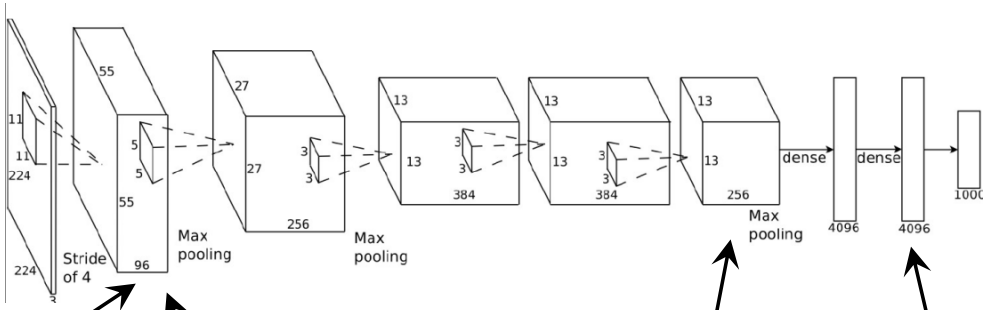
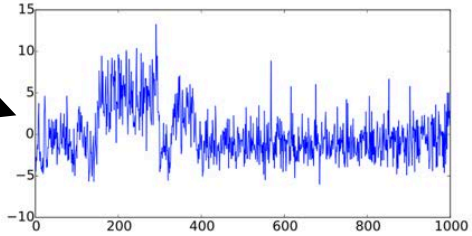
VISUALIZATION OF CNNS

Layer activations for an input

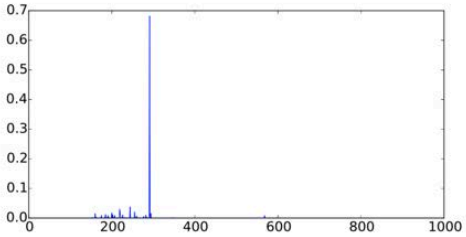
An input

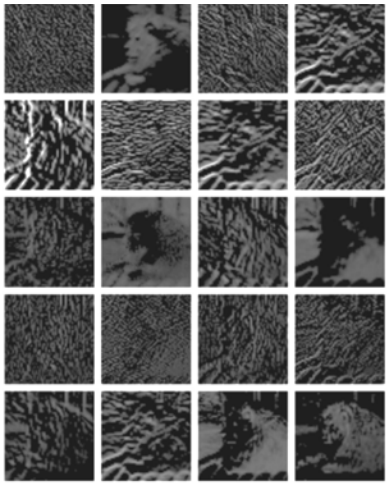


Output layer

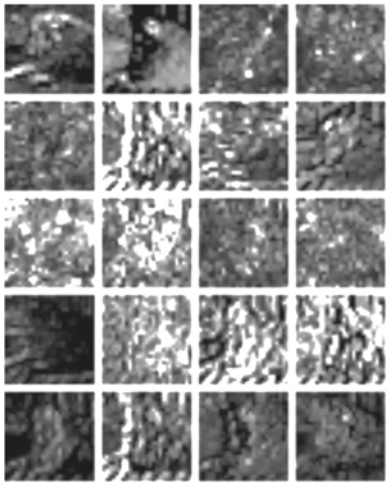


softmax

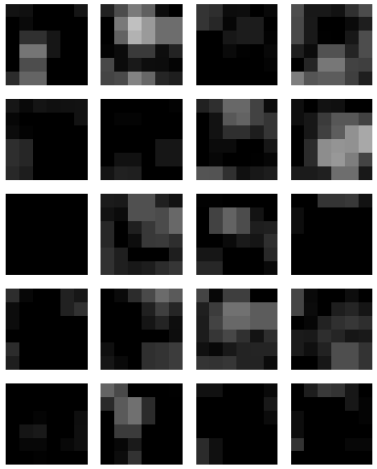




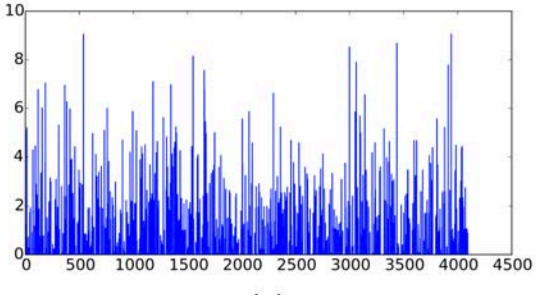
1st conv.



1st pooling

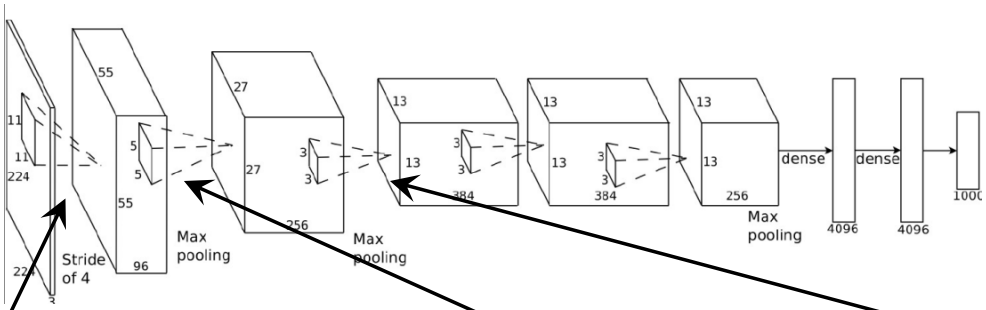


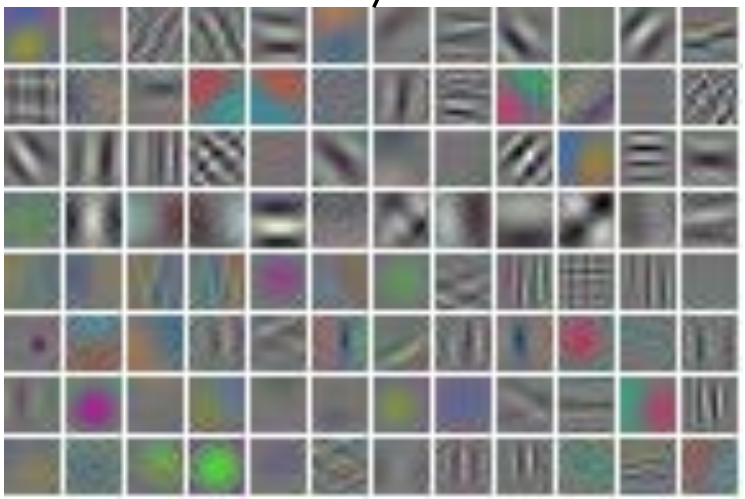
Last conv.



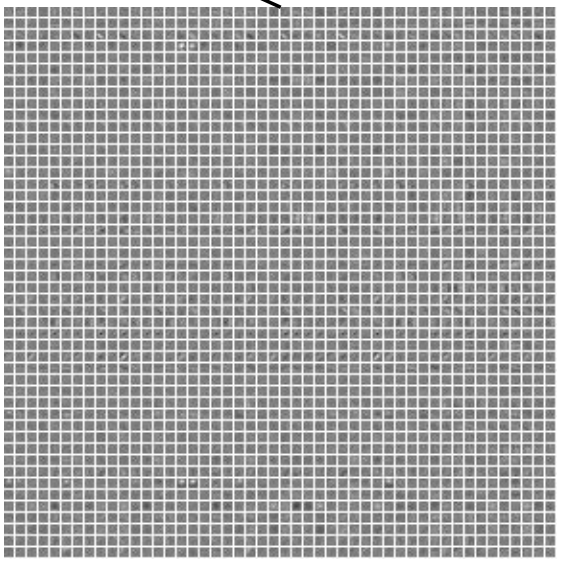
A FC layer

Learned filters

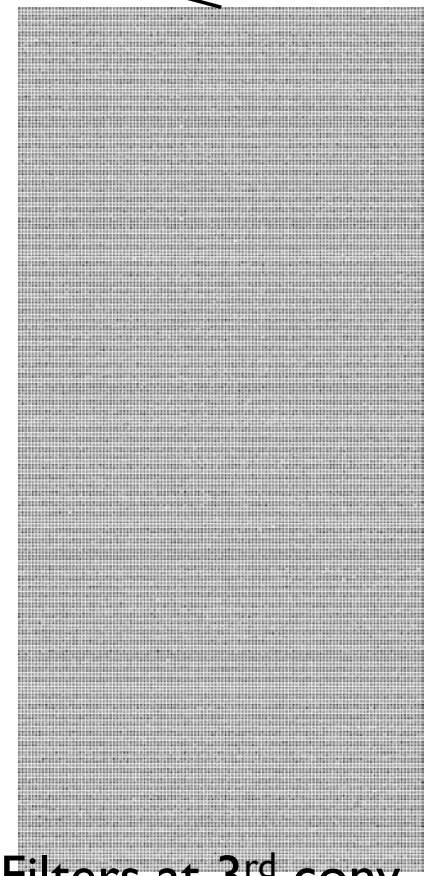




Learned filters at 1st conv.



Filters at 2nd conv.



Filters at 3rd conv.

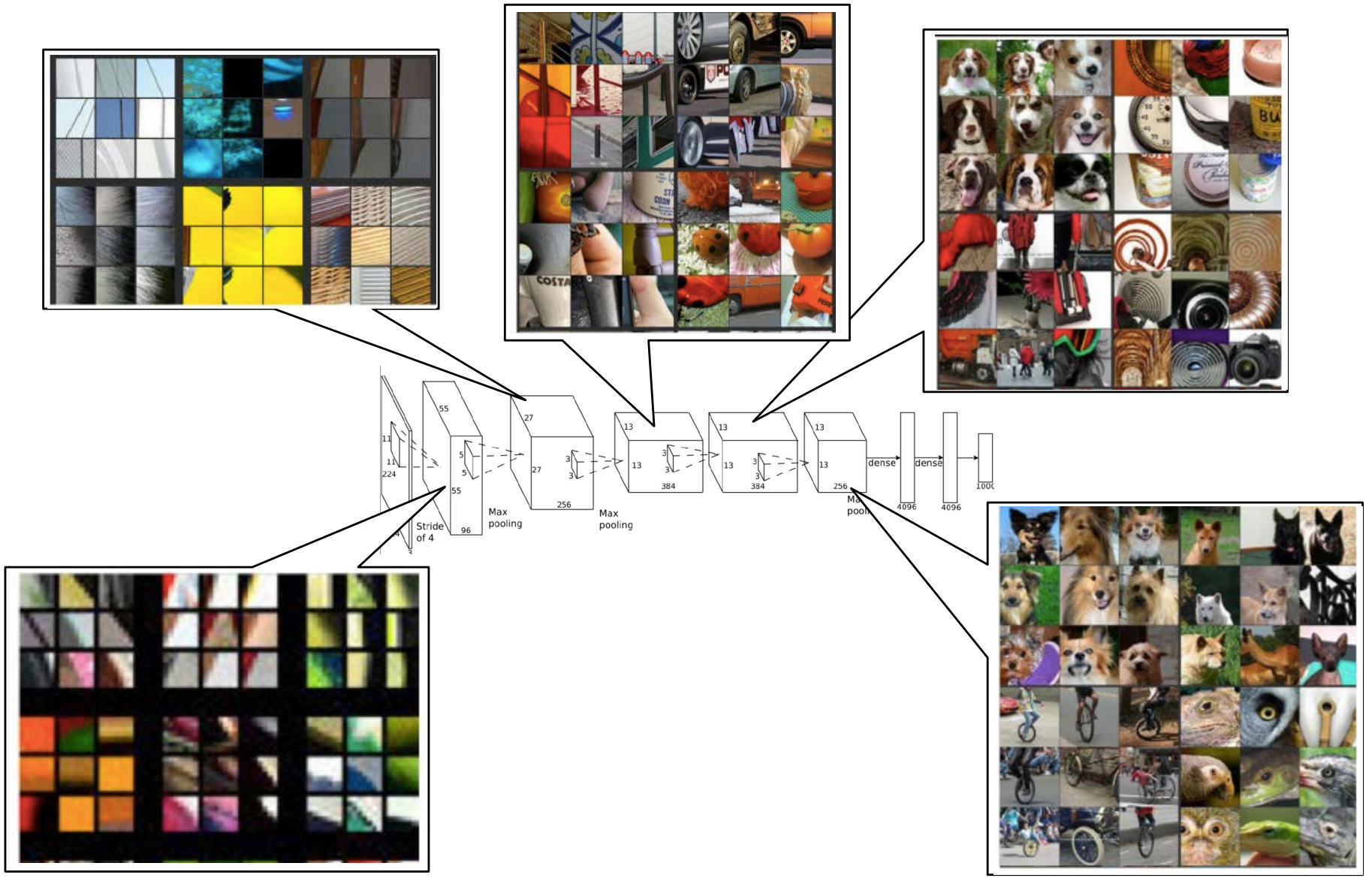
Note: Hard to interpret except for 1st conv filters

Visualization of CNNs

- Two types of visualization
  - Visualization for learned features
    - What has a CNN learned?
  - Visualization for inference
    - Where does a CNN look at for an input?

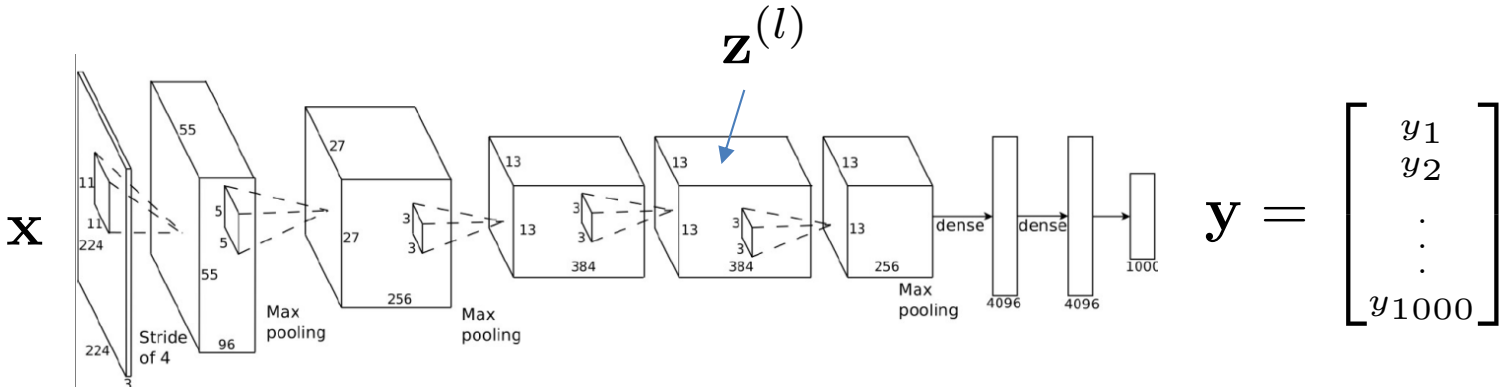
Input images maximizing unit activation

Zeiler, Fergus, Visualizing and understanding convolutional networks, ECCV2014



Recovering an input from its layer activation

- Computing an optimal input x for the score of a chosen class or activation of a layer



*k*th class score is the target:

$$\hat{\mathbf{x}} = \text{argmax}_{\mathbf{x}} y_k(\mathbf{x}) + R'(\mathbf{x})$$

Regularization term ensuring x will be natural

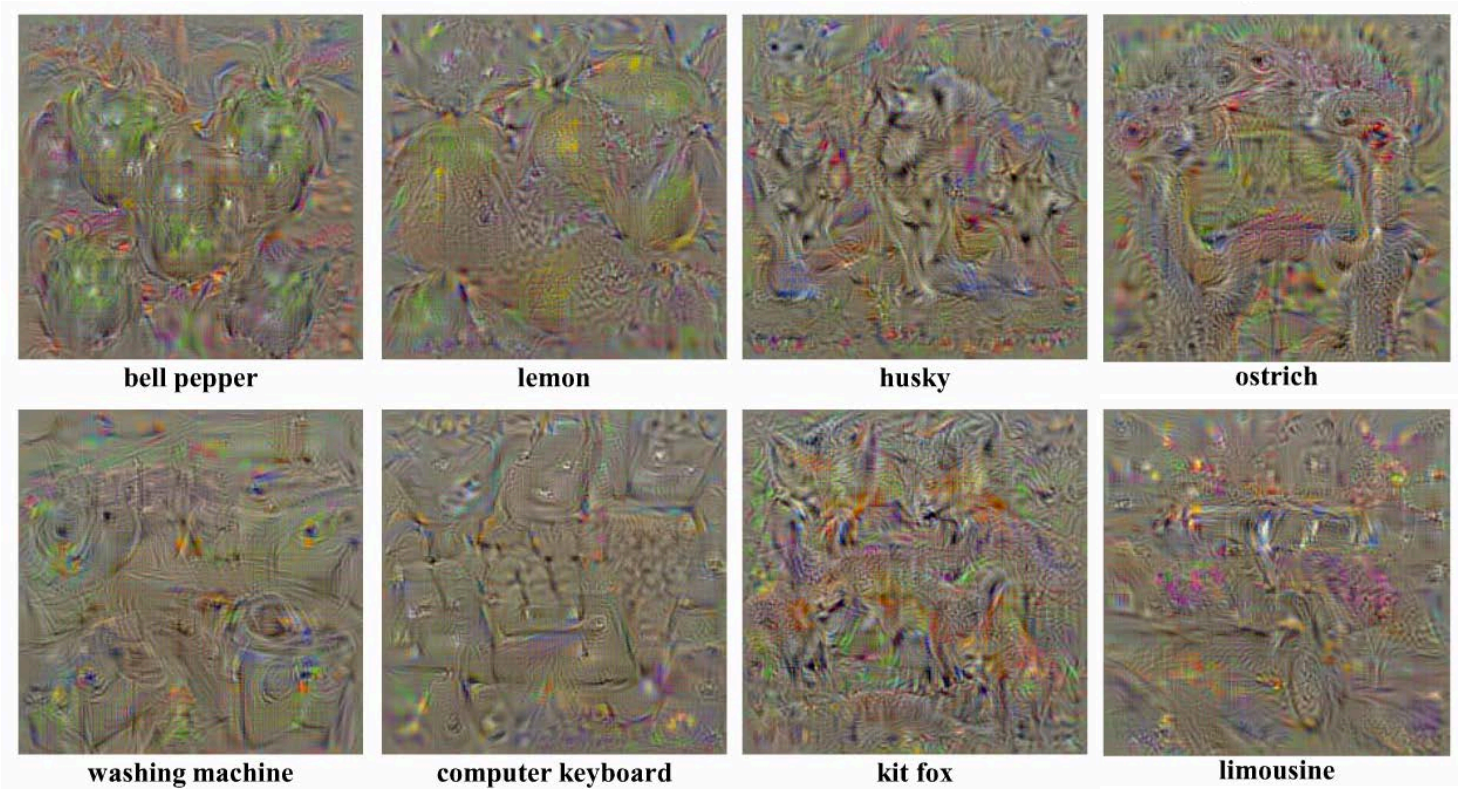
Intermediate layer activation for an input x is the target

$$\hat{\mathbf{x}} = \text{argmin}_{\mathbf{x}} (\tilde{\mathbf{z}}^{(l)} - \mathbf{z}^{(l)}(\mathbf{x}))^2 + R(\mathbf{x})$$

Optimal input maximizing a class score

Simonyan+, Deep Inside Convolutional Networks: Visualizing Image Classification Models and Saliency Maps, ICLR2014

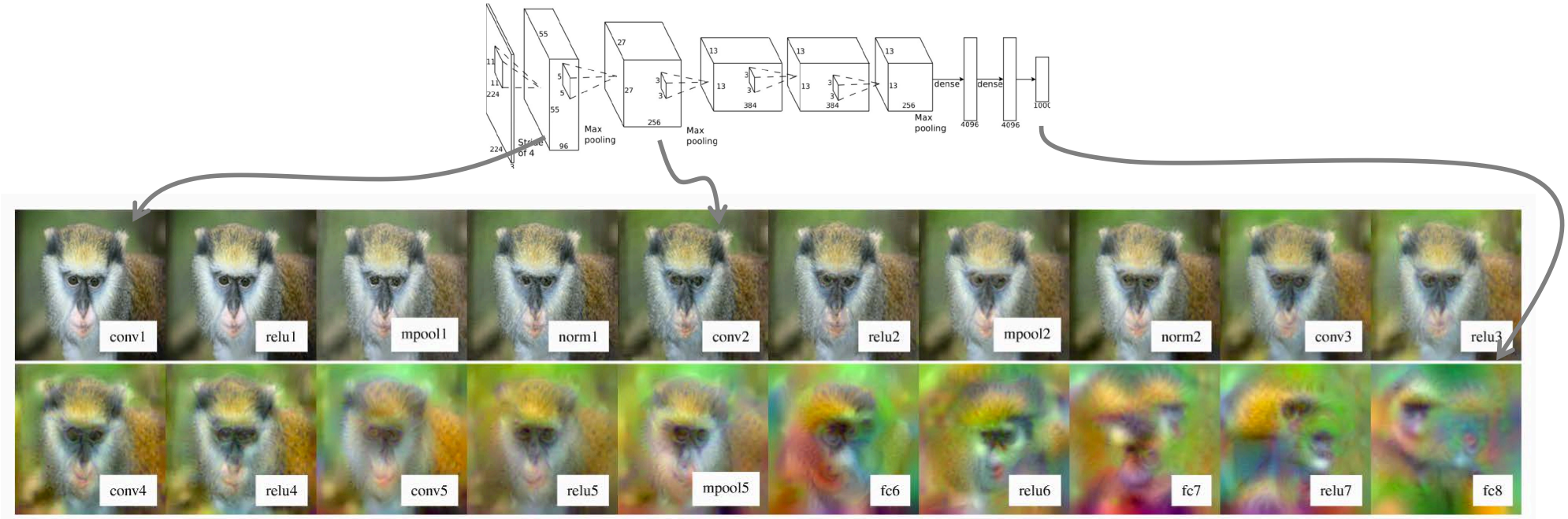
- Input is optimized to maximize the score of a chosen class



Recovering an input from its layer activation

Mehendran+, Understanding Deep Image Representations by Inverting Them, 2015

- Target: intermediate layer activation of an input x



Solution will differ depending on the initial value:

Multifaceted visualization

Nguyen+, Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks, 2016

- Dependency on the initial value → When choosing cluster centers of training data are chosen for initial values?
  - Results show a single concept corresponds to multiple modes

Multifaceted visualization

Nguyen+, Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks, 2016

(a) *Movie theater*: outside (day & night) and inside views.

(c) *Pool table*: Up close & from afar, with different backgrounds.

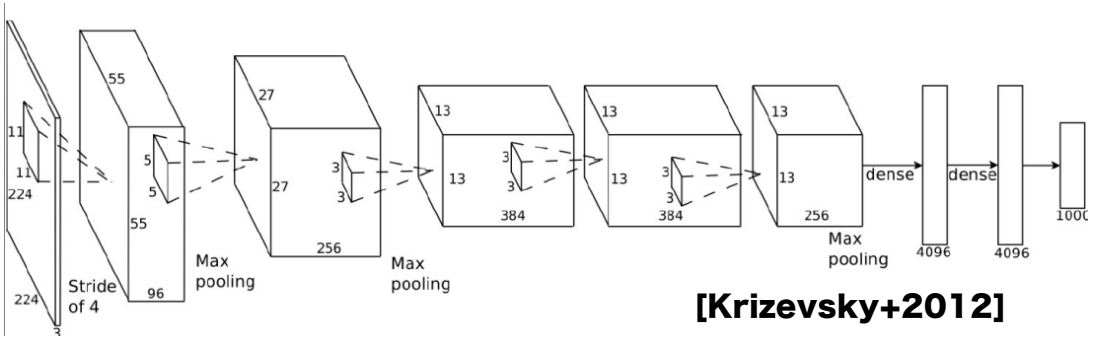
(b) *Convertible*: with different colors and both front & rear views.

(d) *Bow tie*: on a white background, on one or two people, and on cats.

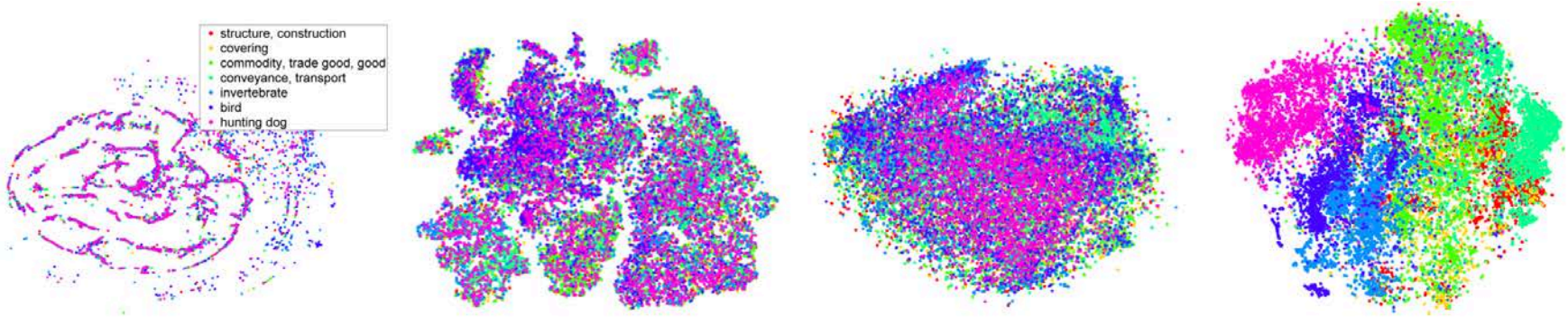
Toward understanding of internal representation

Donahue, DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition, ICML2014

- Plot of activations of each layer for a set of input images onto low-dimensional space
- Activations tend to have semantic meaning for higher layers



[Krizevsky+2012]

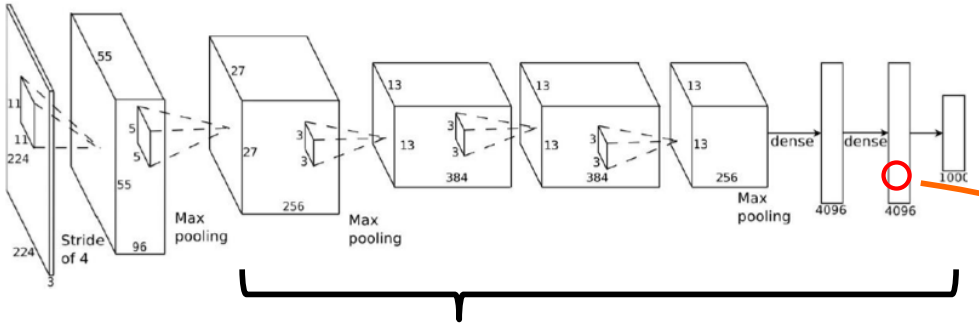


(a) LLC (b) GIST (c) DeCAF$_1$ (d) DeCAF$_6$

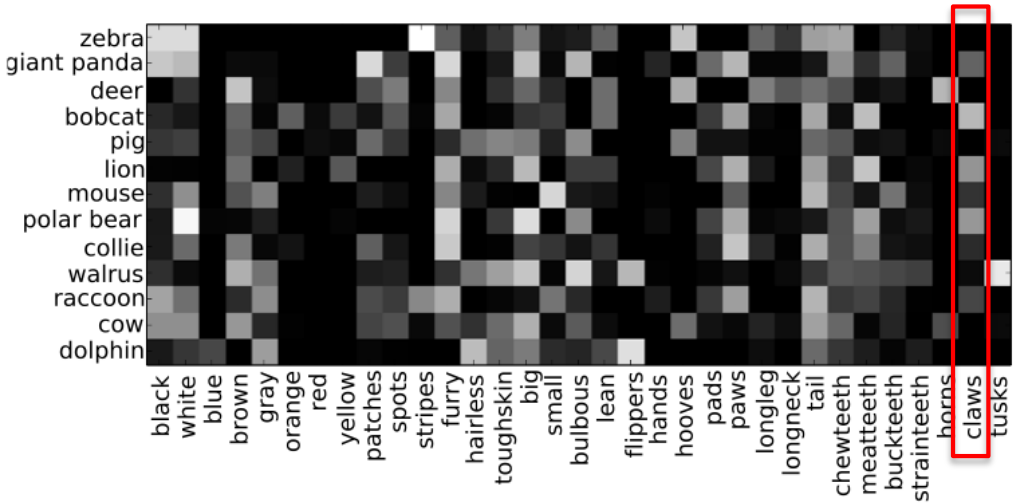
Figure 1. This figure shows several t-SNE feature visualizations on the ILSVRC-2012 validation set. (a) LLC , (b) GIST, and features derived from our CNN: (c) DeCAF$_1$, the first pooling layer, and (d) DeCAF$_6$, the second to last hidden layer (best viewed in color).

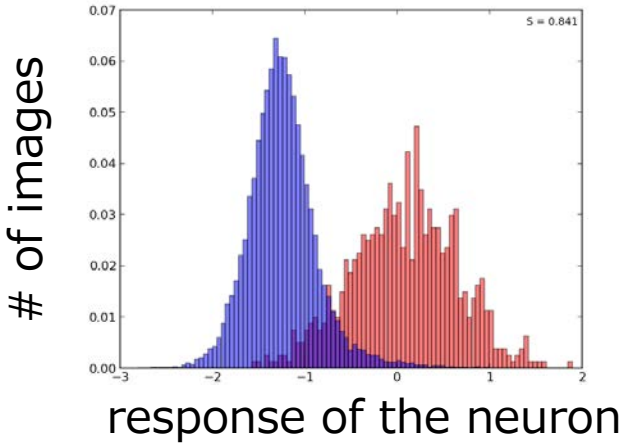
Internal representation ≒ category attributes

Ozeki, Okatani, Understanding Convolutional Networks in Terms of Category-level Attributes, ACCV2014



polar bear

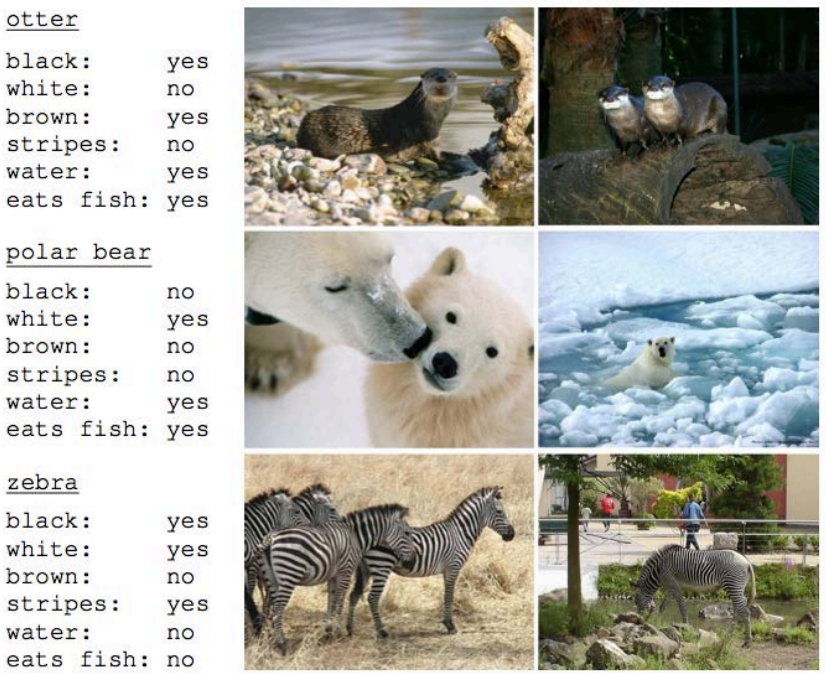


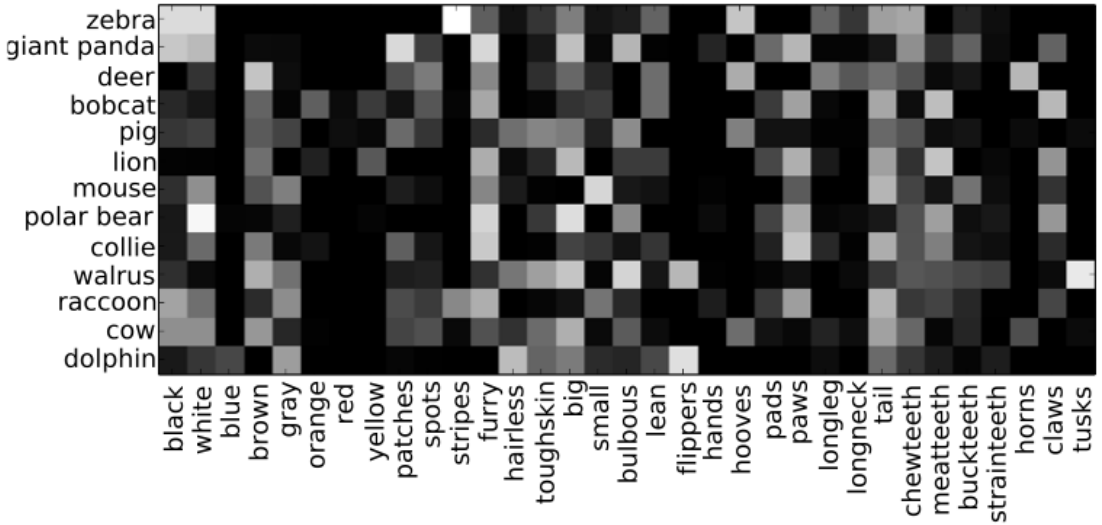


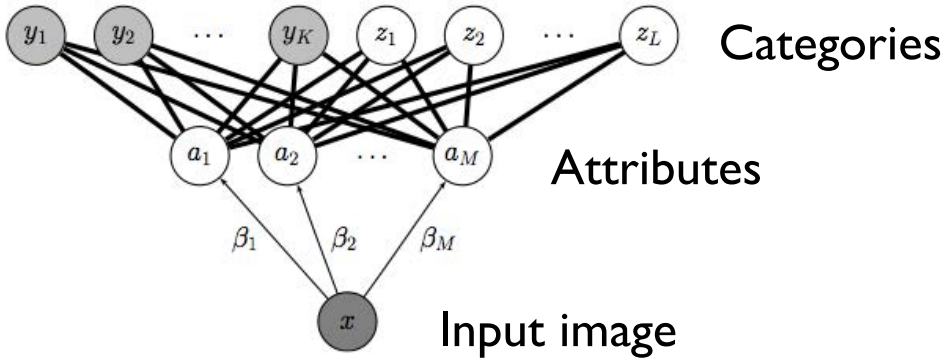
Internal representation ≒ category attributes

Ozeki, Okatani, Understanding Convolutional Networks in Terms of Category-level Attributes, ACCV2014

- Dataset：AwA（Animals with Attributes） [Lampert+2009]
- 50 animal categories and 85 attributes

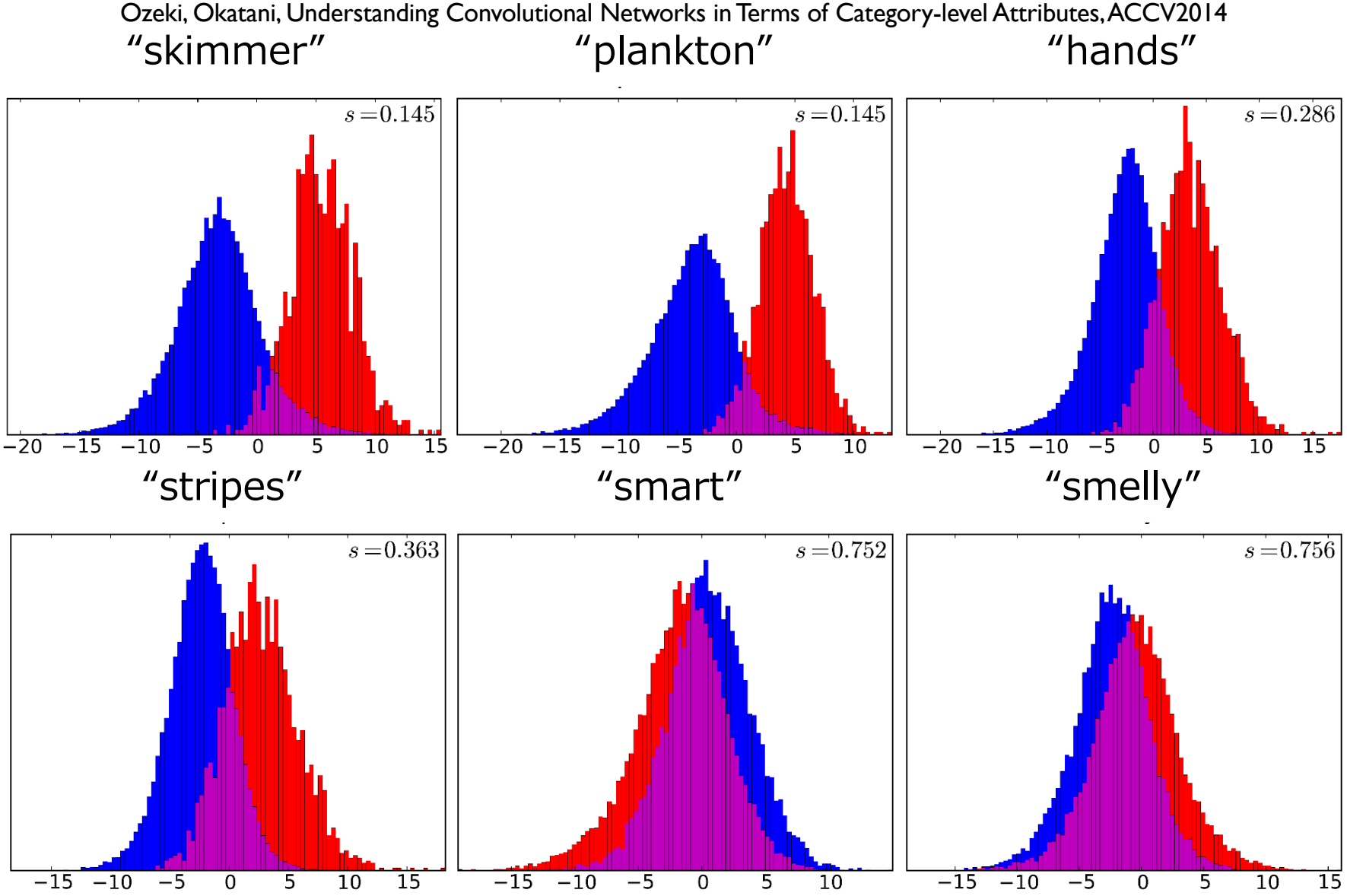






Internal representation ≒ category attributes

Ozeki, Okatani, Understanding Convolutional Networks in Terms of Category-level Attributes, ACCV2014



s = prediction

Internal representation ≒ category attributes

Ozeki, Okatani, Understanding Convolutional Networks in Terms of Category-level Attributes, ACCV2014

"stripes"

bottom | top

"spots"

bottom | top

"hairless"

bottom | top

Internal representation ≒ category attributes

Ozeki, Okatani, Understanding Convolutional Networks in Terms of Category-level Attributes, ACCV2014

"agility"

bottom | top

"smart"

bottom | top

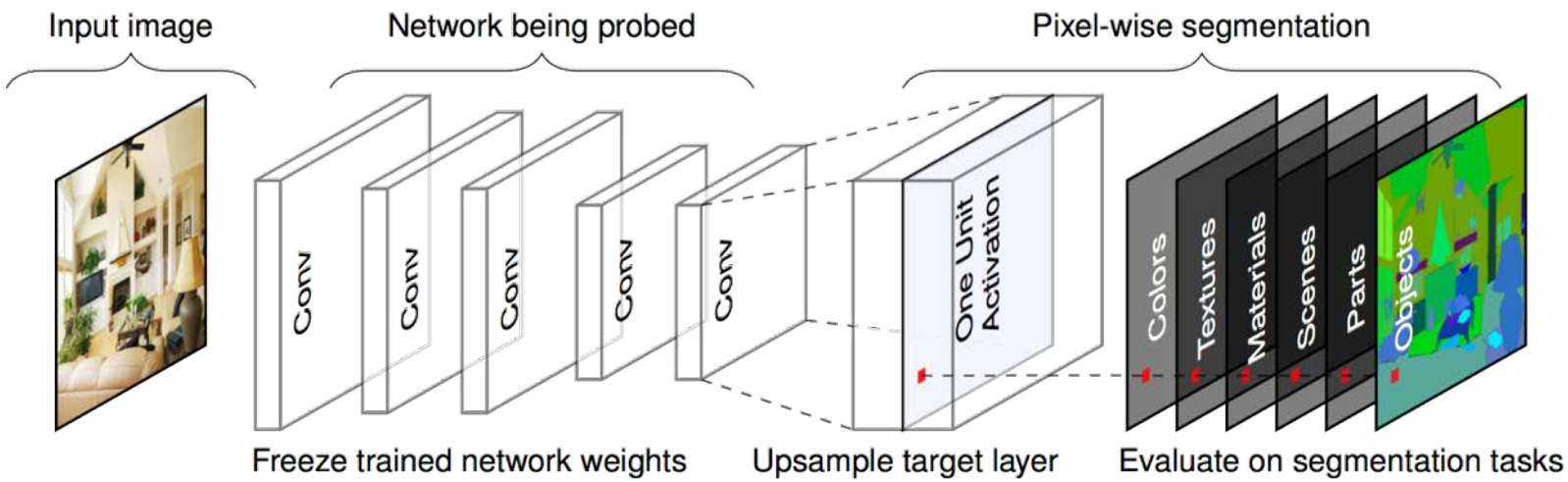
"smelly"

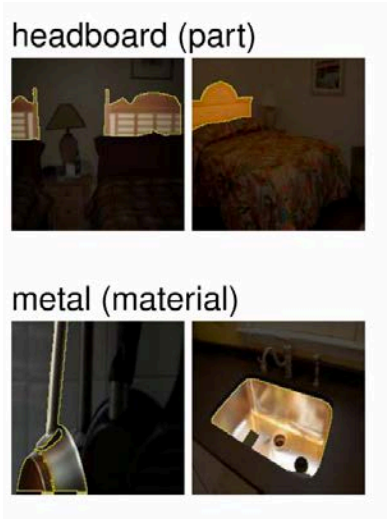
bottom | top

Network dissection

Bau+, Network Dissection: Quantifying Interpretability of Deep Visual Representations, CVPR2017

- Intuition: Find a channel for each of many visual
  - Concepts: Colors, Textures, Materials, Scenes, Parts, Objects
  - A channel of a layer is regarded as a 'concept detector'
- Method
  - Segmentation maps are created for all the concepts for a set of images
  - Find a channel whose activation map is similar to a map of a concept

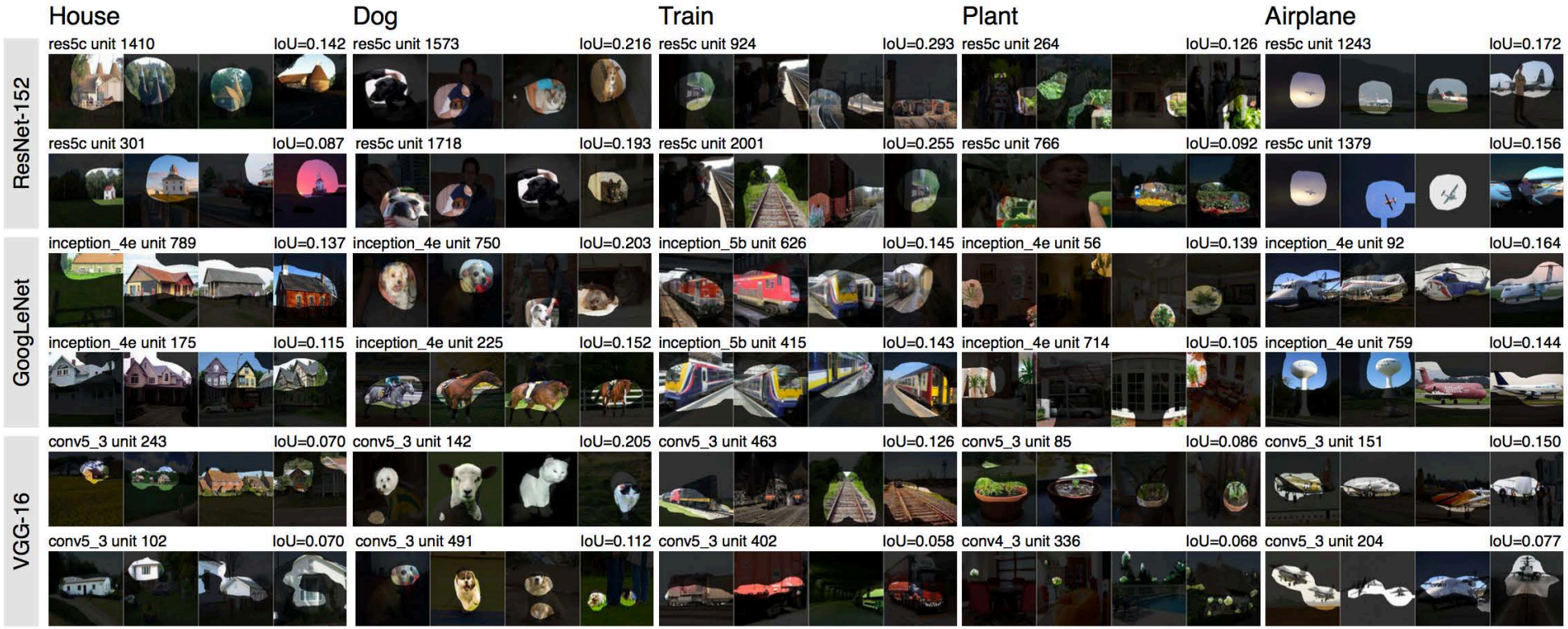




Network dissection

Bau+, Network Dissection: Quantifying Interpretability of Deep Visual Representations, CVPR2017

- Interpretability of channels of the last conv. layer



Visualization for an inference

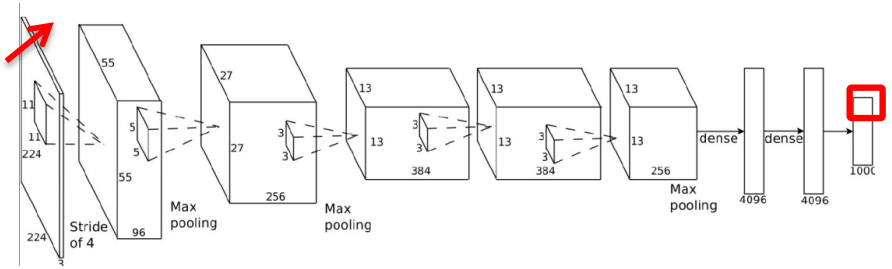
- Goal: to understand prediction of a CNN given an input
  - Gradient-based
    - Sensitivity/Saliency Map [Simonyan+14]
    - SmoothGrad [Smilkov+17]
    - DeConvNet [Zeiler-Fergus+14]
    - Guided Backprop [Springenberg+14]/DeSaliNet[Mahendran-Vedaldi16]
    - Integrated Gradient[Sundararajan+17]
  - Backprop of attribution
    - LRP[Bach+15]/Deep Taylor Decomposition[Montavon+17]/DeepLift[Shrikumar+17]/Shapely value[Lundberg-Lee16]
    - Excitation Backprop[Zhang+16]
    - PatternNet[Kindermans+17]
  - Activations (of global average pooling layers)
    - CAM[Zou+16]/Grad-CAM[Selvaraju+17]
  - Input perturbation/masking
    - LIME[Ribeiro+16]
    - Feedback CNN[Cao+15]
    - Prediction Difference Analysis[Zintgraf+17]
    - Meaningful Perturbation[Fong-Vedaldi17]
    - FGVis(CVPR19)
  - Others: attention-based
    - needs special network design

Gradient/Sensitivity map

Simonyan+, Deep inside convolutional networks: Visualising image classifica- tion models and saliency maps, ICLR2014

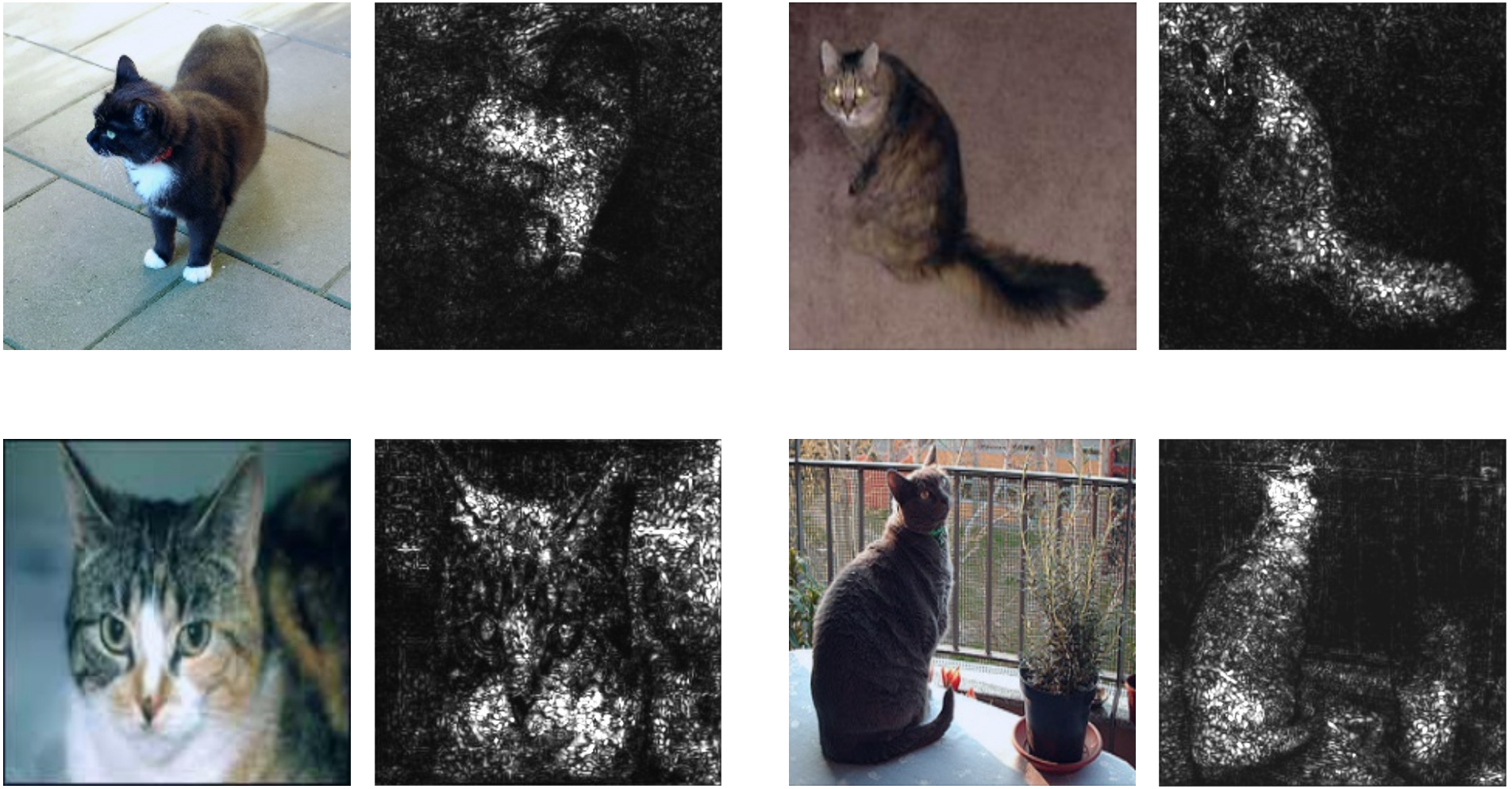
- How much impact does each input pixel have on a score?
  - Given by gradients of the score wrt. each input pixel

class-saliency map:
The absolute value of the gradients of a score of a particular class



Gradient/Sensitivity map

Simonyan+, Deep inside convolutional networks:Visualising image classifica- tion models and saliency maps, ICLR2014



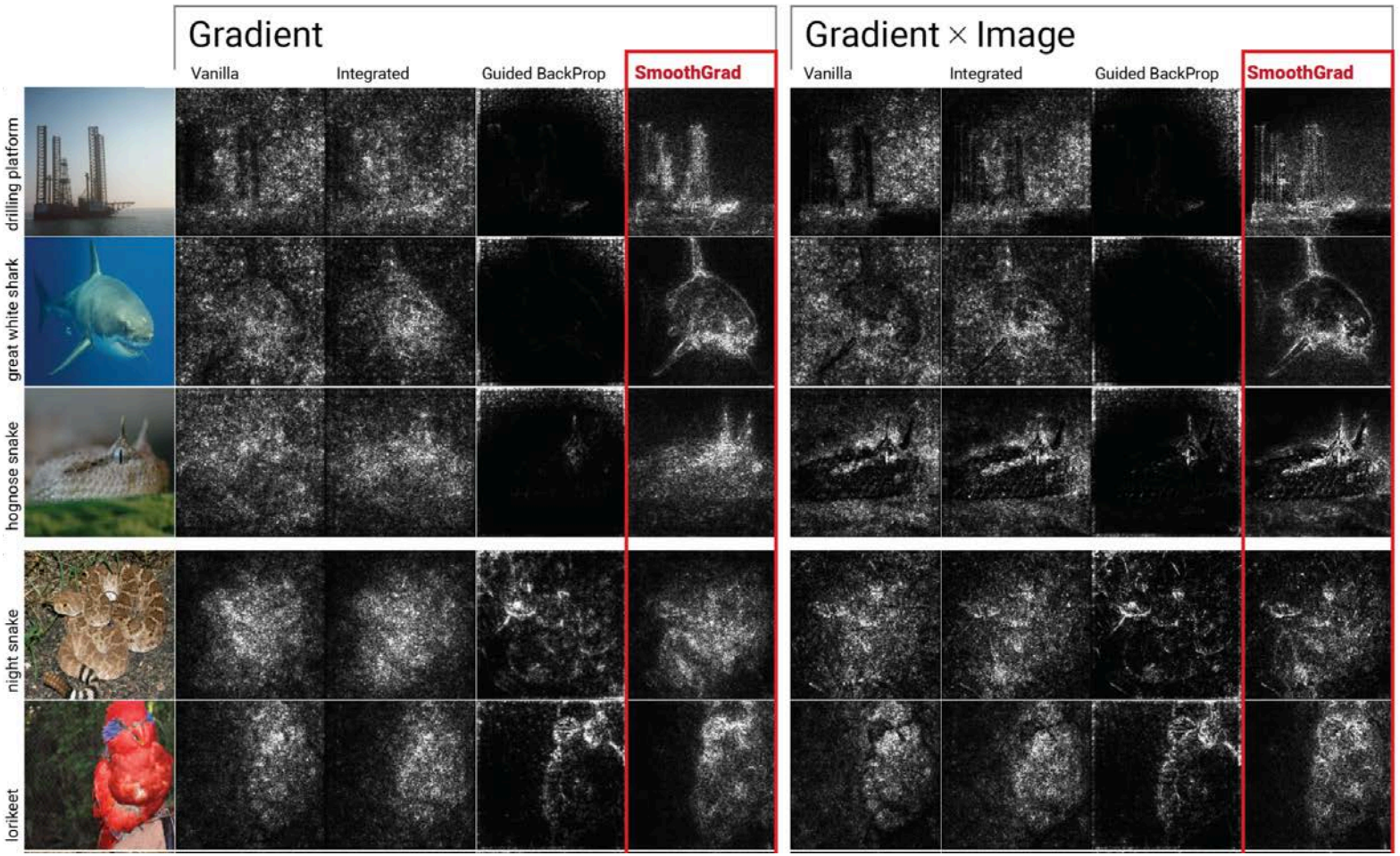
Gradient/Sensitivity map

Smilkov+, SmoothGrad: removing noise by adding noise, ICLR2017

- Gradient maps tend to by noisy

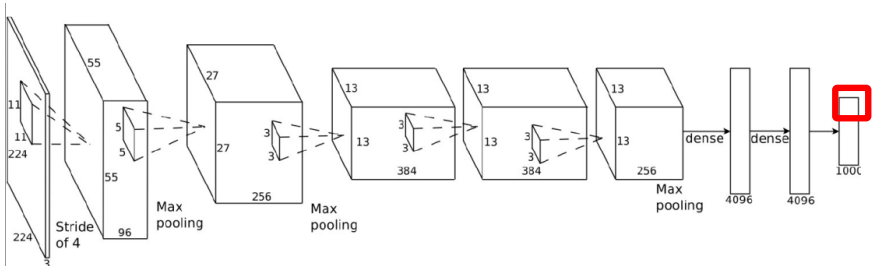
$$\hat{M}_c(x) = \frac{1}{n}\sum_{1}^{n} M_c(x + \mathcal{N}(0, \sigma^2))$$

→Add some noise to input images and take an average

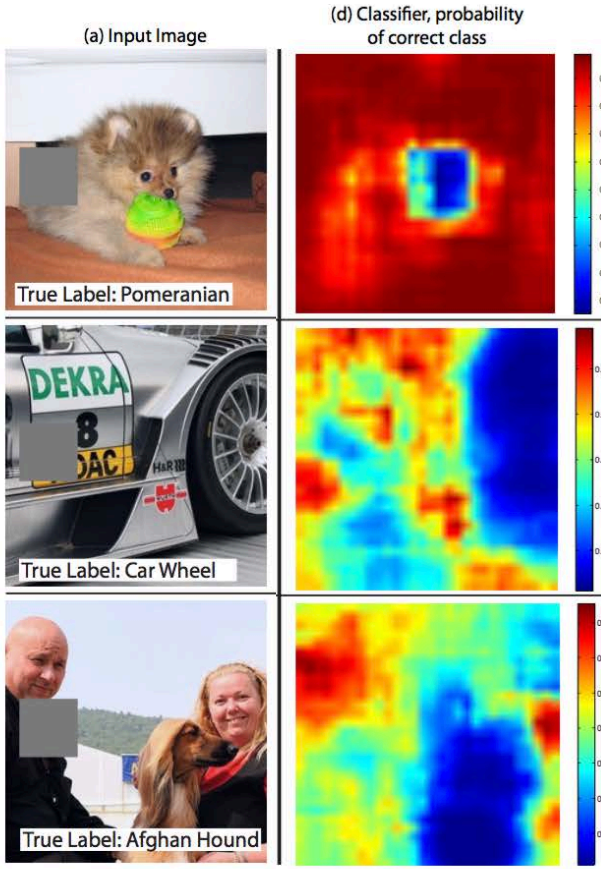


Masking part of input images

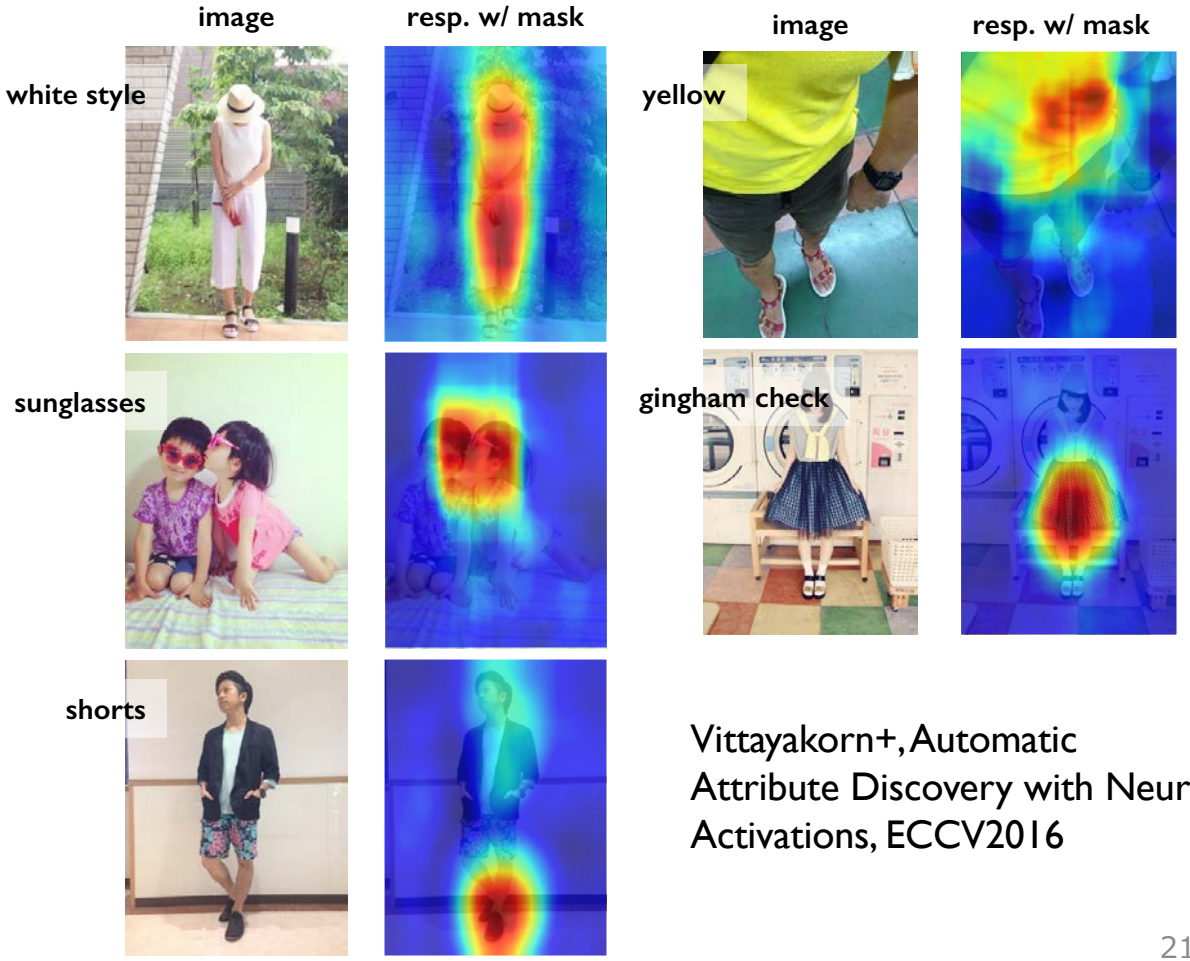
Square mask



Score of `Pomeranian'



[Zeiler+2014]



Vittayakorn+, Automatic Attribute Discovery with Neural Activations, ECCV2016

Class Activation Map(CAM)

Zhou+, Learning Deep Features for Discriminative Localization, CVPR16

- Activation of a global average pooling (GAP) layer is shown

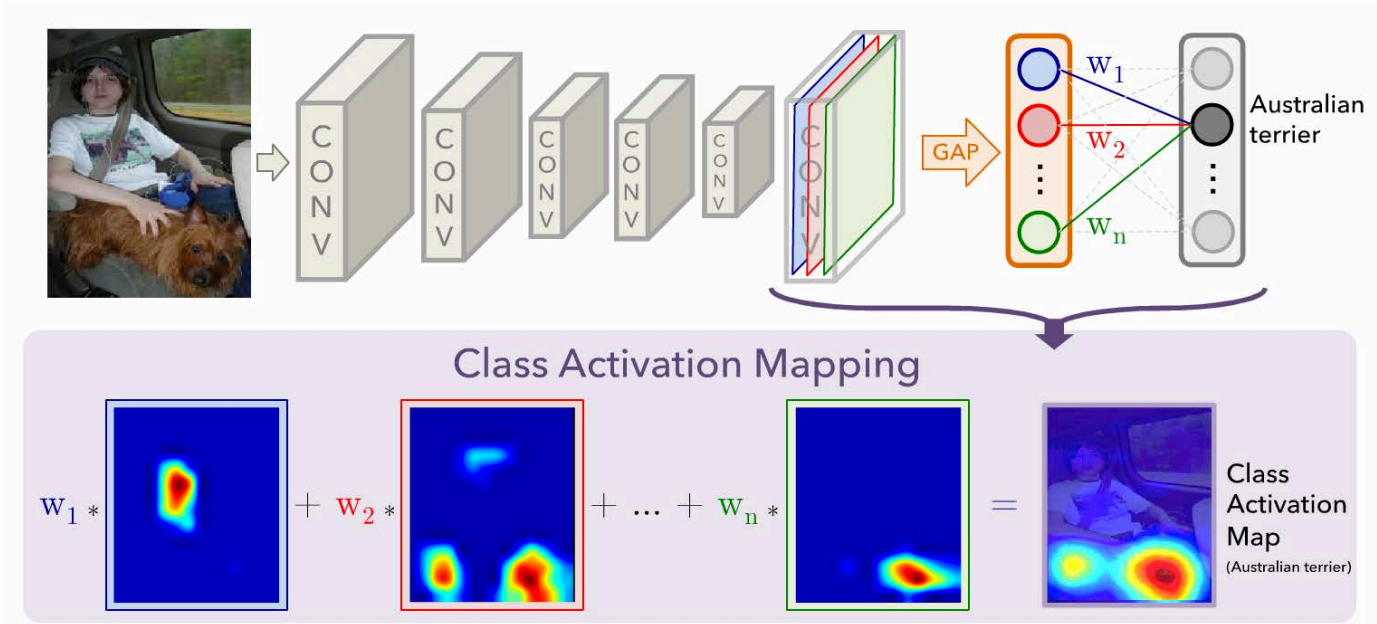
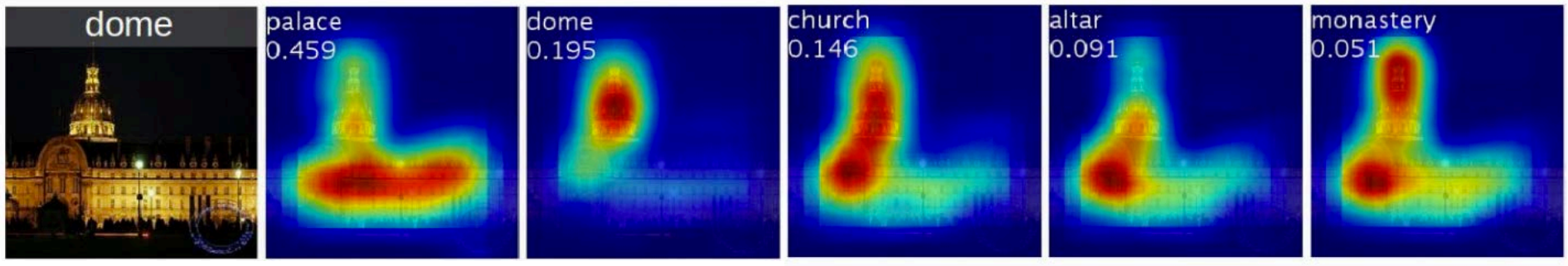


Figure 2. Class Activation Mapping: the predicted class score is mapped back to the previous convolutional layer to generate the class activation maps (CAMs). The CAM highlights the class-specific discriminative regions.

For CNNs without a GAP layer at their final section
→ gradCAM [Selvaraju+ICCV17]

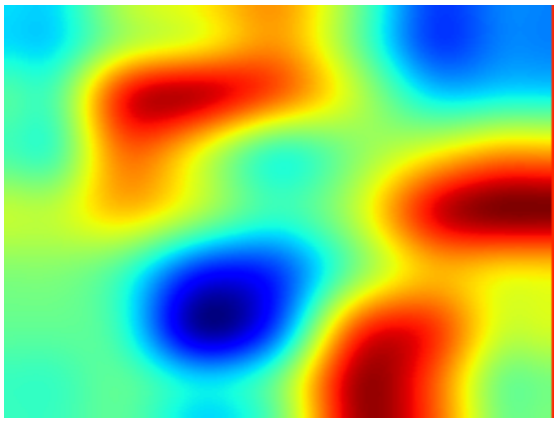


Visual recognition of surface qualities

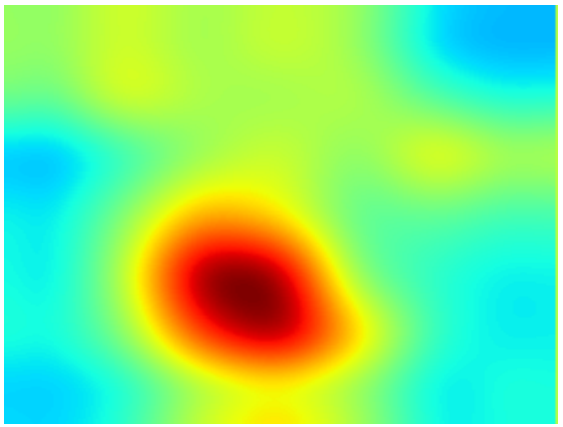
Liu+, Understanding Deep Representations Learned in CNNs for Material Recognition, VSS2016

INPUT

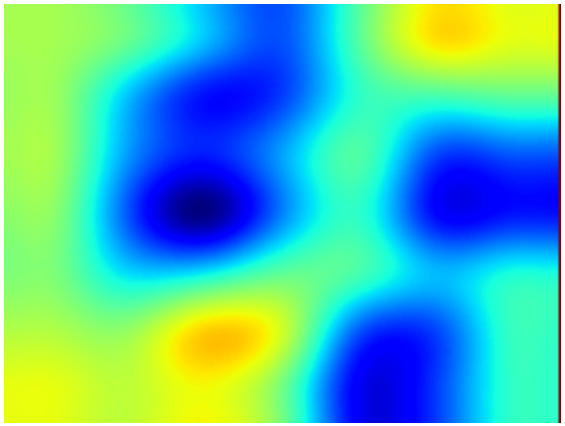
aged



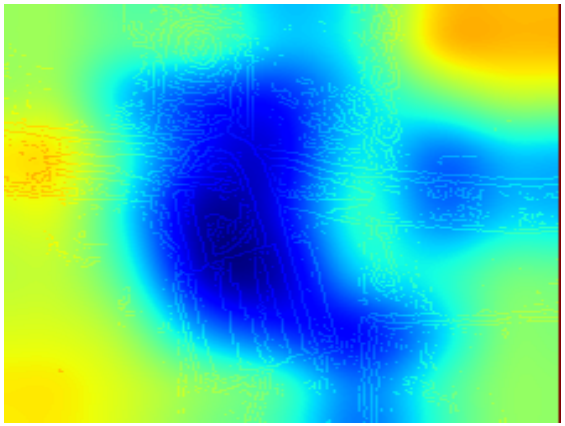
cold



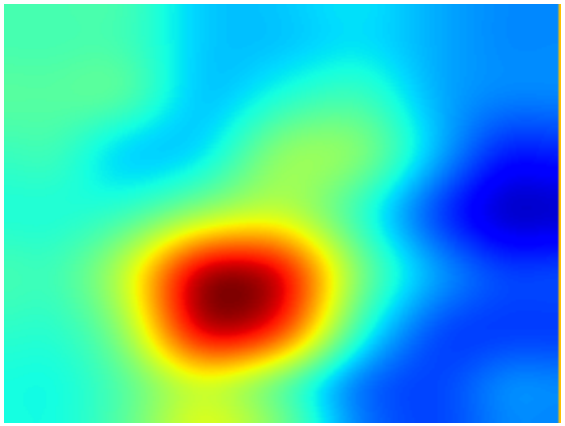
clean



fragile



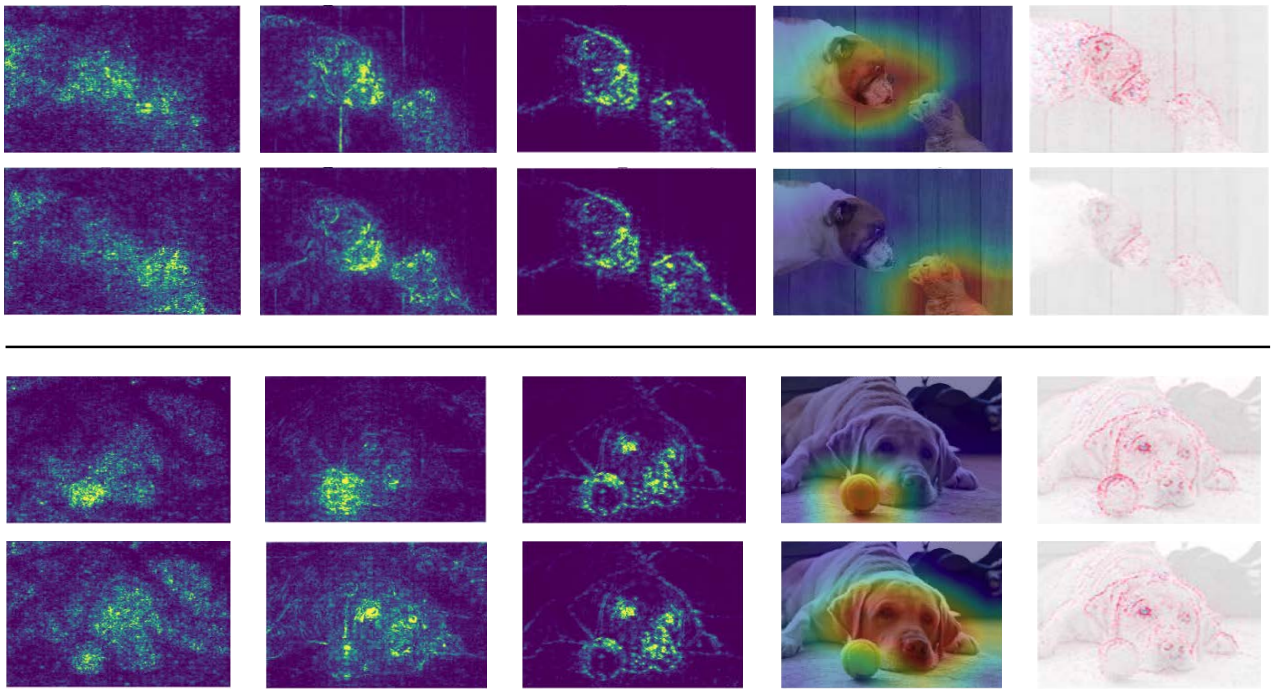
gloss



Summary: visualization of inference

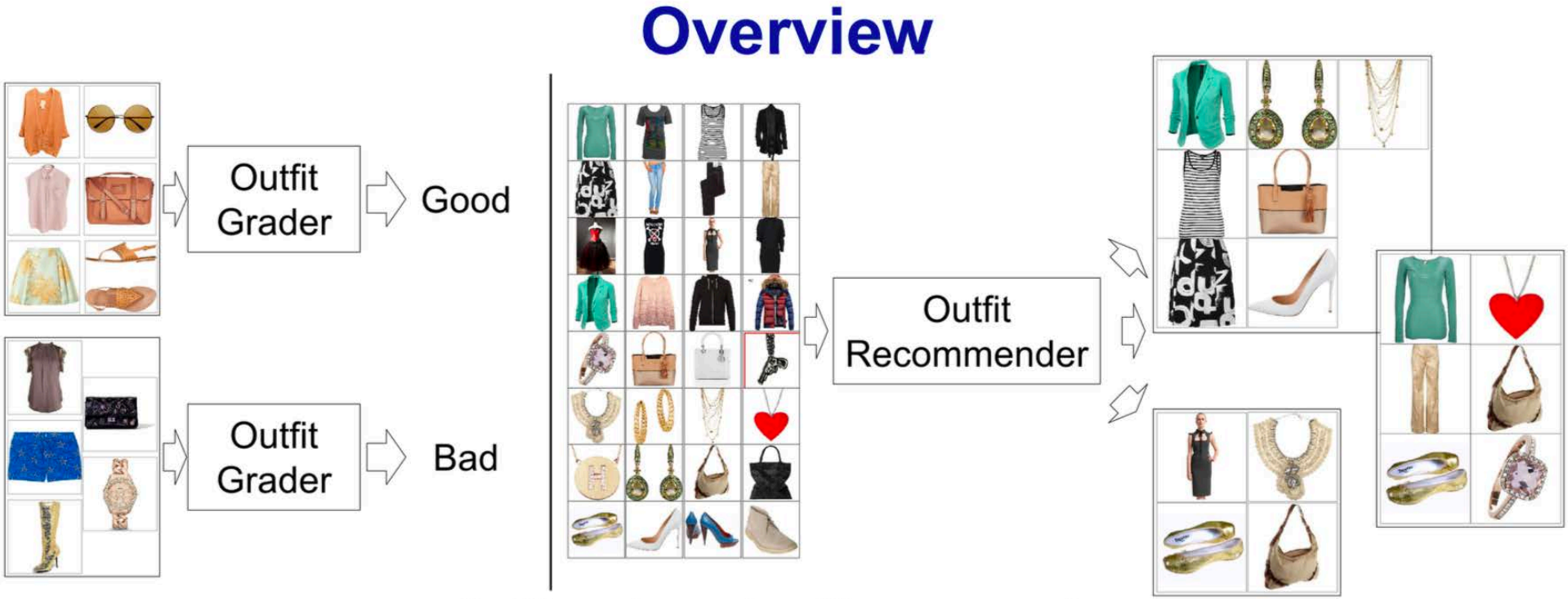
0.86318: boxer
0.09939: French Bulldog
0.01579: bull mastiff
0.00356: American Staffordshire terrier
0.00209: Staffordshire bullterrier

0.90761: tennis ball
0.04521: Chesapeake Bay retriever
0.02405: Labrador retriever
0.00741: viszla
0.00410: bloodhound



“Smart Closet”

Tangseng+, Recommending Outfits from Personal Closet, WACV18



Outfit grader and outfit recommender

Scenario: Create outfits from items.

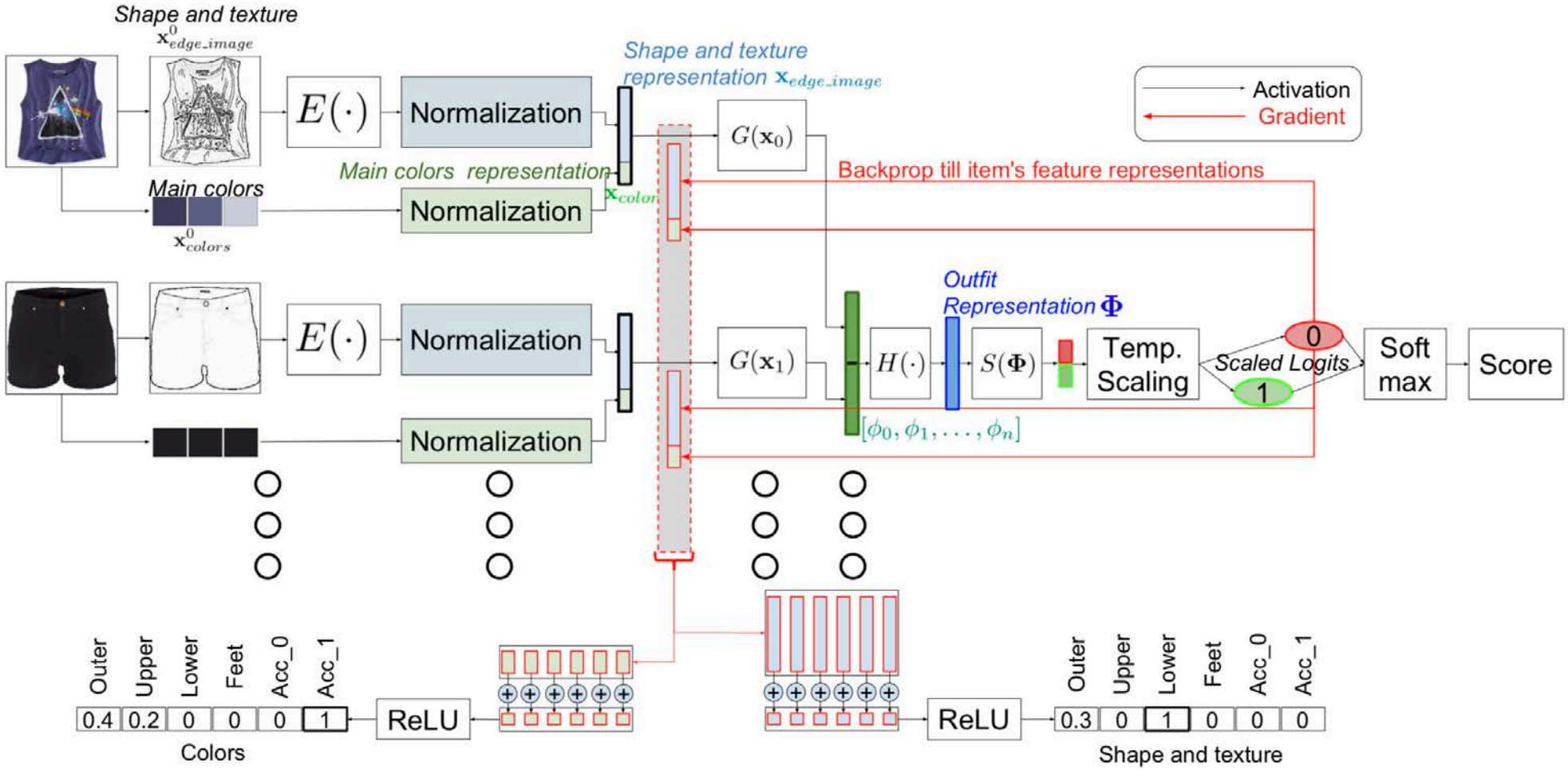
Approach: Beam search with an outfit grader as a value function.

Contribution: dataset, outfit grader, outfit recommender

Explainable fashion recommendation

Tangseng-Okatani, Toward Explainable Fashion Recommendation, WACV2019

- We estimate the impact on the final score of each item-feature by using their gradients times layer activation



Explainable fashion recommendation

Tangseng-Okatani, Toward Explainable Fashion Recommendation, WACV2019

- This reveals which feature of which item of an outfit makes our network give a bad score