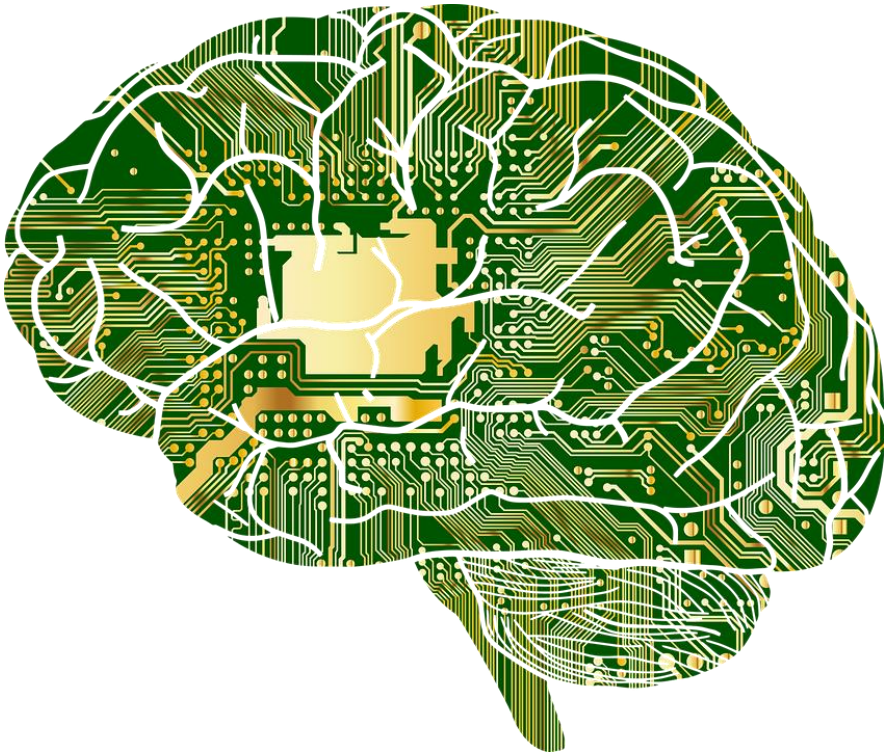
THE UNIVERSITY OF MELBOURNE

Deep Learning II

Semester 2, 2022

Kris Ehinger

Understanding CNNs



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

Outline

- ImageNet results
- CNN visualisation
- Invariance, generalisation

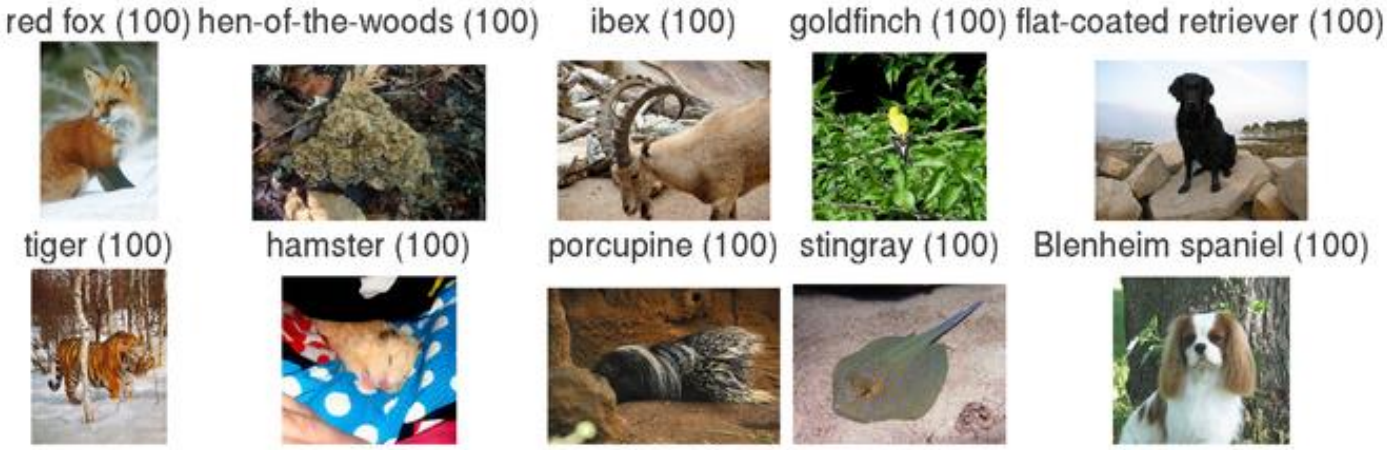
Learning outcomes

- Explain methods that can be used to evaluate computer vision algorithms
- Explain strengths and weaknesses of CNNs for image recognition
- Evaluate computer vision algorithms in terms of their invariance (or tolerance) to image variation

ImageNet results

What affects performance?

Easiest classes:



Hardest classes:

Russakovsky et al. (2014)

Model visualisation

Feature representation

"AlexNet": Krizhevsky, Sutskever, & Hinton (2012)

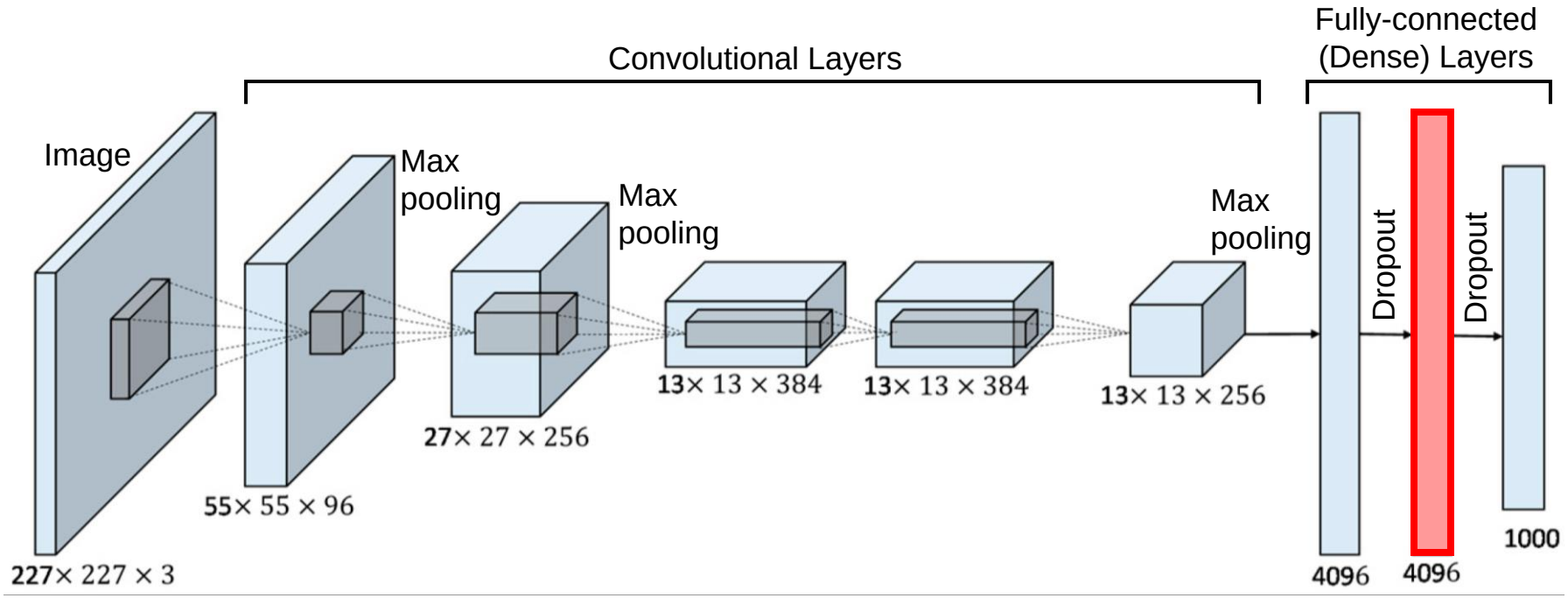
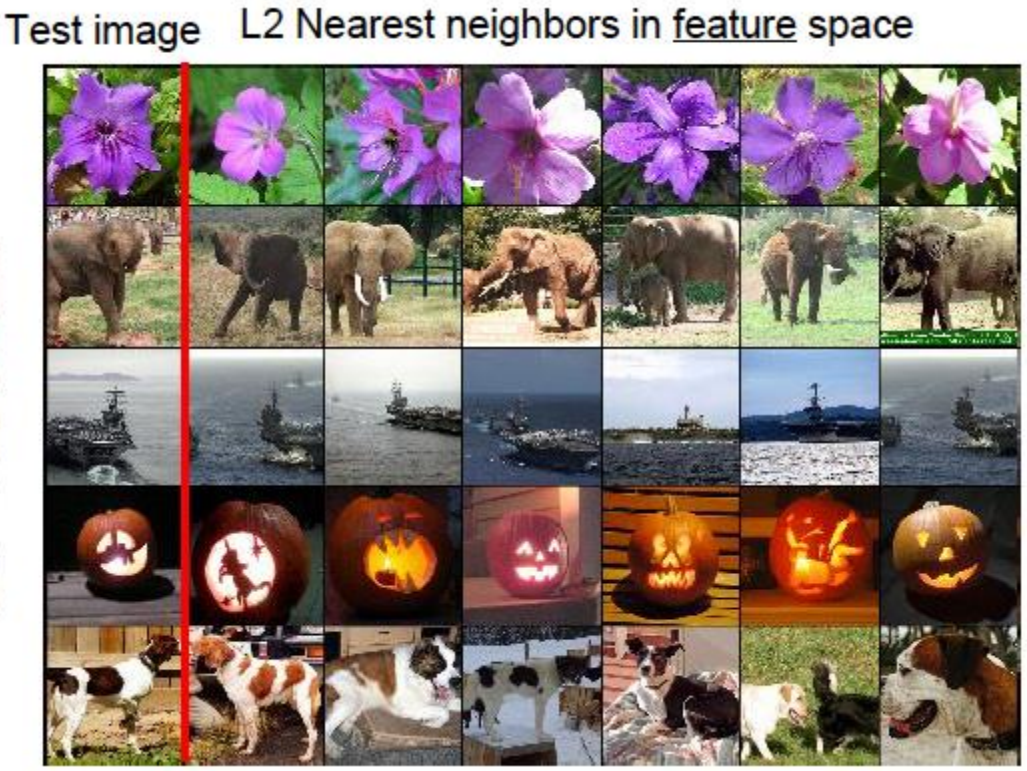


Image: Han, Zhong, Cao, & Zhang (2017)

Visualising features

- 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?)
- What do individual neuron responses represent?

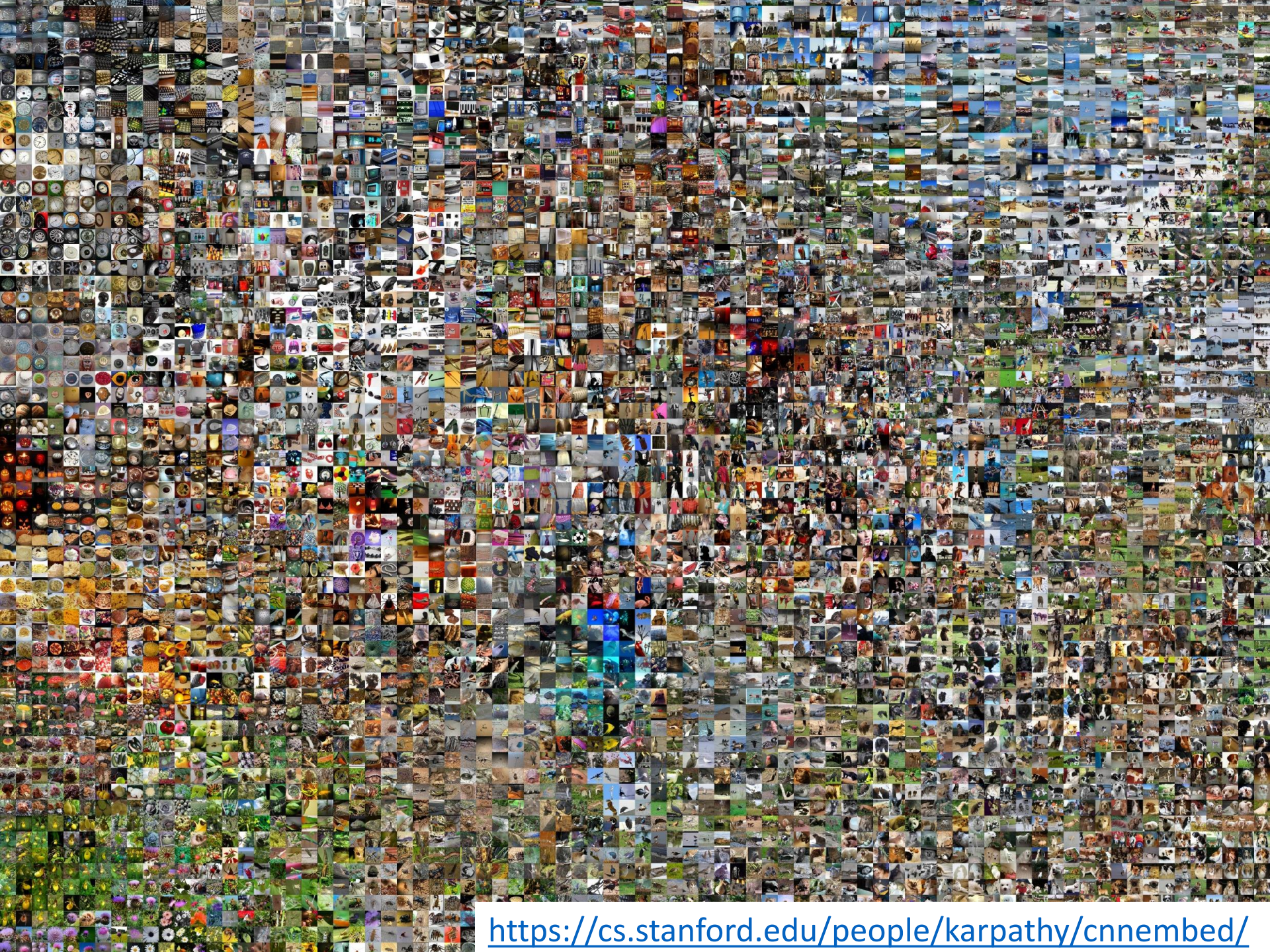
Nearest neighbours



Krizhevsky, Sutskever, & Hinton (2012)

Feature space visualisation

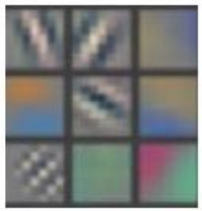
- Options for dimensionality reduction
- PCA (principal component analysis)
  - Show the dimensions with the most variance
  - Simple but often hard to interpret since only a few dimensions can be visualised simultaneously
- t-SNE (t-distributed stochastic neighbor embedding)
  - Flatten high-dimensional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



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

Visualising convolutional kernels

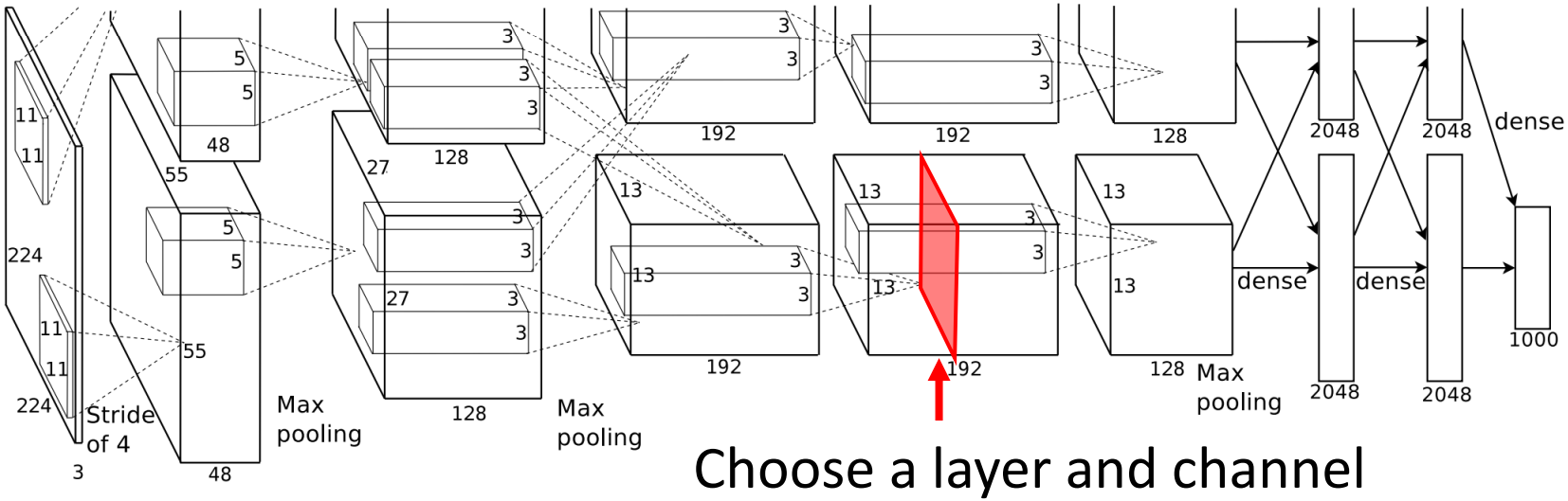
- Visualising the first convolutional layer kernels is easy because the input channels are RGB
- Visualising layer conv kernels is harder because the channels are high-dimensional and represent complex features



Layer 1

Zeiler & Fergus (2014)

Maximally activating patches

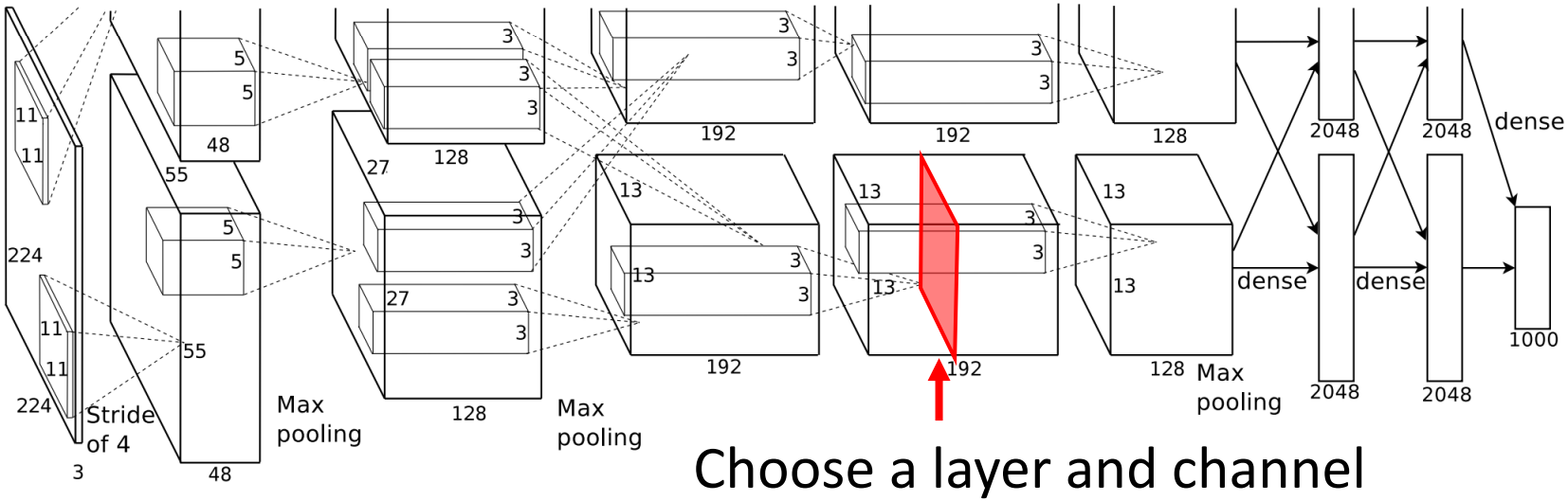


Choose a layer and channel

Run many images through the network and find patches that give the highest response in this channel:

Zeiler & Fergus (2014)

Guided backprop



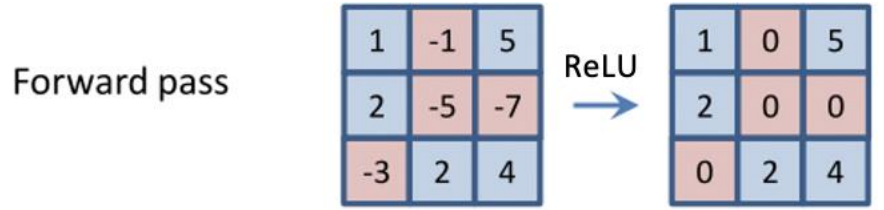
Choose a layer and channel

- Compute gradient of neuron value with respect to image pixels
- Which pixels matter most for correct classification?

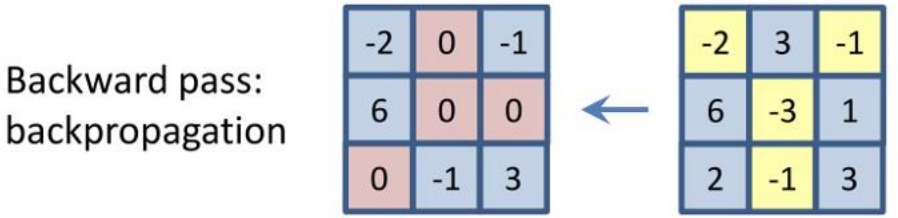
Zeiler & Fergus (2014)

Guided backprop

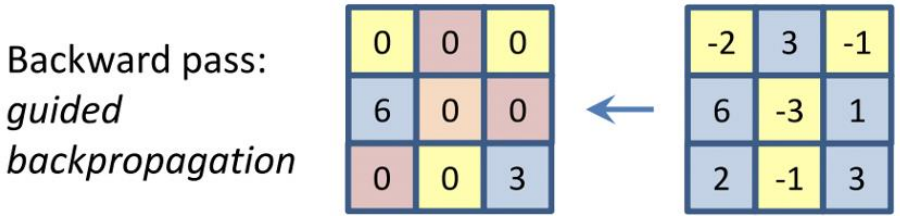
- ReLU activation function means neurons with response <0 are set to 0



- Traditional backprop does not pass back gradient when neuron response is 0

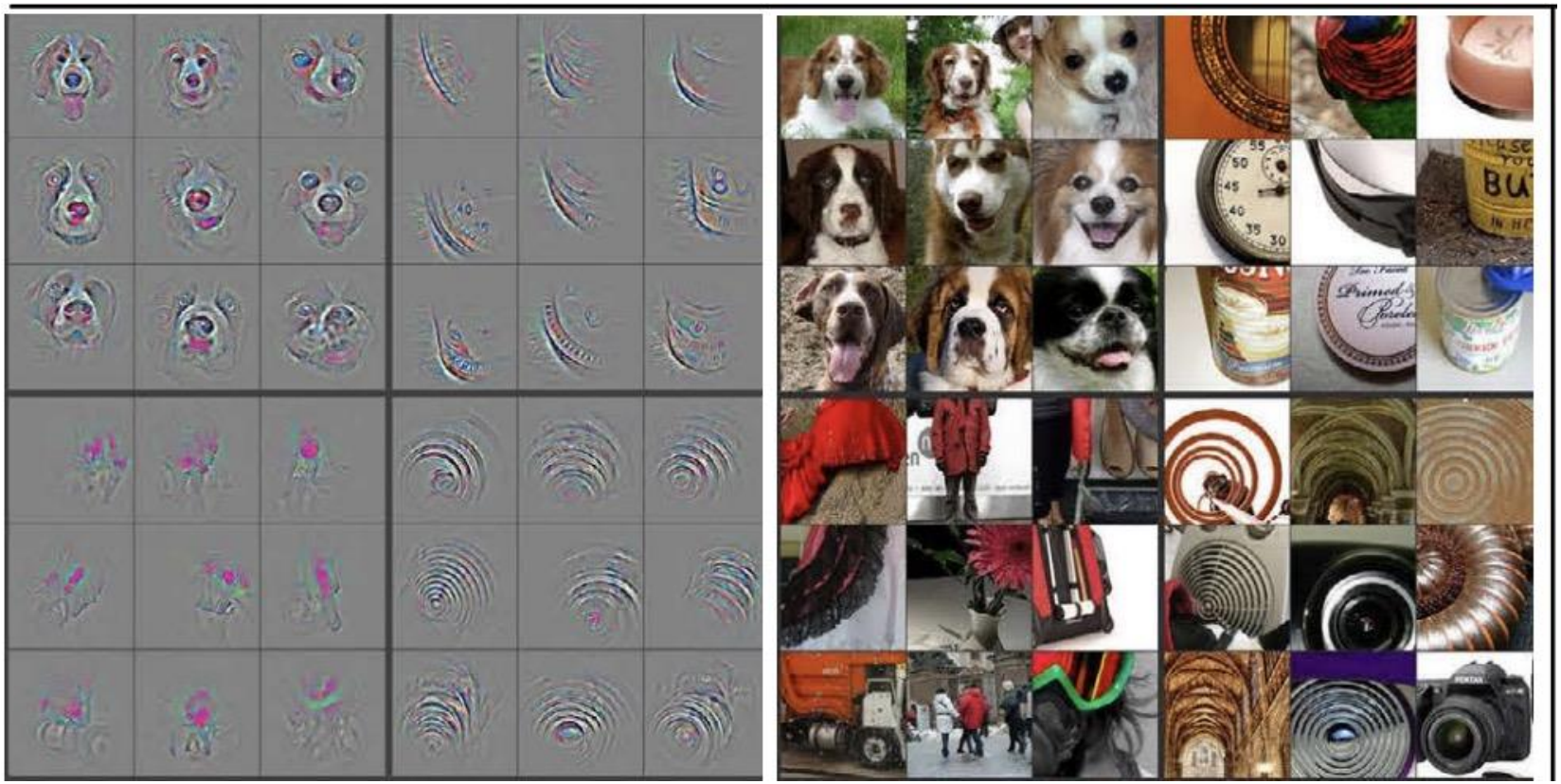


- Guided backprop also does not pass back negative gradient



Zeiler & Fergus (2014)

Visualising convolutional kernels



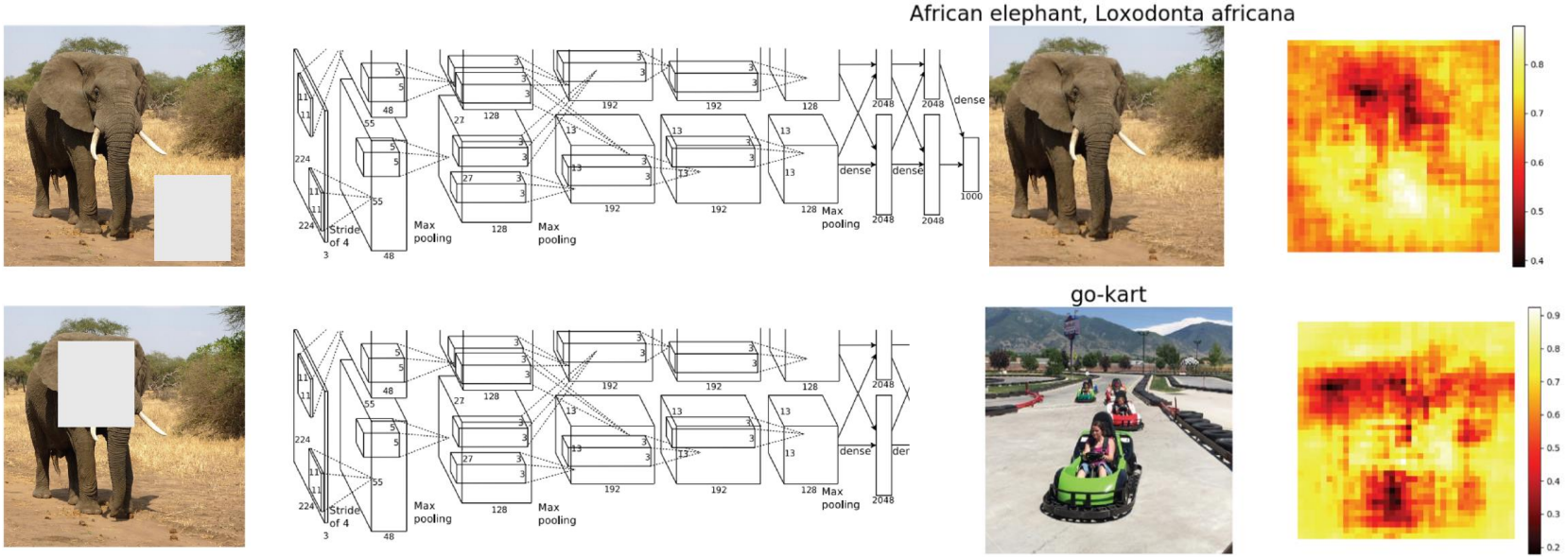
Zeiler & Fergus (2014)

Visualising image regions

- 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

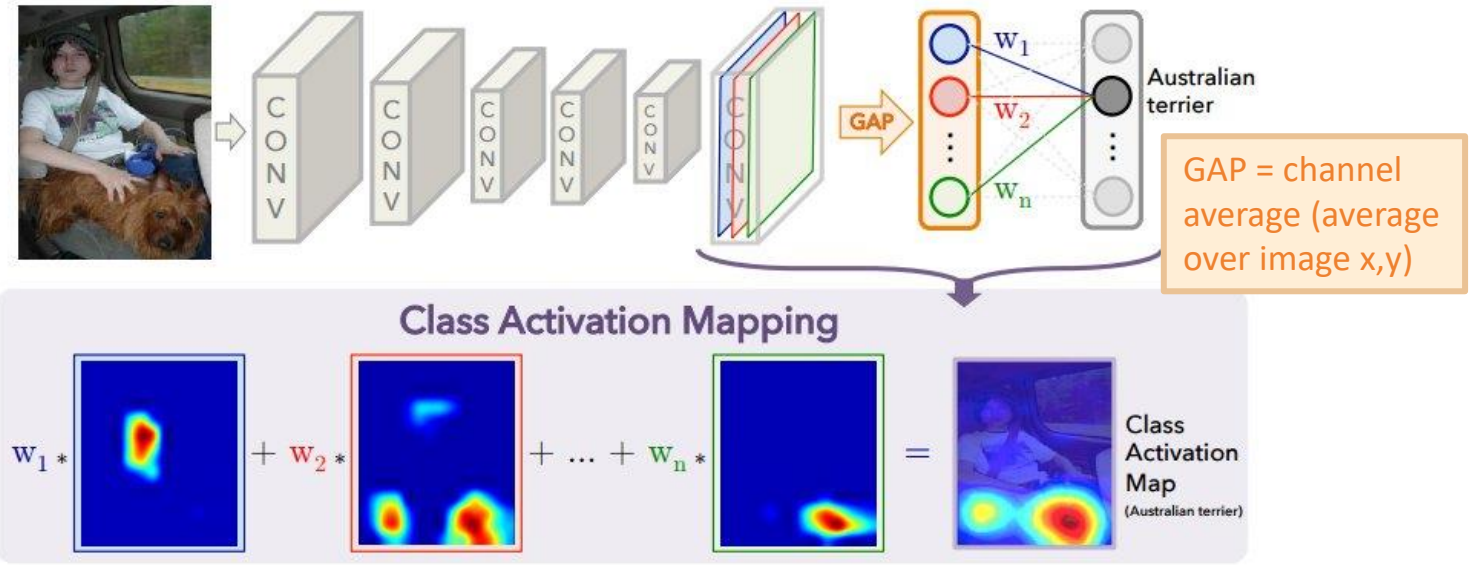
- Occlusion method: mask image and see how much class probability changes



Zeiler & Fergus (2014)

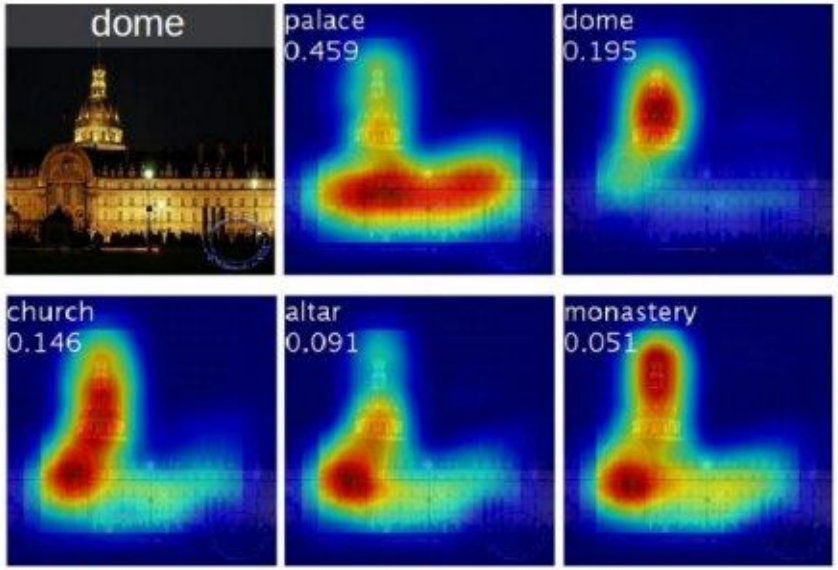
CAM (Class Activation Mapping)

- 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)



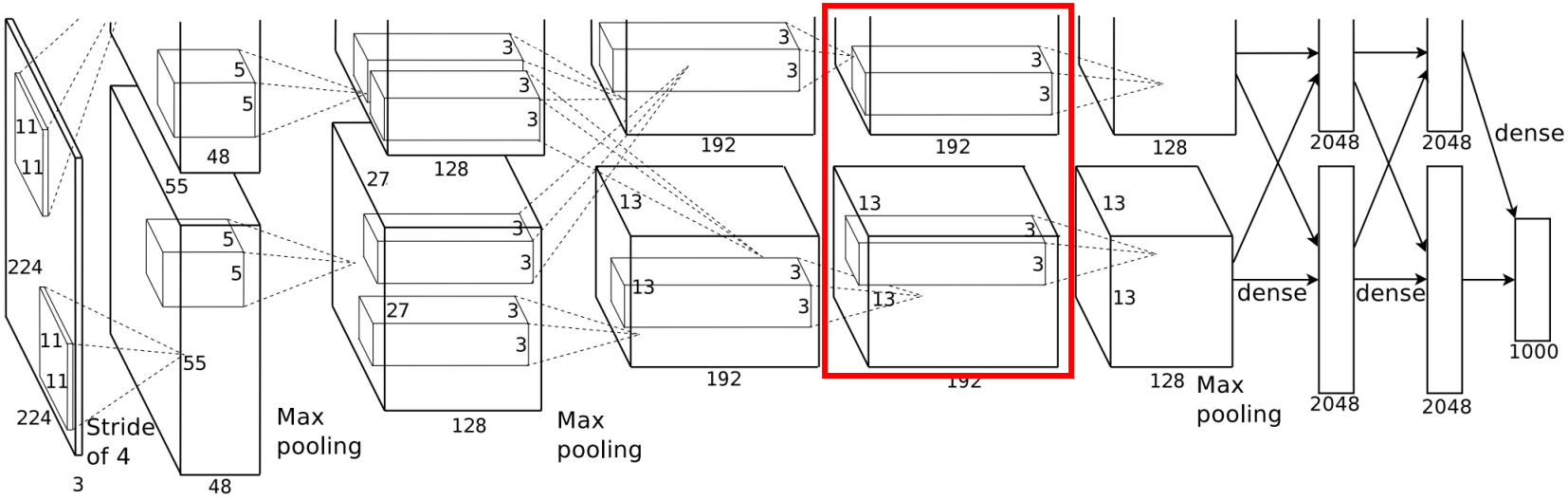
Class activation maps of top 5 predictions

Class activation maps for one object class

CAM (Class Activation Mapping)

- Disadvantages of CAM:
  - Most models don't use GAP, so the GAP layer must be added to a pretrained network and then finetuned
  - Only allows visualisation of the last layer
- More flexible alternative: Grad-CAM (Gradient-Weighted Class Activation Mapping)

Grad-CAM

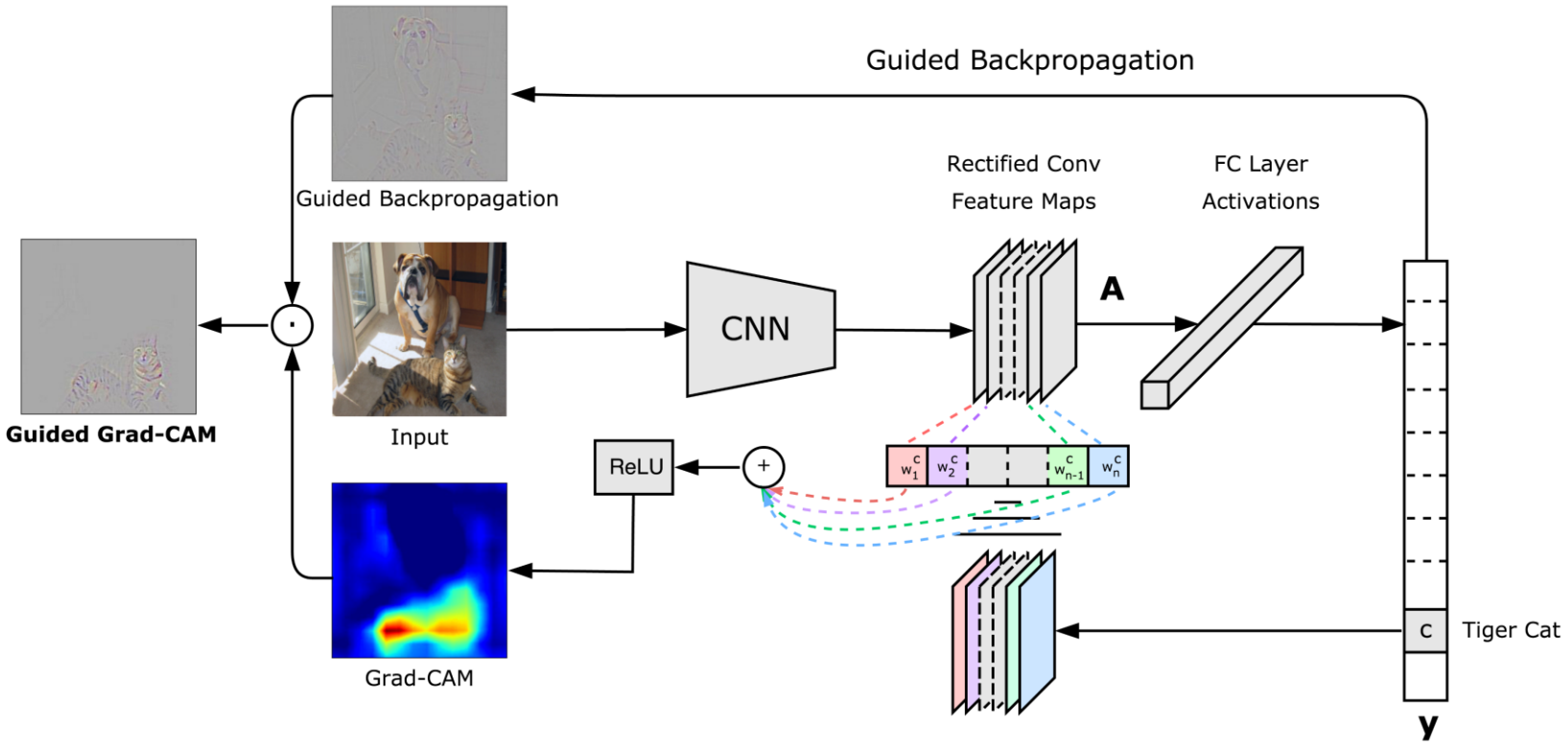


- Take response from some layer $A \in \mathbb{R}^{HxWxK}$
- Compute gradient of class score with respect to layer response
- Global Average Pool (average over image x,y) the gradients to get a vector of weights α_k (1 weight per channel)
- Compute activation map $ReLU(\sum_k \alpha_k A_{h,w,k})$

Selvaraju, et al. (2016)

Grad-CAM

Gradient-Weighted Class Activation Mapping (Grad-CAM)



Selvaraju, et al. (2016)

Grad-CAM

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

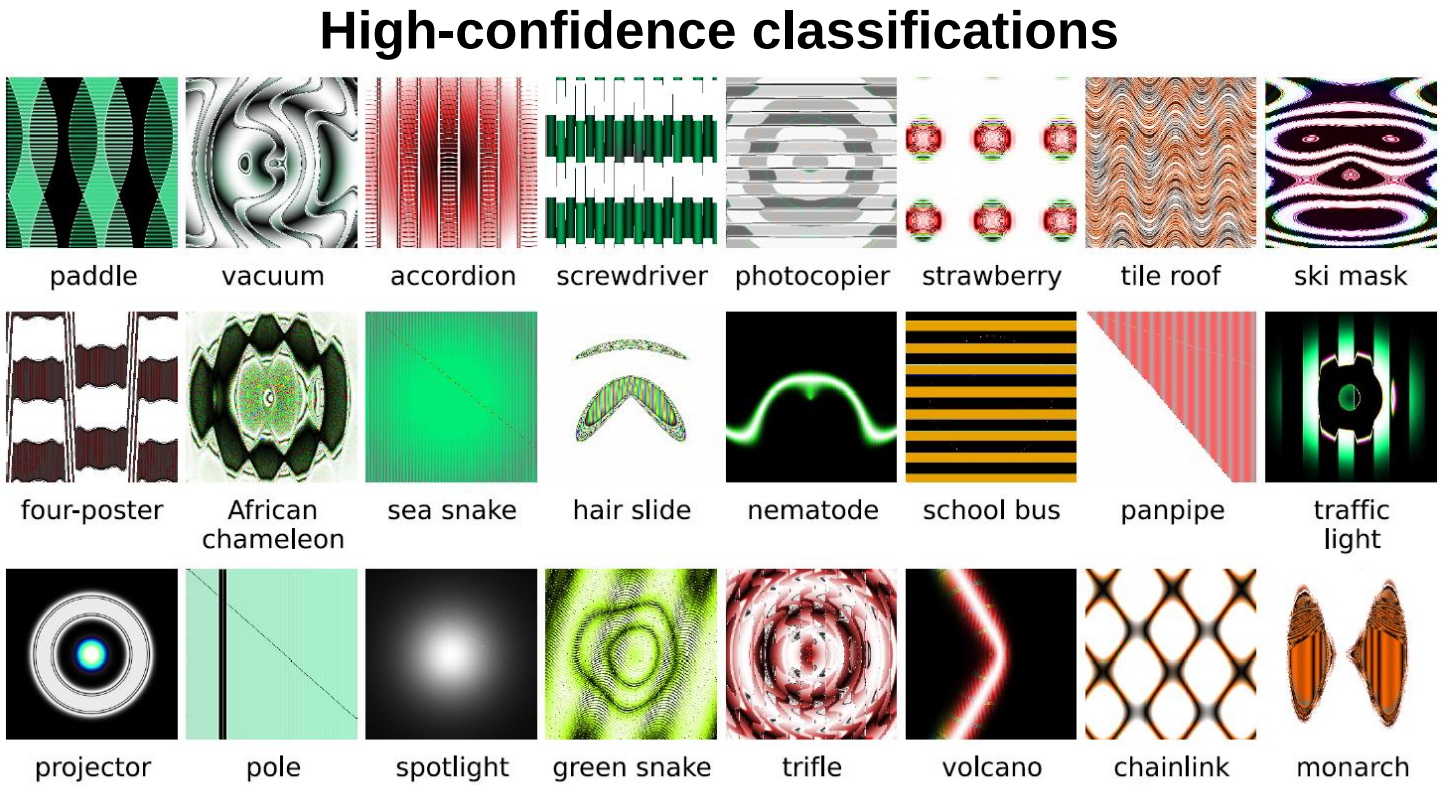
(g) Original Image (h) Guided Backprop 'Dog' (i) Grad-CAM 'Dog'

Selvaraju, et al. (2016)

Visualising classes

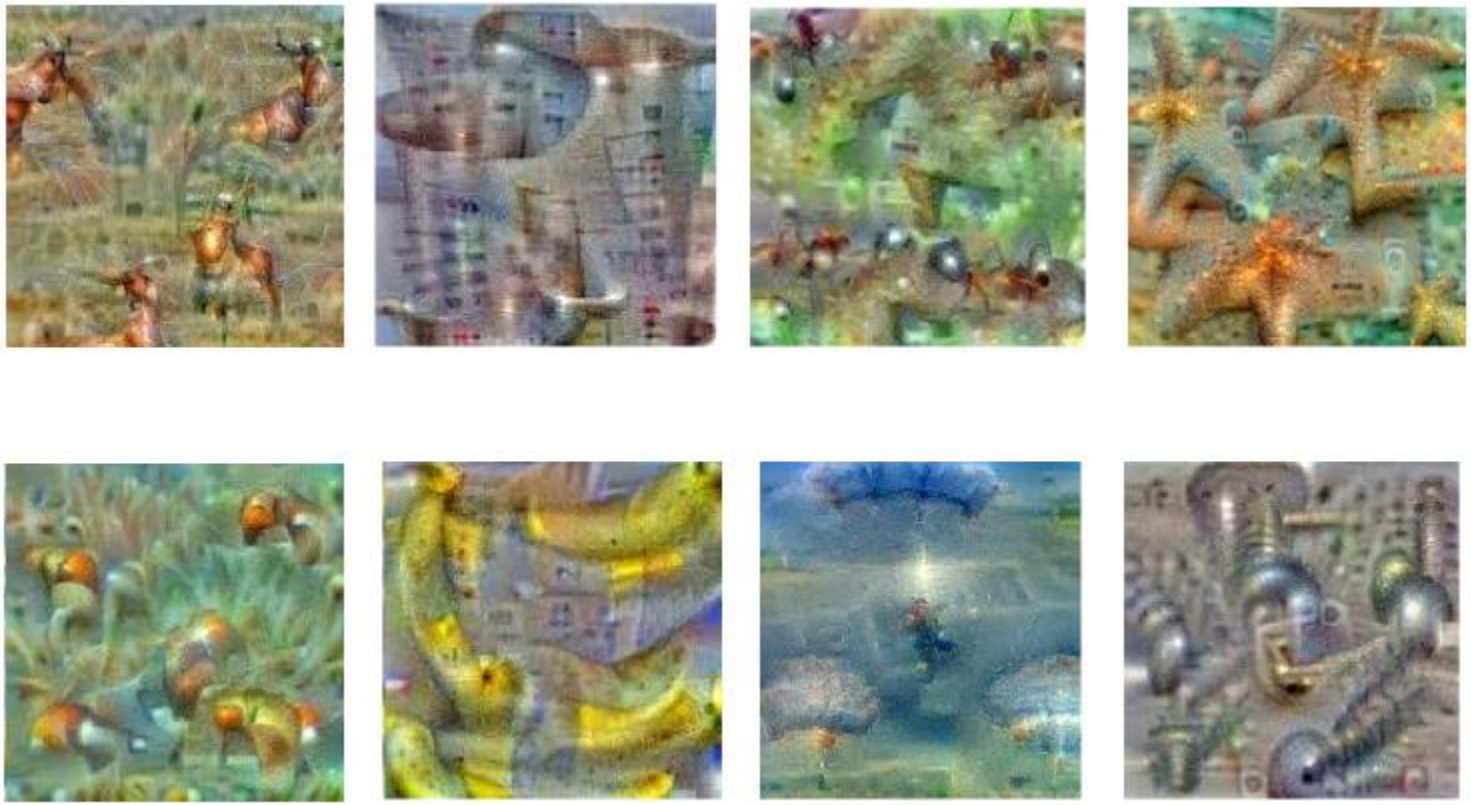
- Usually based on gradient ascent – synthesize image that maximises class label response
  - Initialise an image with zeros or small random noise
  - Run image through network, compute gradient
  - Update *image pixels* in a direction that minimises loss
- Problem: there are many possible arrays of 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
  - Feature space visualisations
  - Visualising image regions that support a class decision
  - Class visualisations
- Each method provides different information, so it's usually best to try multiple approaches

Invariance & generalisation

Invariance / tolerance

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

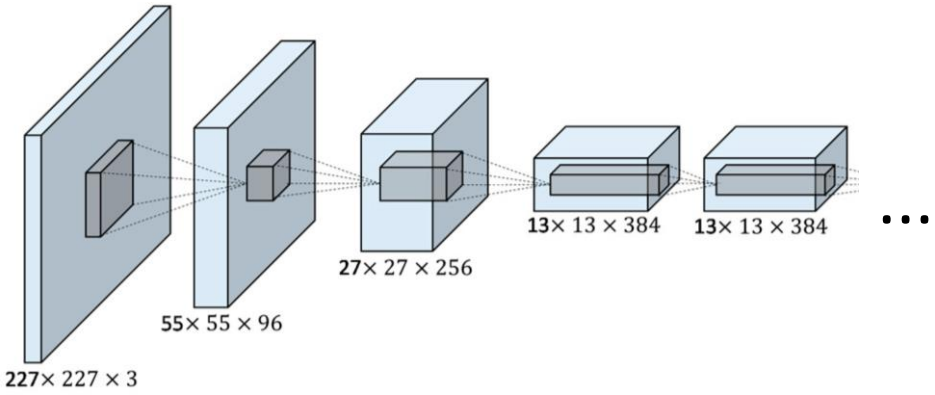
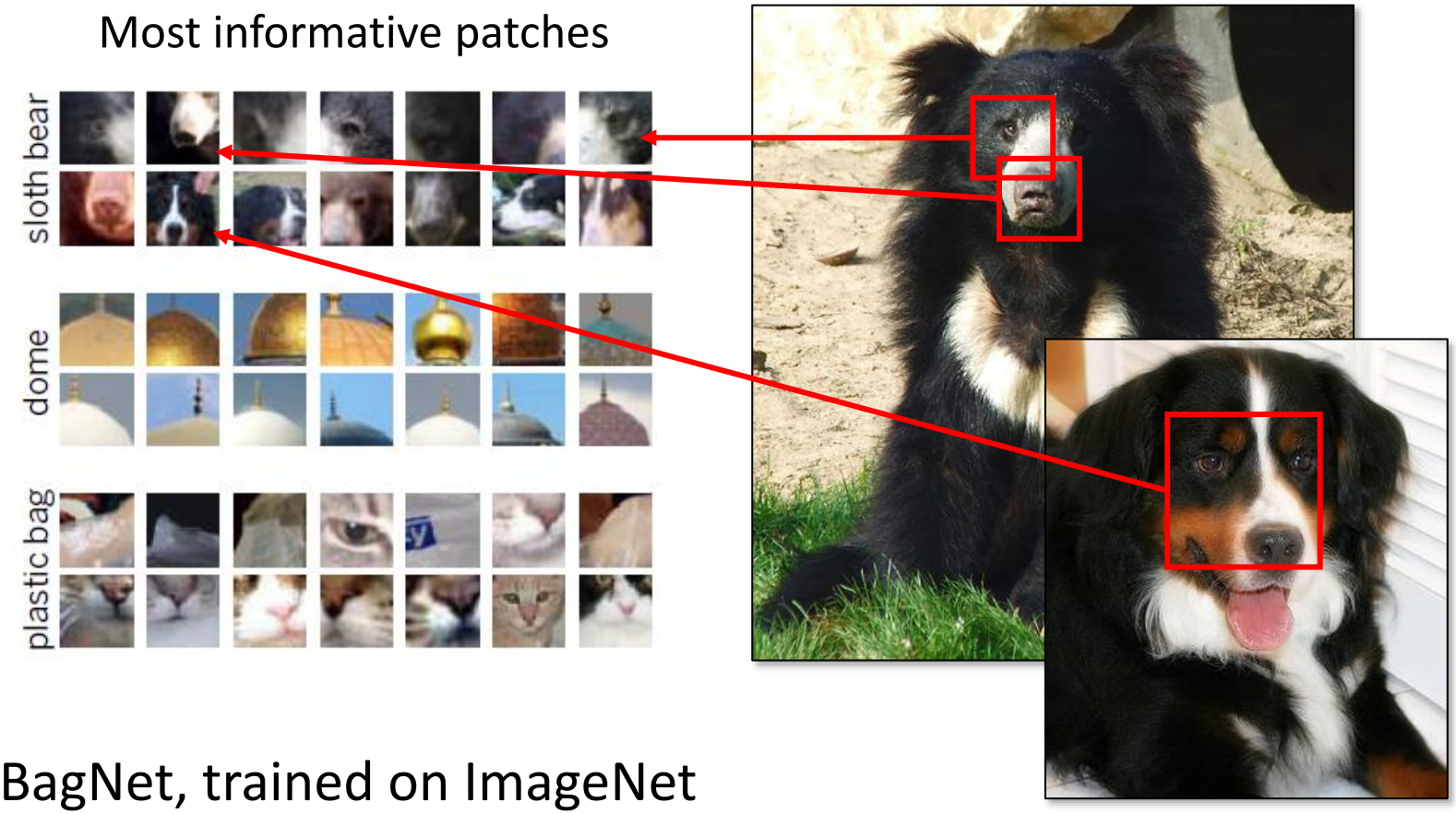


Image: Han, Zhong, Cao, & Zhang (2017)

Visualising classes



BagNet, trained on ImageNet

Brendel & Bethge (2019)

Shape and texture


original


greyscale

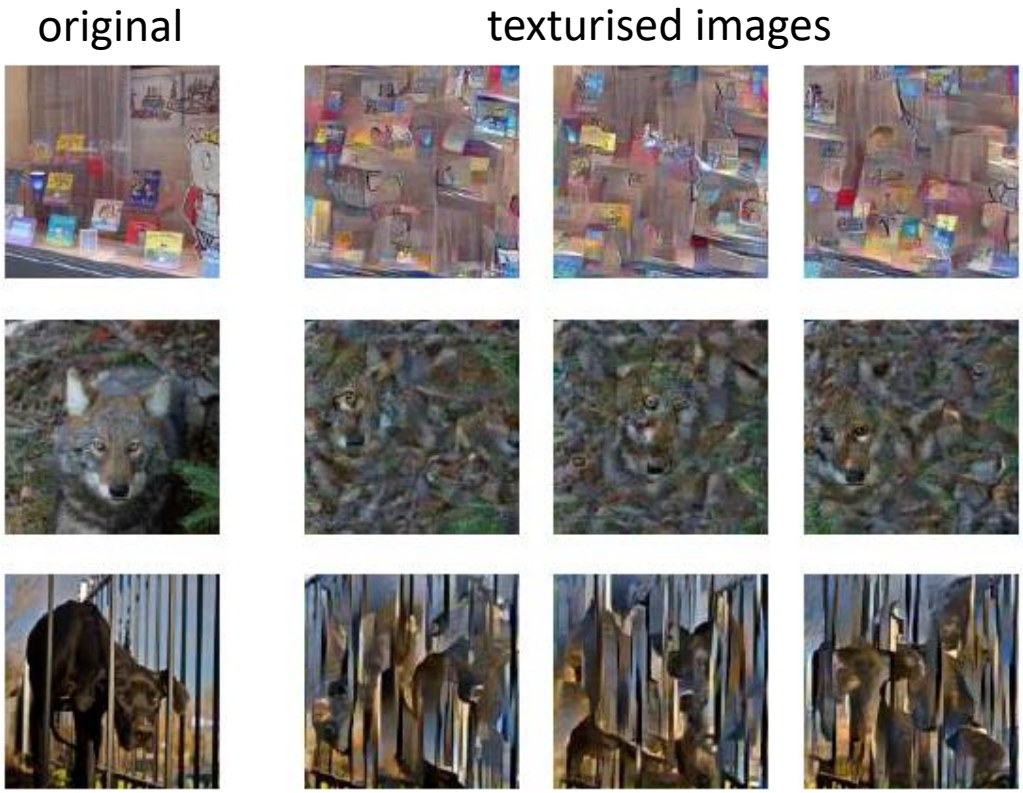

silhouette


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:

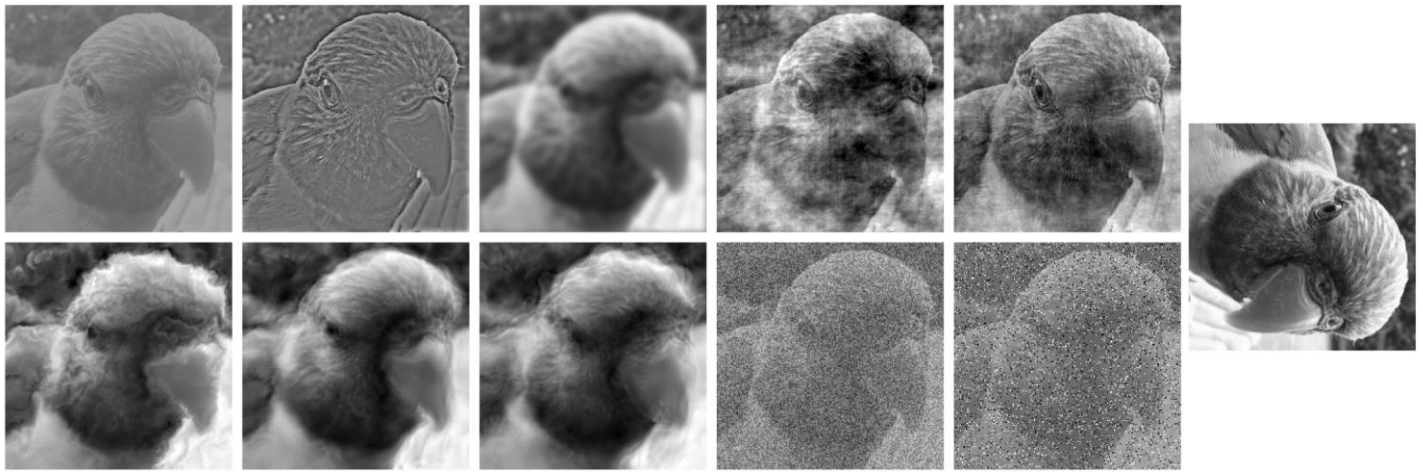
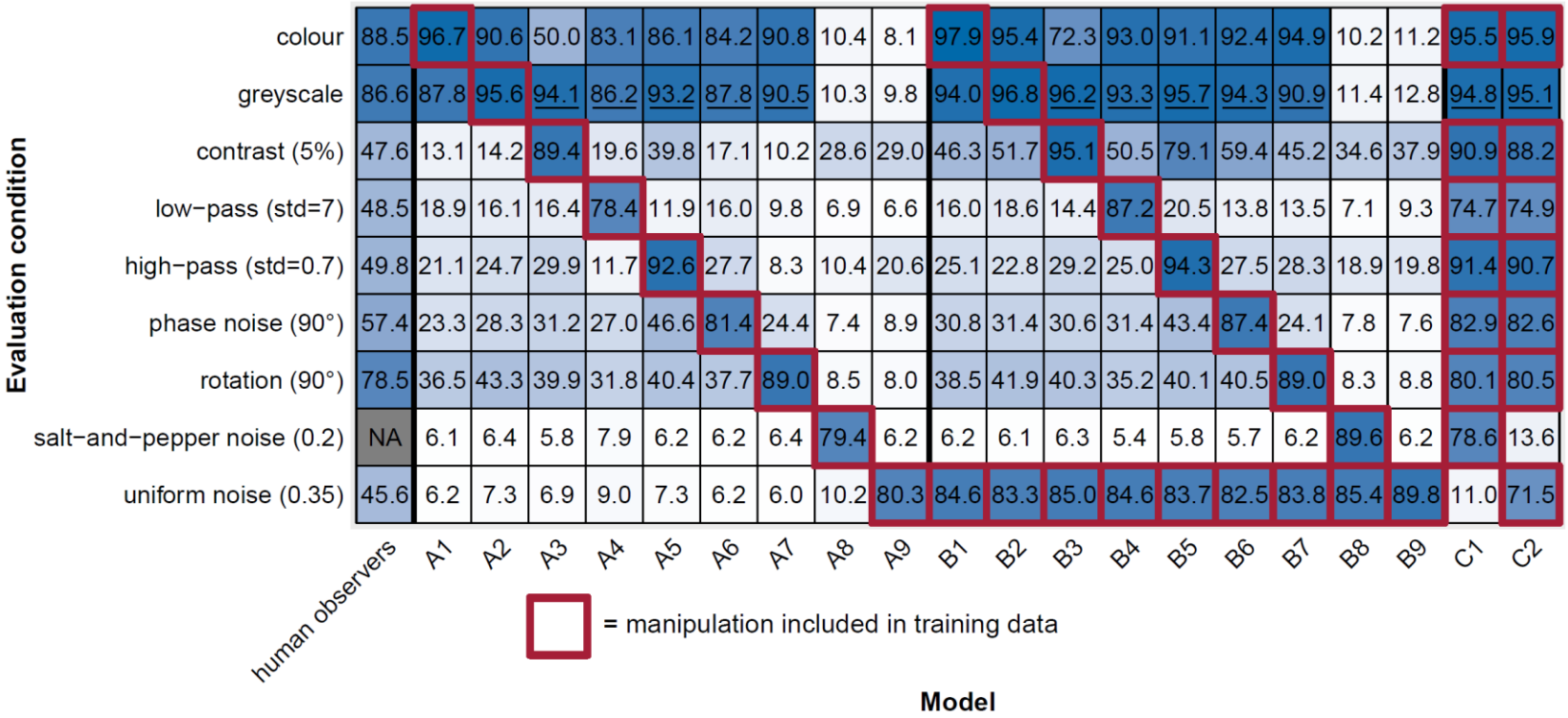


Image: Geirhos, Temme, Rauber, Schütt, Bethge, & Wichmann (2018)

Generalisation



ResNet-50 retrained on ImageNet subset with various manipulations

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

Invariance / tolerance

- CNNs are tolerant to variation included in the training data
- But often not tolerant to variation that didn't appear in the training data
- Classification tends to rely on recognizing a few key features / local texture elements

Adversarial images

- Adding small amounts of noise to an image can completely change the model's perception

Original image: "panda"
(57.7% confidence)

+

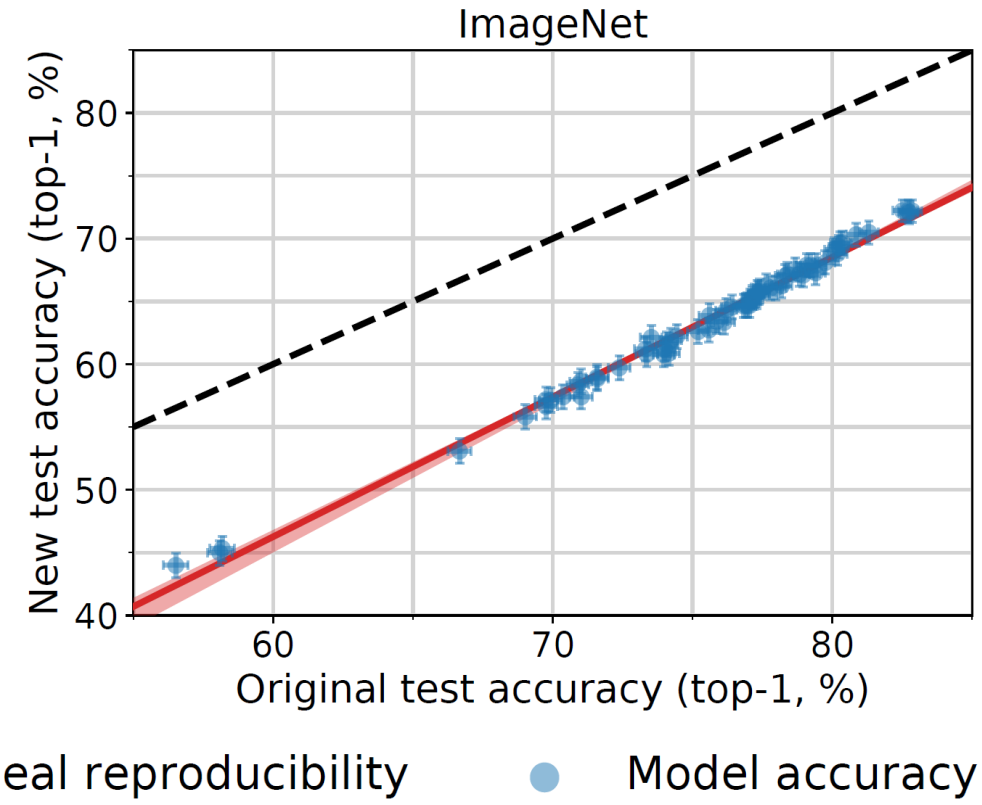
=

New image: "gibbon"
(99.3% confidence)

Image: Goodfellow, Shlens, & Szegedy (2015)

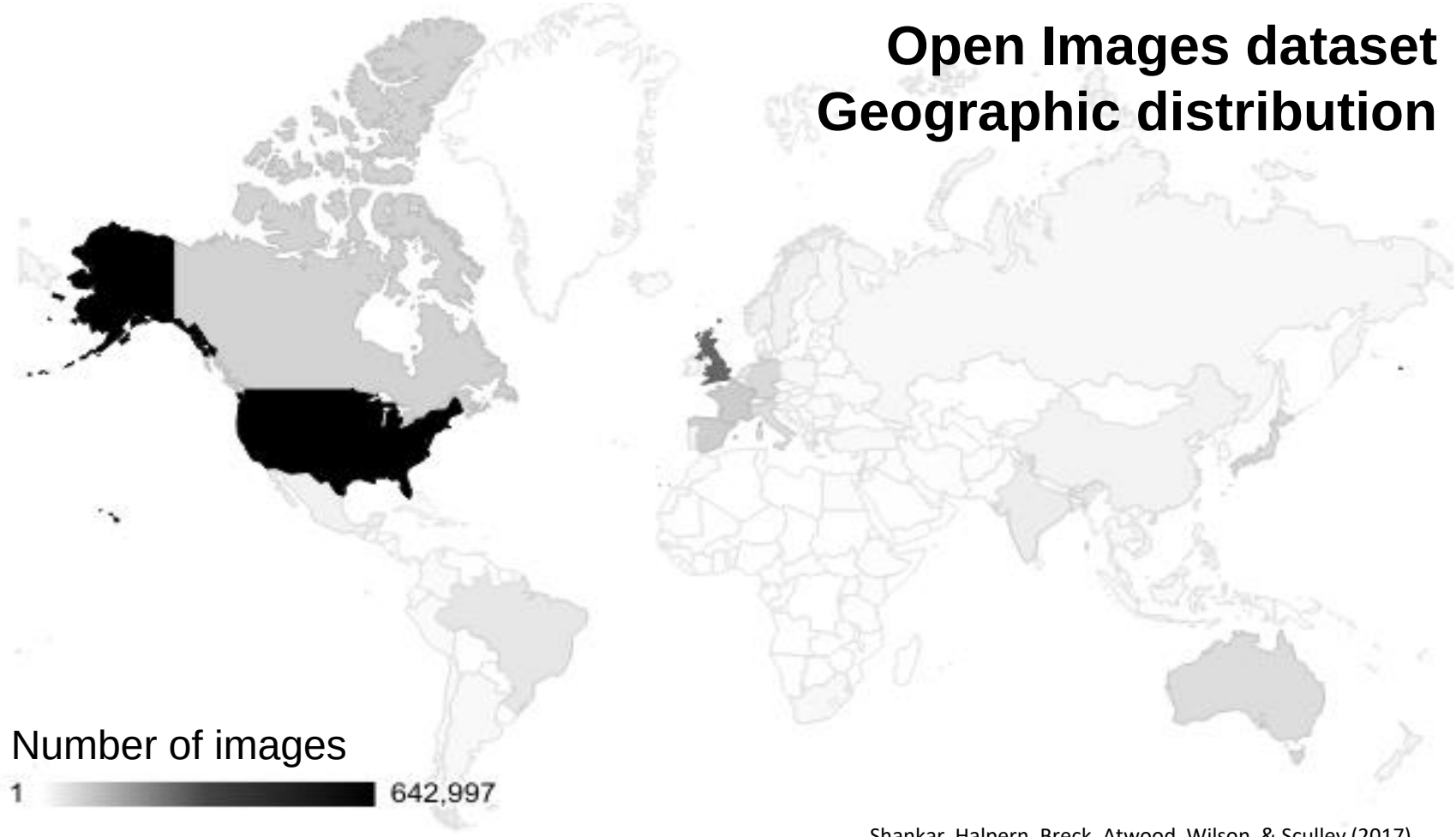
Generalisation

- Performance of top models on ImageNet vs. ImageNetV2
- Drop of about 10% suggests some overfitting to quirks of ImageNet



Recht, Roelofs, Schmidt, & Shankar (2019)

Geographic bias?



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

Summary

- ImageNet-trained features are used for a variety of visual tasks
- But ImageNet-trained CNNs have some issues:
  - Very sensitive to noise
  - Recognize local texture features but not global shape
  - Generalisation errors on very similar datasets
  - Biases due to dataset construction