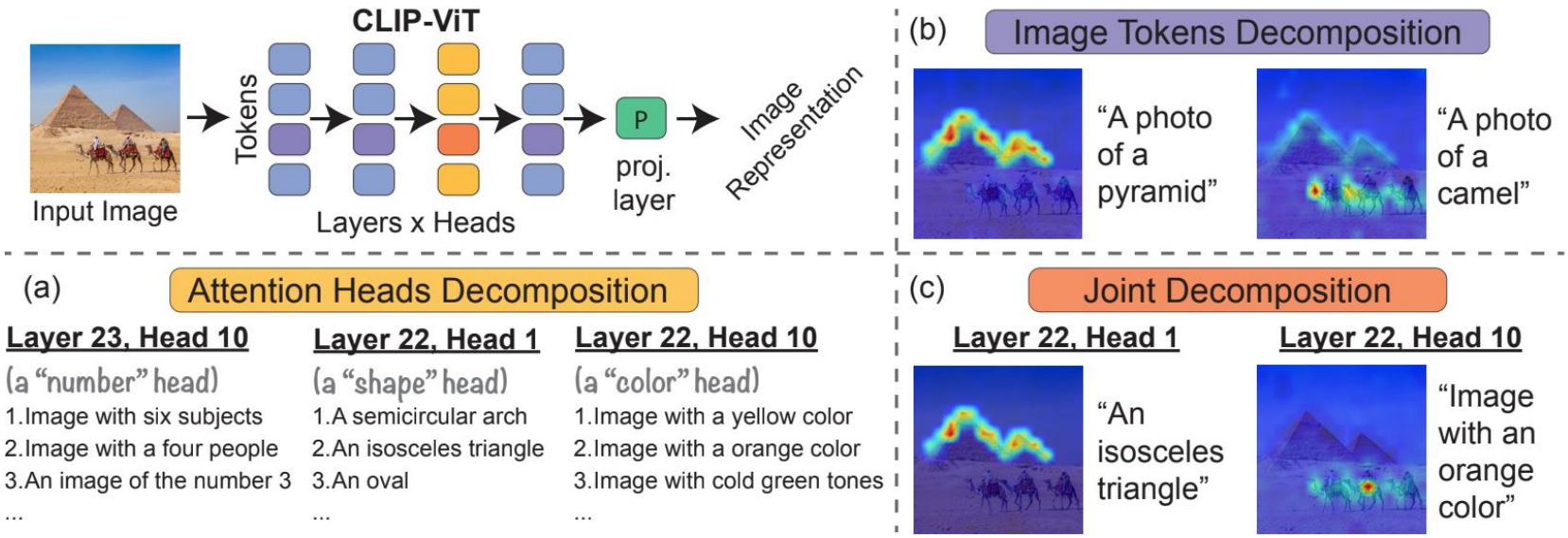
Selvaraju, et al. (2016)

# "What" does the model encode?

- What features does a particular (layer/neuron/attention head/...) encode?
- Visualise the activation maps, filters, or show images that produce a high response
  - Or, for models like CLIP, output text that produces a high response

# "What" does the model encode?

Break down response into contributions of each attention head ("what") and patch location ("where"):




Gandelsman, Efros, & Steinhardt (2024)

# "What" does the model encode?

| **L21.H11** ("Geo-locations") | **L23.H10** ("Counting") | **L22.H8** ("Letters") |
|---|---|---|
| Photo captured in the Arizona desert | Image with six subjects | A photo with the letter V |
| Picture taken in Alberta, Canada | Image with four people | A photo with the letter F |
| Photo taken in Rio de Janeiro, Brazil | An image of the number 3 | A photo with the letter D |
| Picture taken in Cyprus | An image of the number 10 | A photo with the letter T |
| Photo taken in Seoul, South Korea | The number fifteen | A photo with the letter X |
| **L22.H11** ("Colors") | **L22.H6** ("Animals") | **L22.H3** ("Objects") |
| A charcoal gray color | Curious wildlife | An image of legs |
| Sepia-toned photograph | Majestic soaring birds | A jacket |
| Minimalist white backdrop | An image with dogs | A helmet |
| High-contrast black and white | Image with a dragonfly | A scarf |
| Image with a red color | An image with cats | A table |
| **L23.H12** ("Textures") | **L22.H1** ("Shapes") | **L22.H2** ("Locations") |
| Artwork with pointillism technique | A semicircular arch | Urban park greenery |
| Artwork with woven basket design | An isosceles triangle | Cozy home interior |
| Artwork featuring barcode arrangement | An oval | Urban subway station |
| Image with houndstooth patterns | Rectangular object | Energetic street scene |
| Image with quilted fabric patterns | A sphere | Tranquil boating on a lake |

Gandelsman, Efros, & Steinhardt (2024)

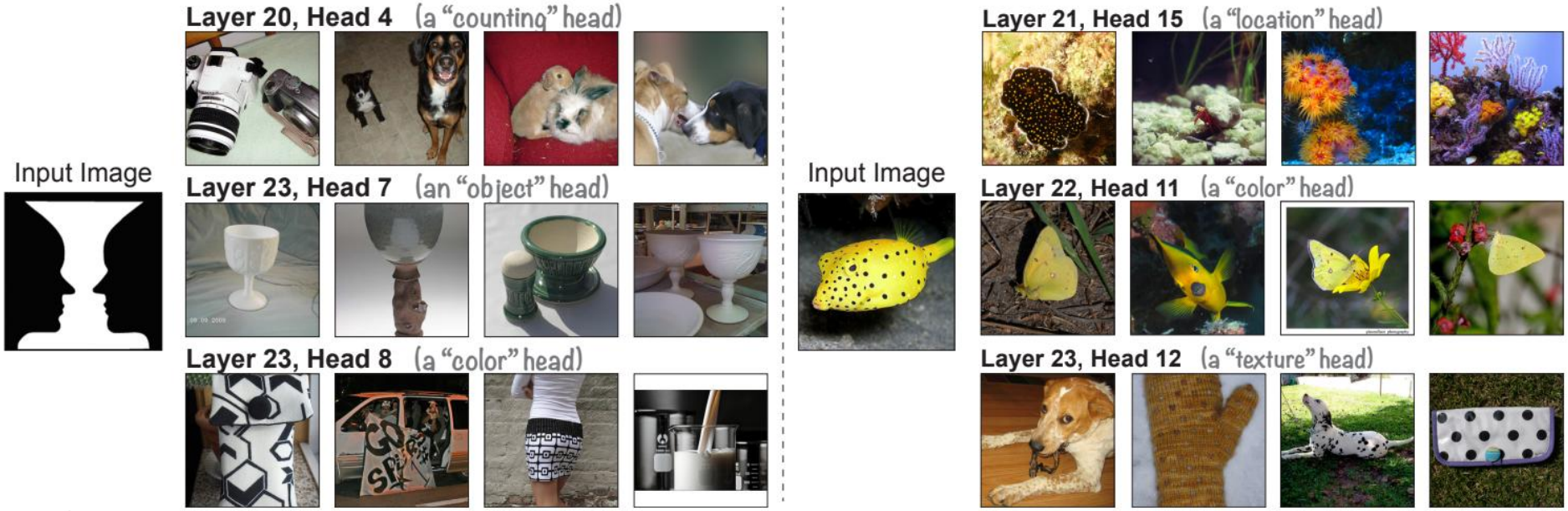
# "What" does the model encode?

| Layer 23, Head 0 | Layer 23, Head 1 |
|---|---|
| Intrica wood carvingte | Photograph taken in a retro diner |
| Nighttime illumination | Intense athlete |
| Image with woven fabric design | Detailed illustration of a futuristic bioreactor |
| Image with shattered glass reflections | Image with holographic retro gaming aesthetics |
| A photo of food | Antique historical artifact |
| **Layer 23, Head 2** | **Layer 23, Head 3** |
| Image showing prairie grouse | Bustling city square |
| Image with a penguin | Serene park setting |
| A magnolia | Warm and cozy indoor scene |
| An image with dogs | Modern airport terminal |
| An image with cats | Remote hilltop hut |
| **Layer 23, Head 4** | **Layer 23, Head 5** |
| Playful siblings | Intertwined tree branches |
| A photo of a young person | Flowing water bodies |
| Image with three people | A meadow |
| A photo of a woman | A smoky plume |
| A photo of a man | Blossoming springtime blooms |

Gandelsman, Efros, & Steinhardt (2024)

# "What" does the model encode?

Nearest neighbors of an input image, according to different attention heads:

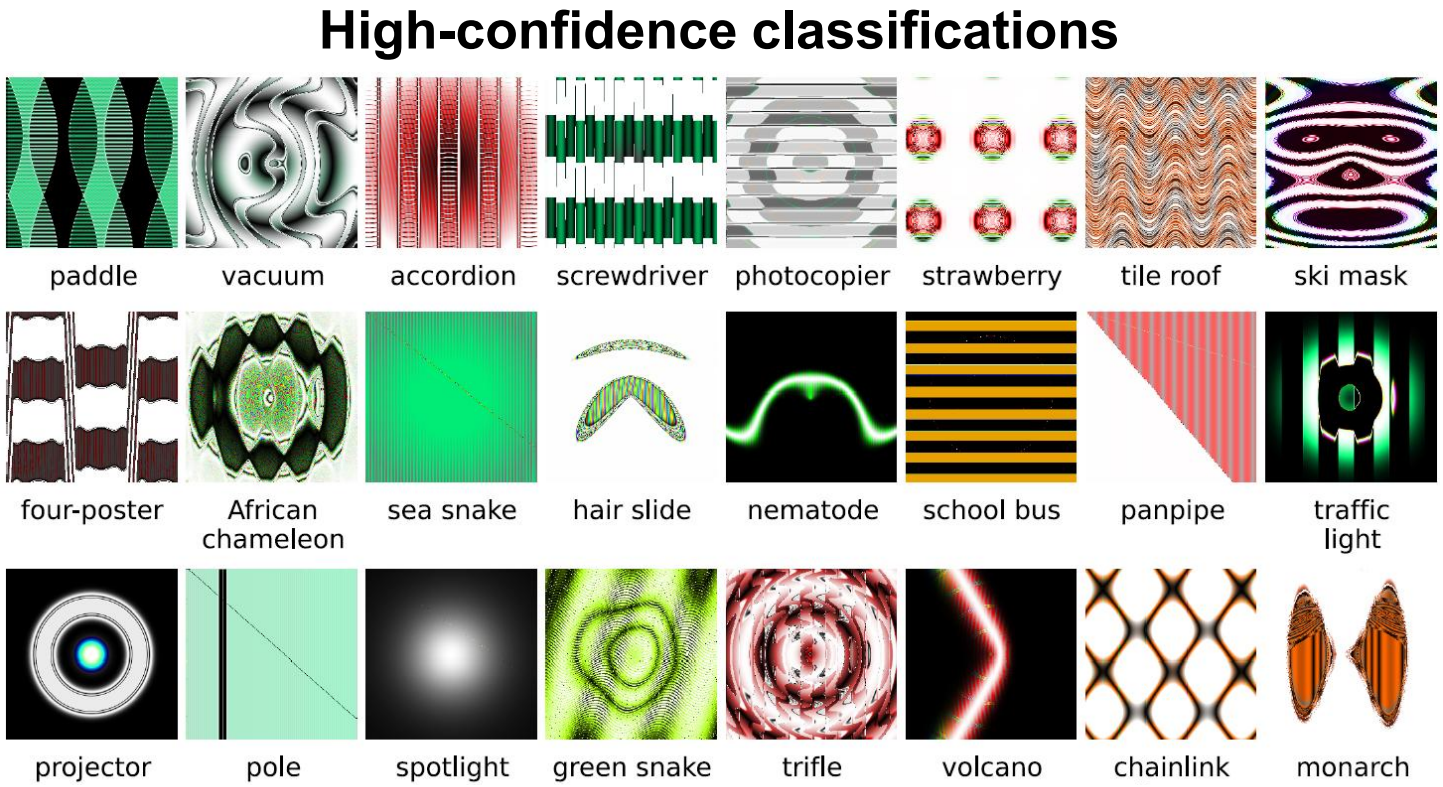



Gandelsman, Efros, & Steinhardt (2024)

# Visualising classes

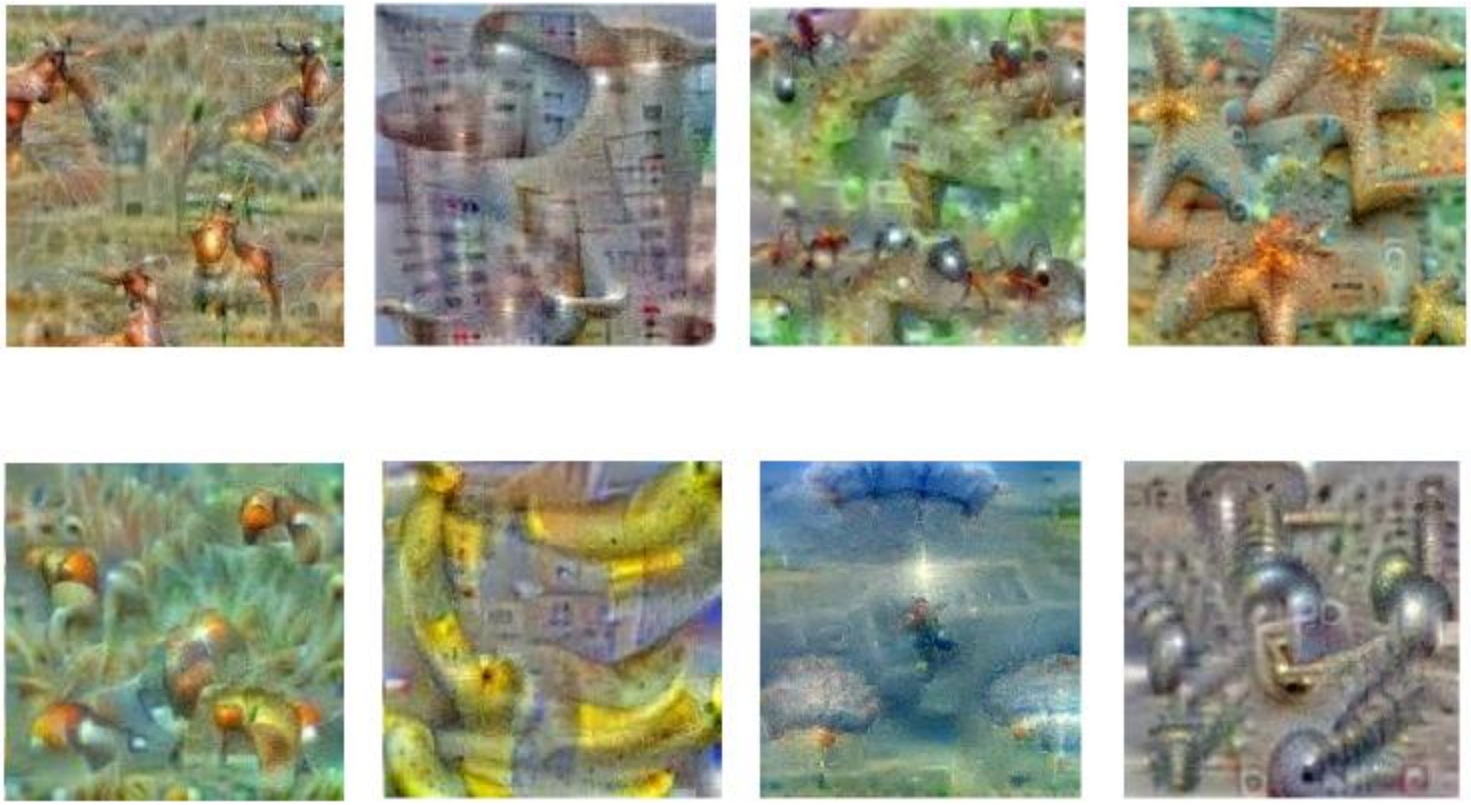
- Usually based on gradient ascent – synthesize image that maximises class label response
  - Initialise an image with zeros or small random noise
  - Run image through network, compute gradient
  - Update *image pixels* in a direction that minimises loss
- Problem: there are many possible arrays of pixels that can generate very high model response; not all of these will look like realistic images

# Visualising classes




Nguyen, Yosinski, & Clune (2015)

# Visualising classes



Google AI Blog: Mordvintsev, Olah, & Tyka (2015)
https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html

# Visualising classes


Nguyen, et al. (2016)

# Summary

- Various ways to visualise what a model is doing
  - Visualise the feature space
  - Visualise the classes
  - Check where in the image a model is looking
  - Check what each part of a model (e.g., neuron or attention head) represents

# Invariance & generalisation

# Invariance / tolerance

- Are the features learned by CNNs invariant to
  - Lighting?
  - Translation?
  - Image plane rotation?
  - Scale?
  - 3D rotation / pose?

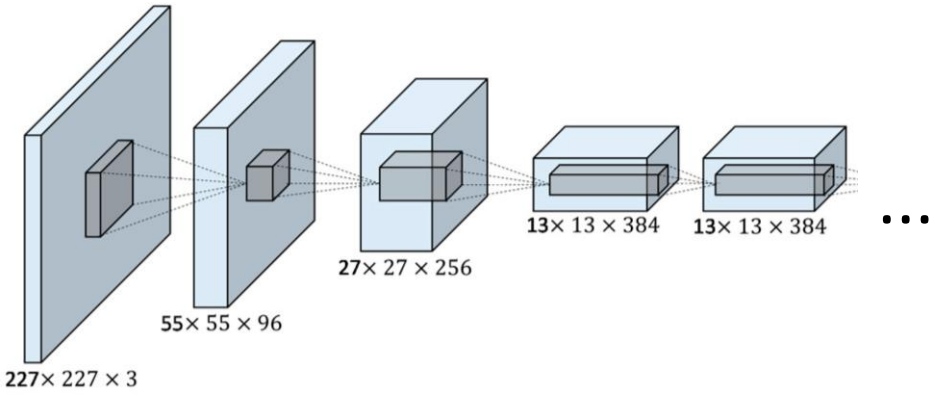
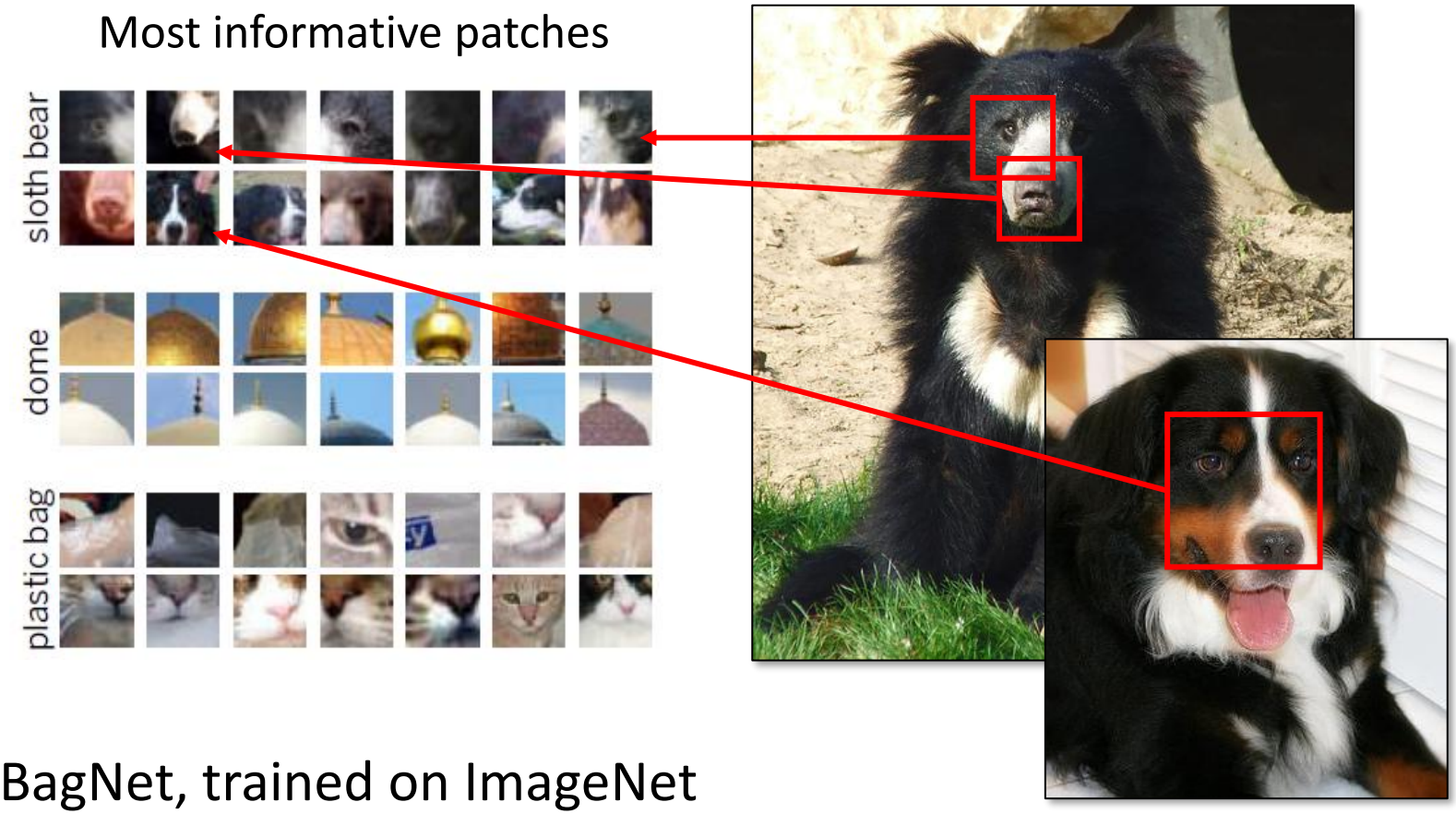



Image: Han, Zhong, Cao, & Zhang (2017)

# Visualising classes




BagNet, trained on ImageNet

Brendel & Bethge (2019)

# Shape and texture


original


greyscale

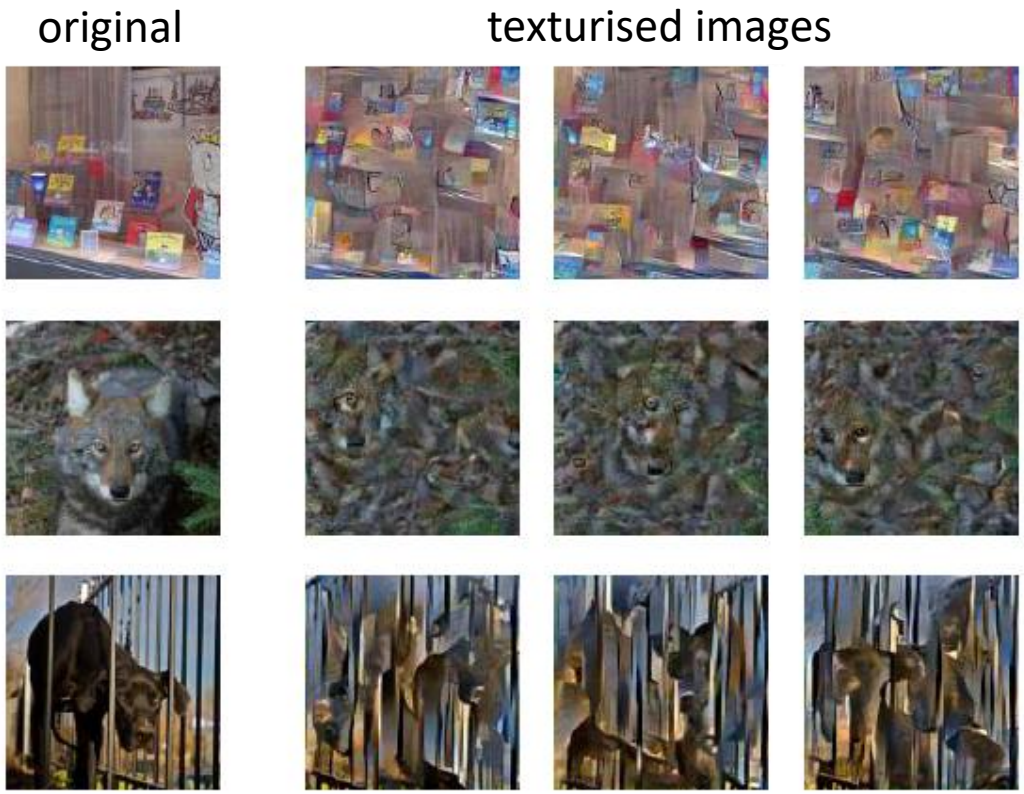

silhouette


edges


texture

Geirhos, Rubisch, Michaelis, Bethge, Wichmann, & Brendel (2019)

# Shape and texture




VGG-16, trained on ImageNet

Performance drop: 90% → 79%

Image: Brendel & Bethge (2019)

# Generalisation

- Models are very sensitive to some types of noise

Train on regular images: 

Can recognize: 

Can't recognize:

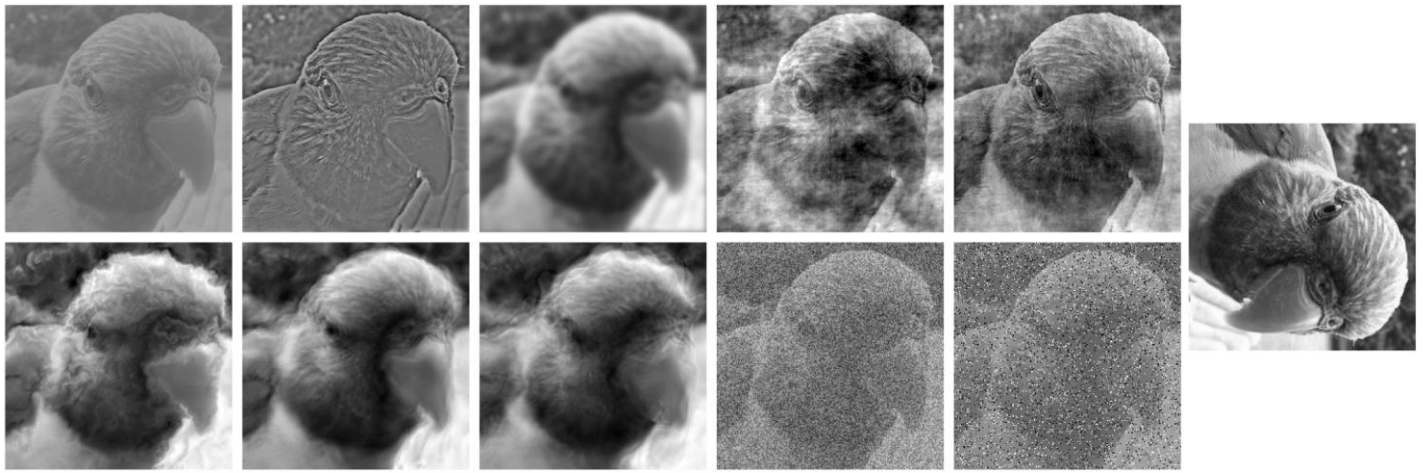
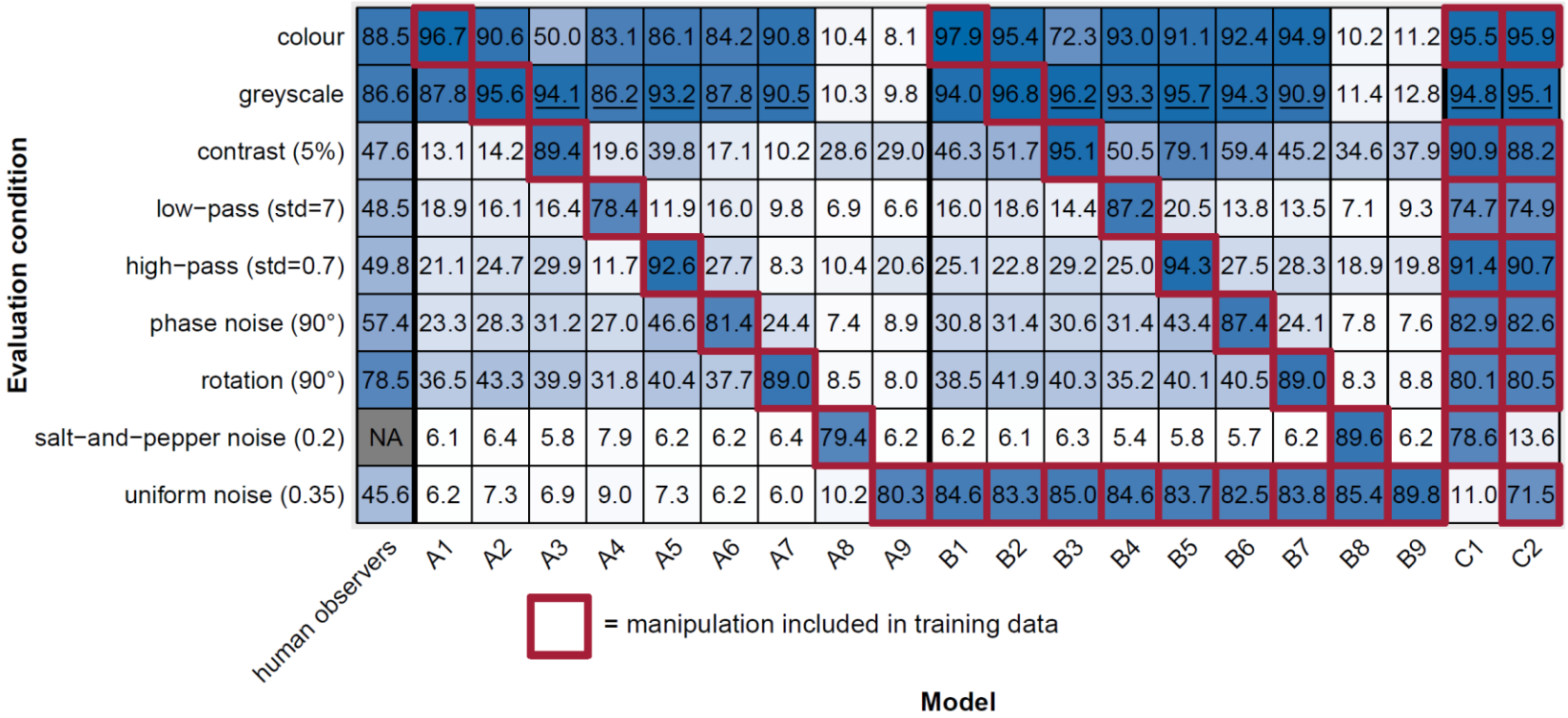


Image: Geirhos, Temme, Rauber, Schütt, Bethge, & Wichmann (2018)

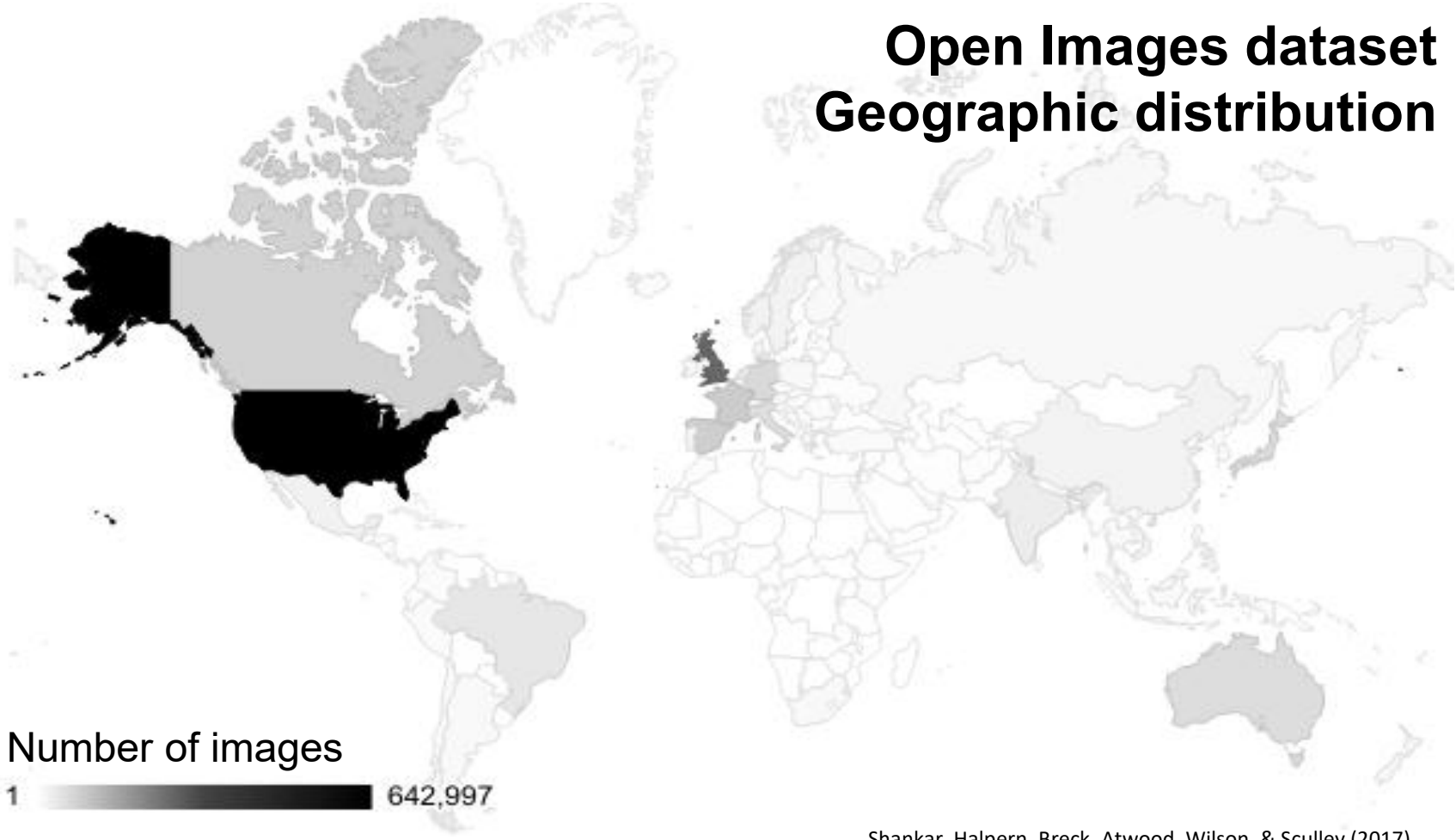
# Generalisation




ResNet-50 retrained on ImageNet subset with various manipulations

Image: Geirhos, Temme, Rauber, Schütt, Bethge, & Wichmann (2018)

# Cultural bias?




Shankar, Halpern, Breck, Atwood, Wilson, & Sculley (2017)

**Ground truth: Soap** **Nepal, 288 $/month**

**Azure**: food, cheese, bread, cake, sandwich
**Clarifai**: food, wood, cooking, delicious, healthy
**Google**: food, dish, cuisine, comfort food, spam
**Amazon**: food, confectionary, sweets, burger
**Watson**: food, food product, turmeric, seasoning
**Tencent**: food, dish, matter, fast food, nutriment


**Ground truth: Soap** **UK, 1890 $/month**

**Azure**: toilet, design, art, sink
**Clarifai**: people, faucet, healthcare, lavatory, wash closet
**Google**: product, liquid, water, fluid, bathroom accessory
**Amazon**: sink, indoors, bottle, sink faucet
**Watson**: gas tank, storage tank, toiletry, dispenser, soap dispenser
**Tencent**: lotion, toiletry, soap dispenser, dispenser, after shave

DeVries, Misra, Wang, & van der Maalen (2019)

# Invariance / tolerance

- CNNs are tolerant to variation included in the training data
- But may not generalise well to variation outside their training distribution
- SOTA image training datasets have biases (geographic, wealth) which impact generalisation