Computer Vision

CNN Visualization

Tae-Hyun Oh (오태현)
전자전기공학과
POSTECH

Slide by Juyong Lee (이주용)
TAs: {Dongmin Choi , Jongha Kim, Juyong Lee, Sungbin Kim} (in alphabetic order)

boostcamp AI Tech

1. Visualizing CNN

1.1 What is CNN visualization?
1.2 Vanilla example: filter visualization
1.3 How to visualize neural network

2. Analysis of model behaviors

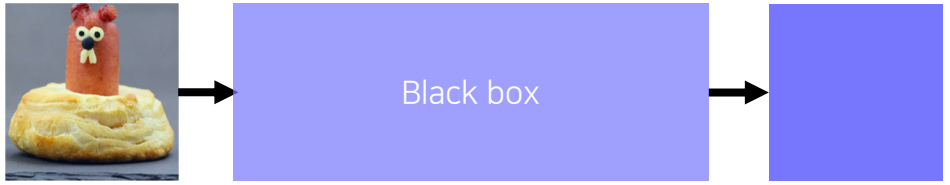
2.1 Embedding feature analysis
2.2 Activation investigation

3. Model decision explanation

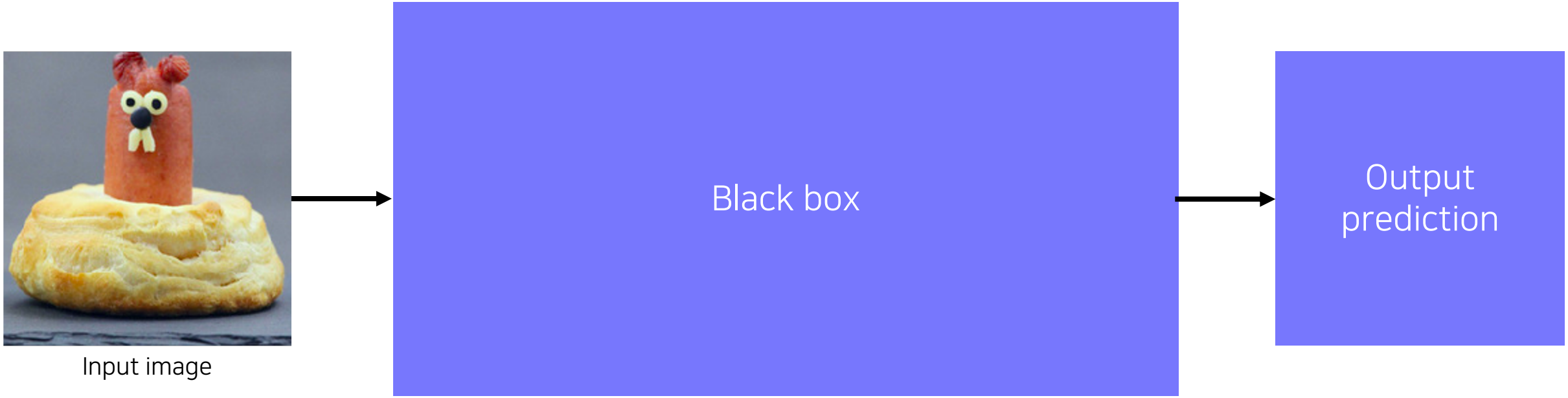
3.1 Saliency test
3.2 Backpropagate features
3.3 Class activation mapping

1.
Visualizing CNN

1.1 What is CNN visualization?



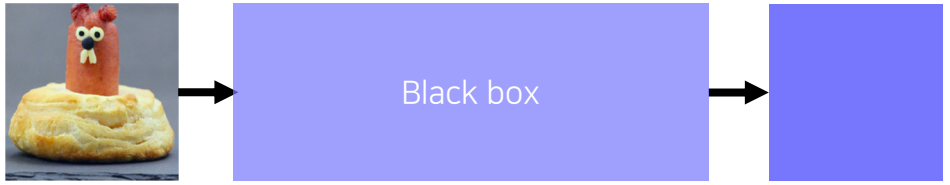
CNN is a black box



Input image

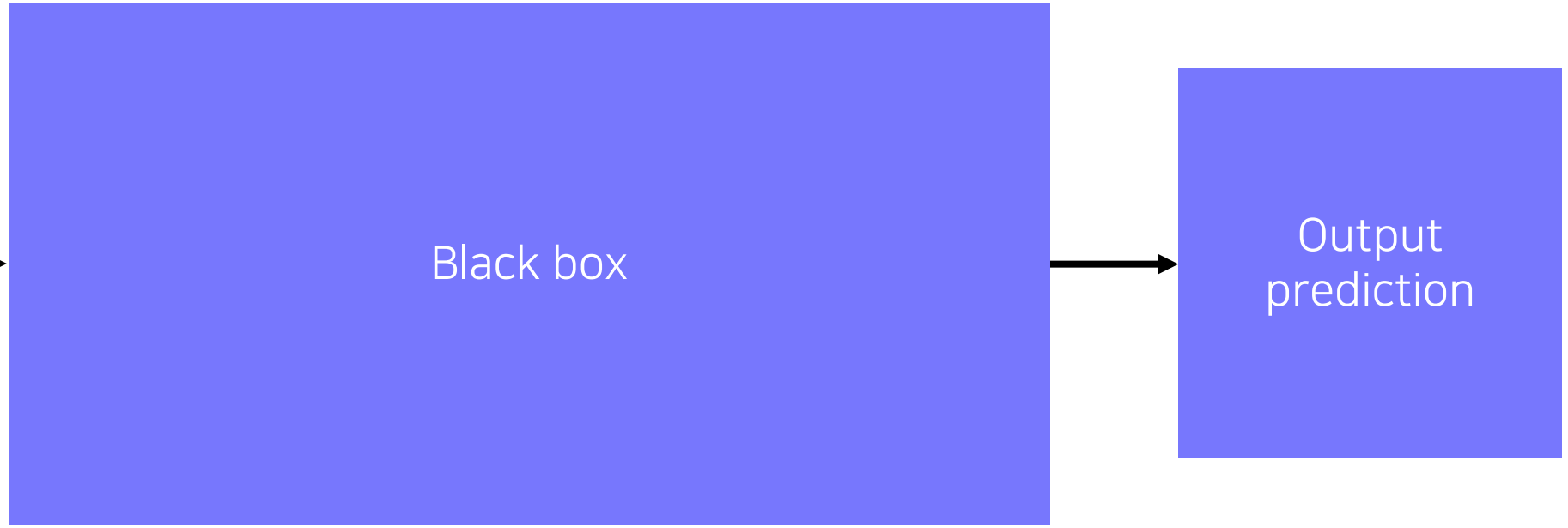
1) What is inside CNNs (black box)?

1.1 What is CNN visualization?



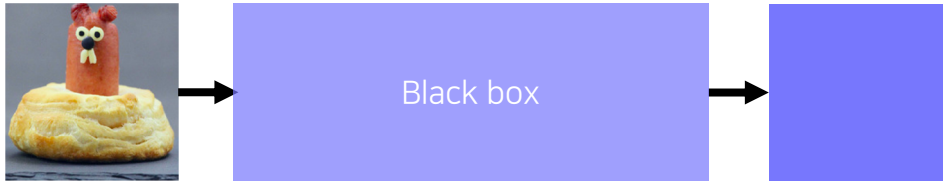
CNN is a black box

Input image



1) What is inside CNNs (black box)?
2) Why do they perform so well?

1.1 What is CNN visualization?



CNN is a black box

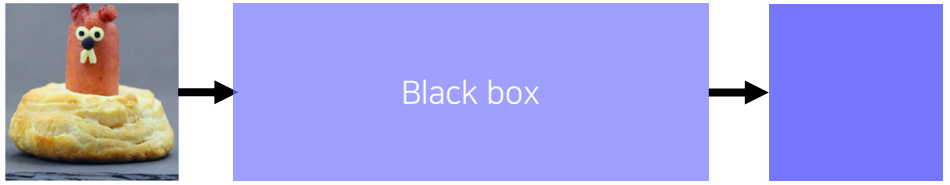
Input image

Black box

Output prediction

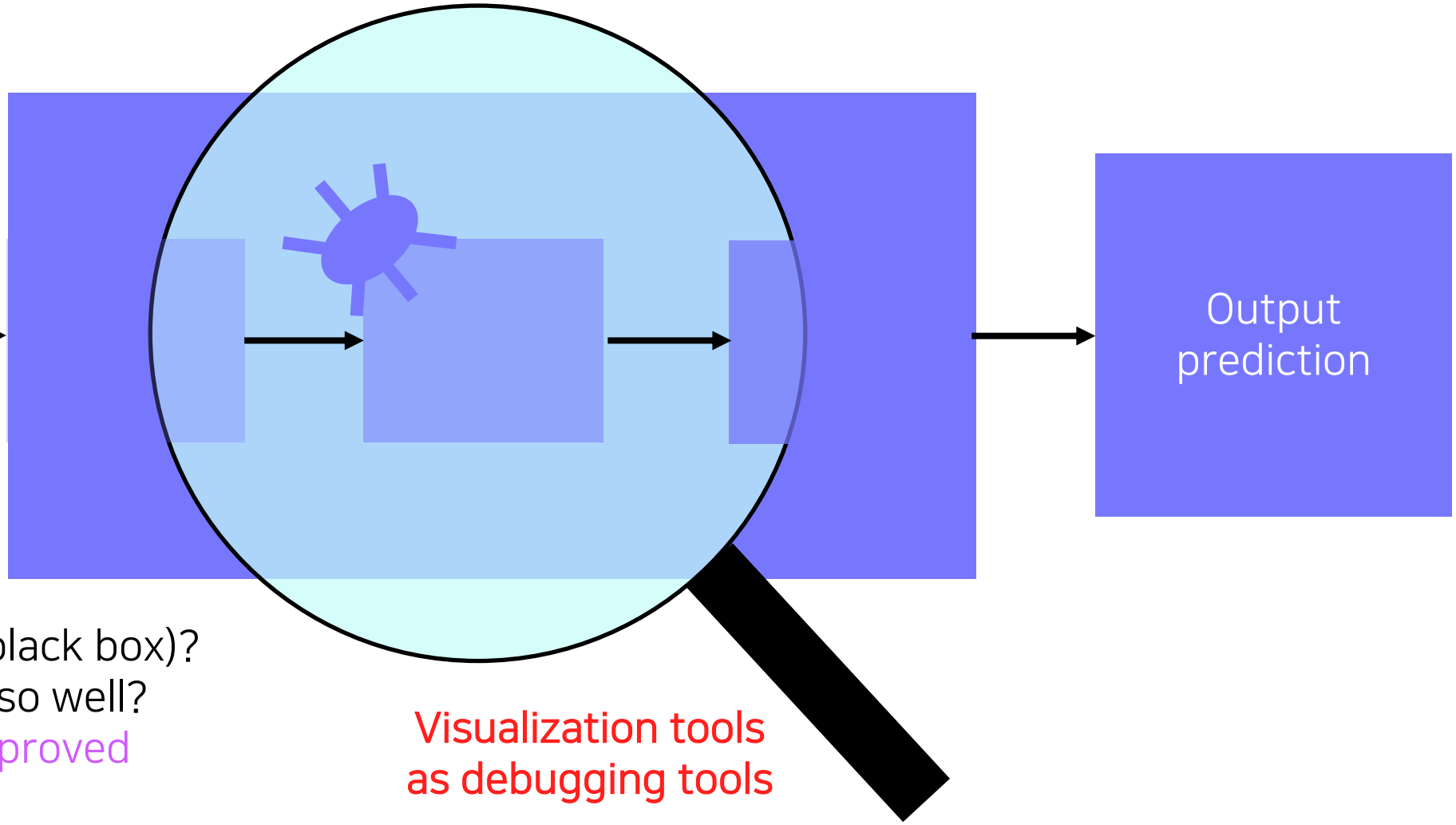
1) What is inside CNNs (black box)?
2) Why do they perform so well?
3) How would they be improved

1.1 What is CNN visualization?



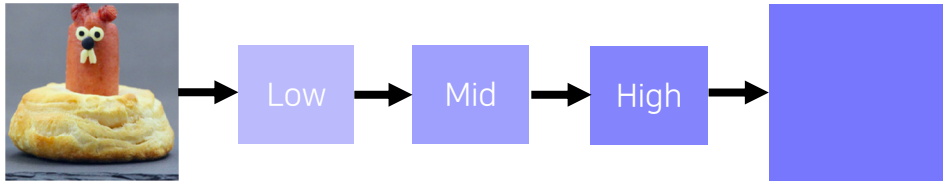
CNN is a black box

Input image



1) What is inside CNNs (black box)?
2) Why do they perform so well?
3) How would they be improved

1.1 What is CNN visualization?



ZFNet example – the winner of ImageNet Challenge 2013

[Zeiler and Fergus, ECCV 2014]

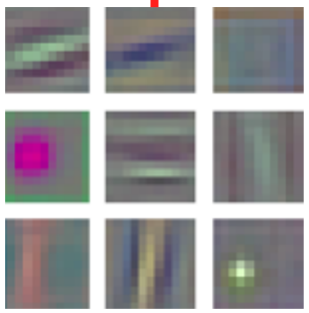
Input image

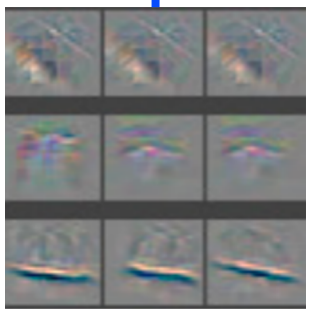
Low level feature

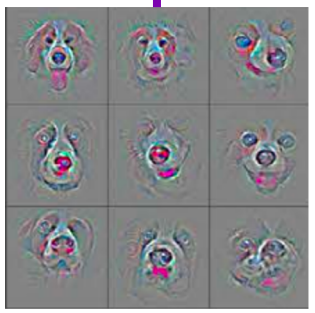
Middle level feature

High level feature

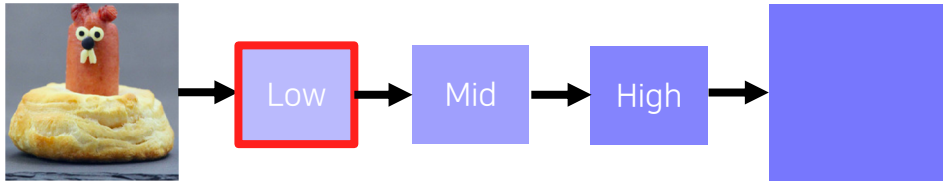
Output prediction







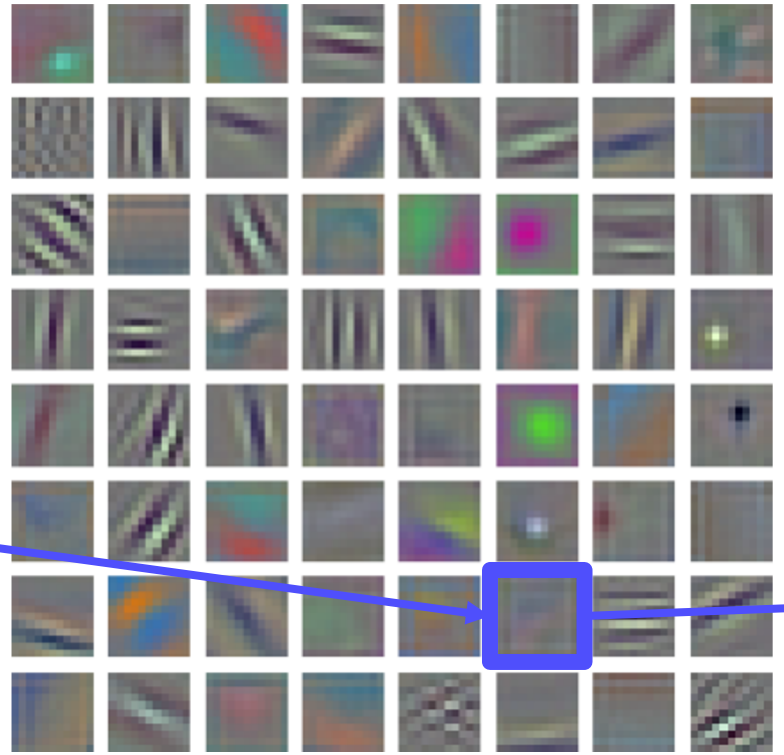
1.2 Vanilla example: filter visualization



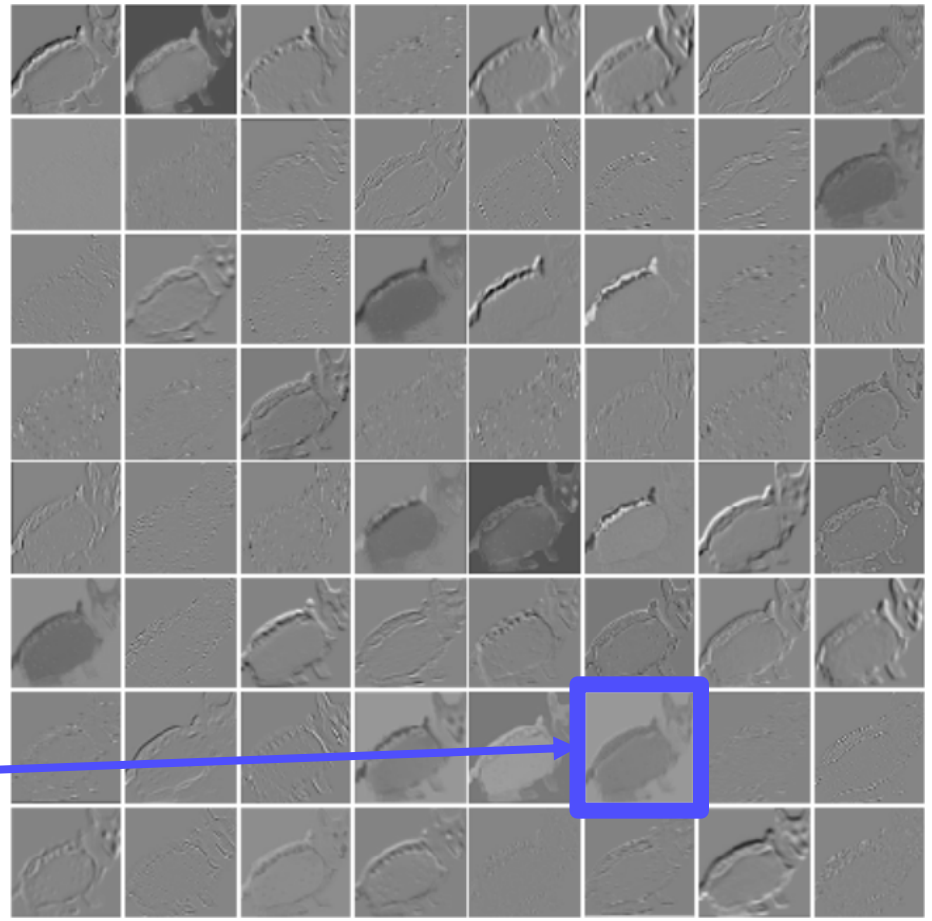
Filter weight visualization

Input image

×

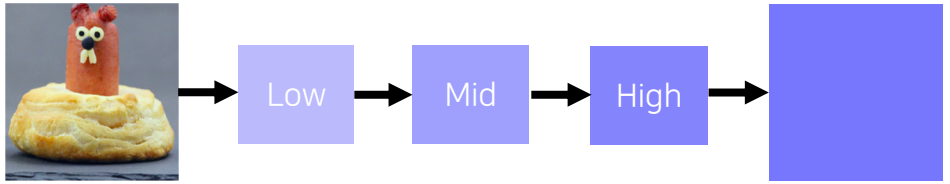


Filter visualization (1st conv. layer)



Activation visualization (1st conv. layer)

1.3 How to visualize neural network



Types of neural network visualization

Analysis of **model behaviors**

Model decision explanation

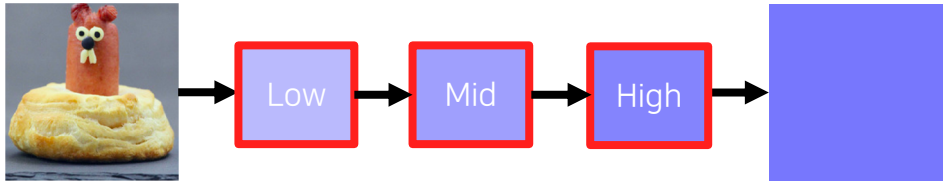
Parameter examination	Feature analysis	Sensitivity analysis	Decomposition
Filter visualization Factorization lens	t-SNE Gradient ascent	Saliency map GradCAM	DeepLIFT LRP

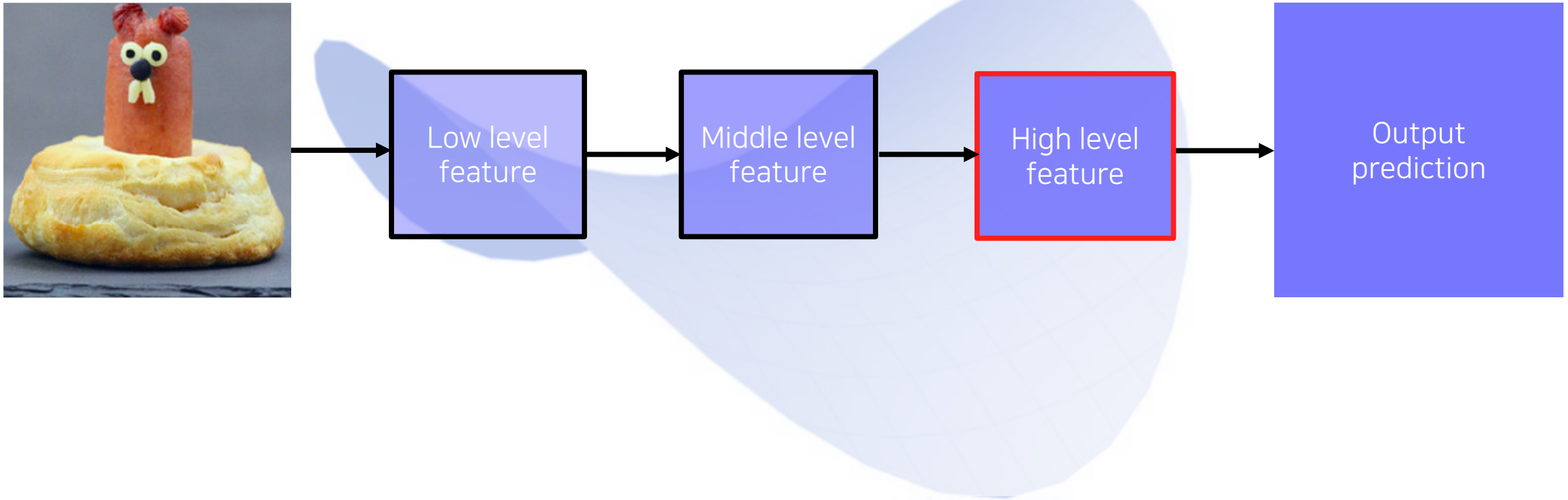
focus on **models** ⟷ focus on **data**

2.

Analysis of model behaviors

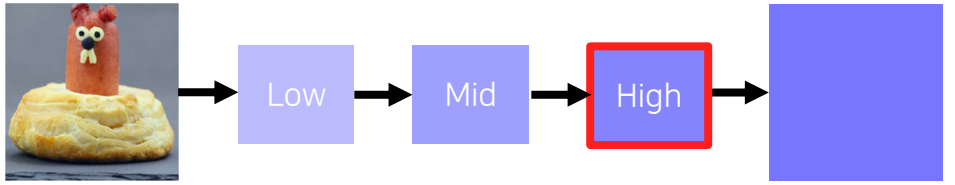
2.1 Embedding feature analysis





[https://www.h-its.org/projects/hyperbolic-polyspheriicons/]

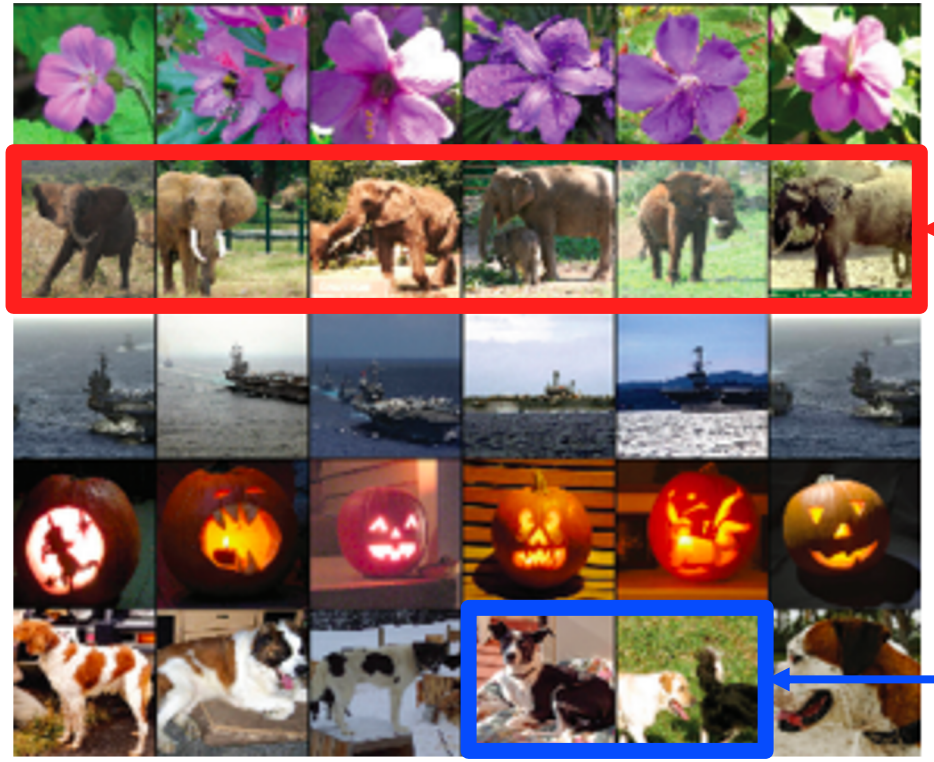
2.1 Embedding feature analysis 1



Nearest neighbors (NN) in a feature space - Example

[Krizhevsky et al., NIPS 2012]

Query images

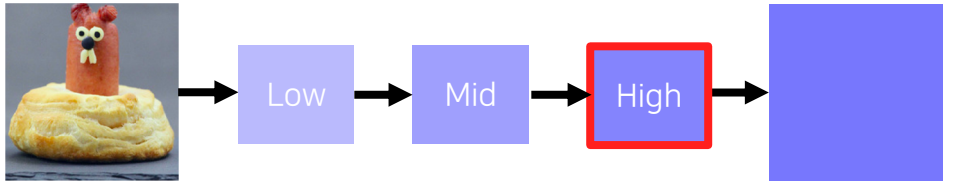


Top-6 neighbors in the feature space

We can notice semantically similar concepts are well clustered

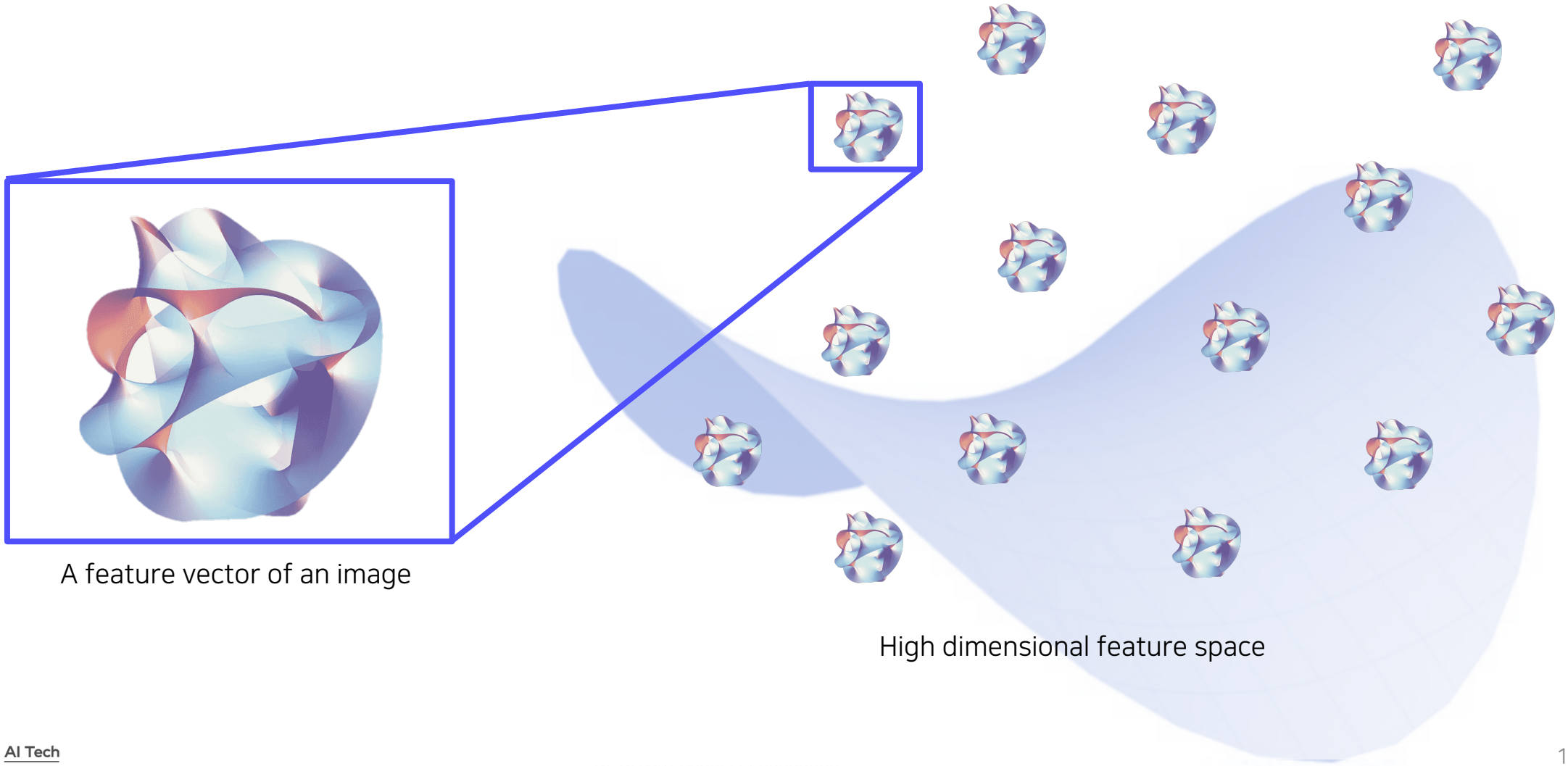
The embedding feature comparison does not have the limitation of the simple pixel-wise comparison

2.1 Embedding feature analysis 1



Nearest neighbors in a feature space

[Krizhevsky et al., NIPS 2012]

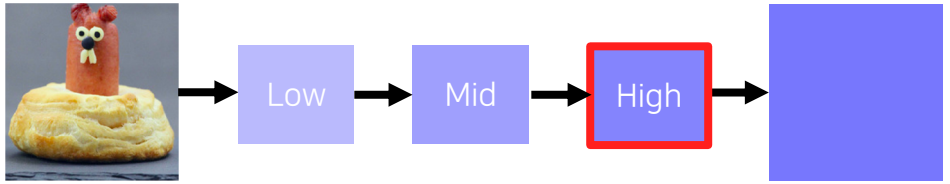


A feature vector of an image

High dimensional feature space

[https://www.h-its.org/projects/hyperbolic-polyspheriicons/, https://www.pngegg.com/en/png-tktnn]

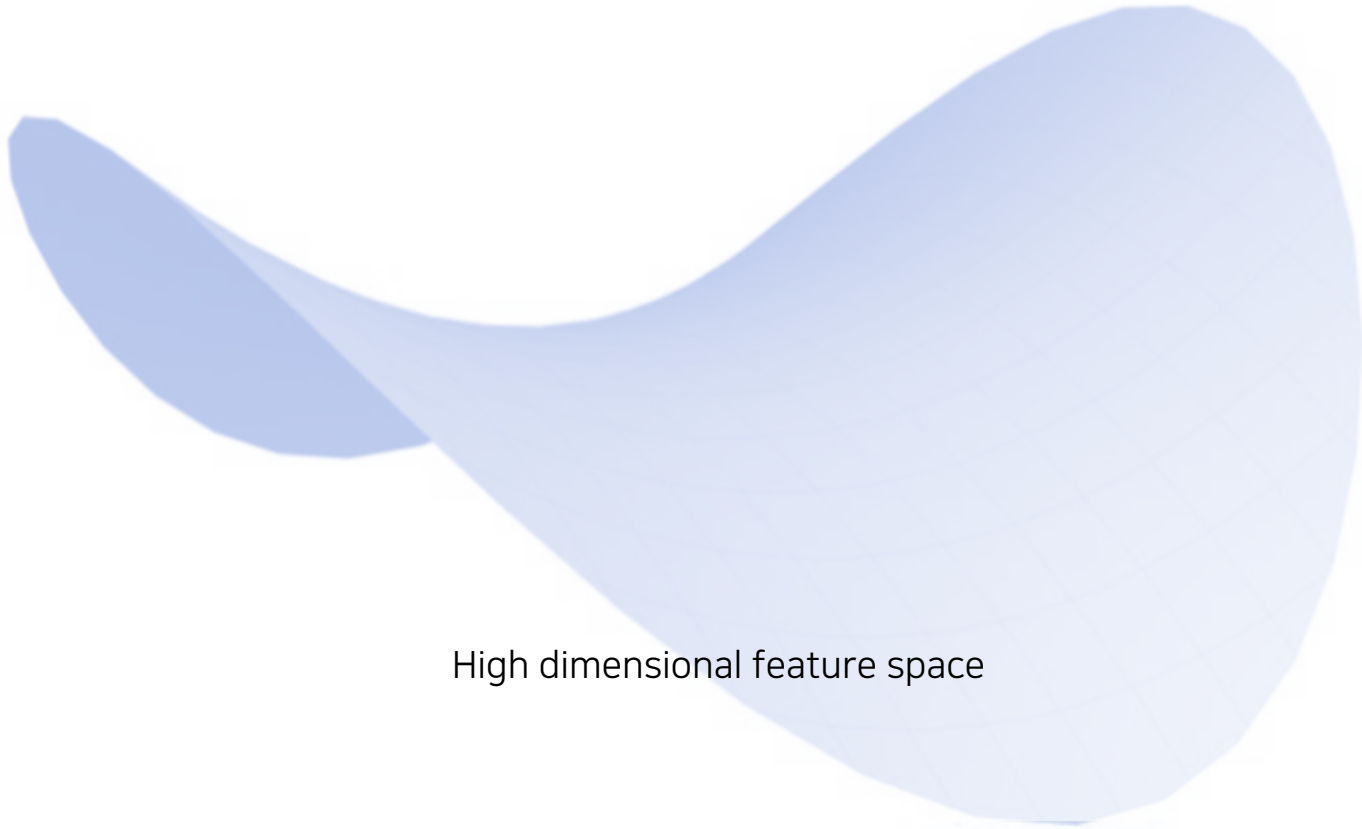
2.1 Embedding feature analysis 1



Nearest neighbors in a feature space

[Krizhevsky et al., NIPS 2012]

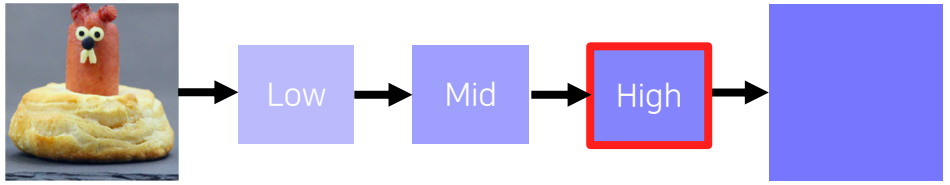
Neural networks



High dimensional feature space

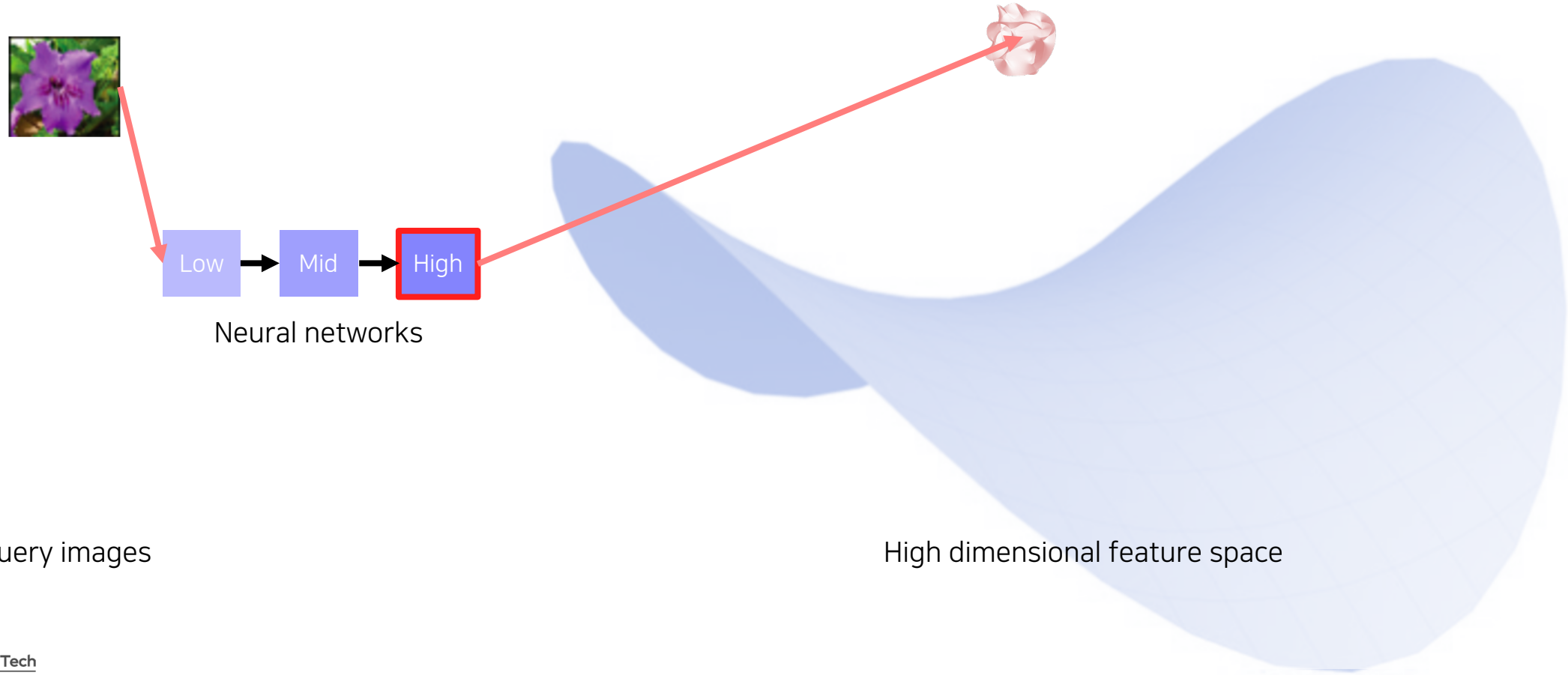
[https://www.h-its.org/projects/hyperbolic-polyspheriicons/, https://www.pngegg.com/en/png-tktnn]

2.1 Embedding feature analysis 1



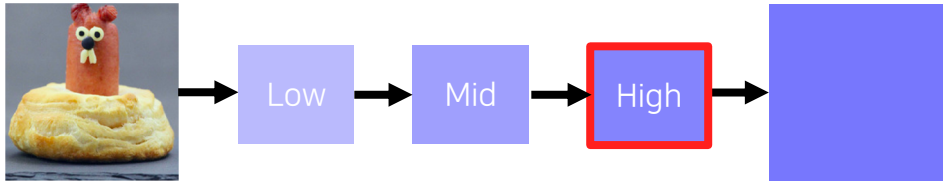
Nearest neighbors in a feature space

[Krizhevsky et al., NIPS 2012]



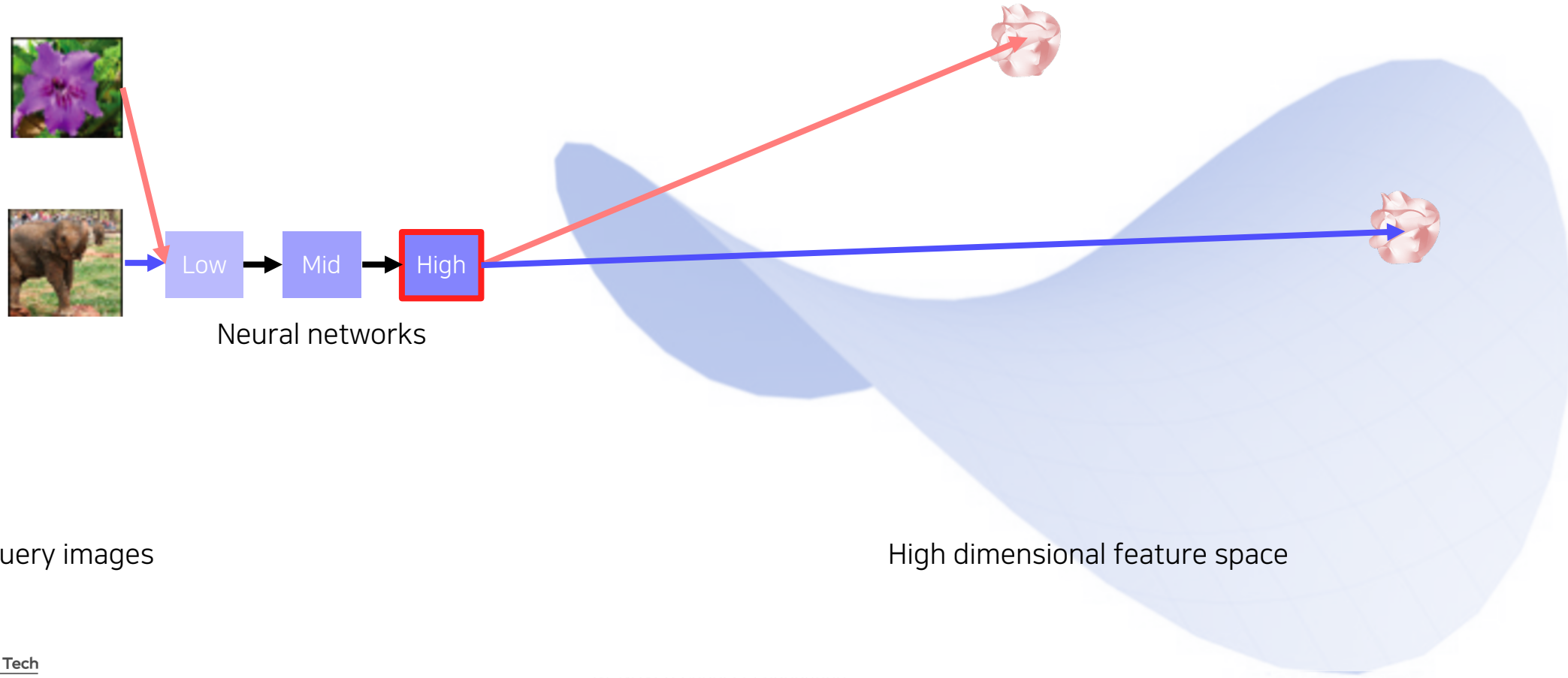
[https://www.h-its.org/projects/hyperbolic-polyspheriicons/, https://www.pngegg.com/en/png-tktnn]

2.1 Embedding feature analysis 1



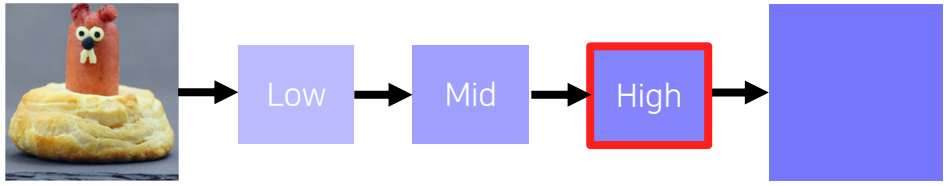
Nearest neighbors in a feature space

[Krizhevsky et al., NIPS 2012]



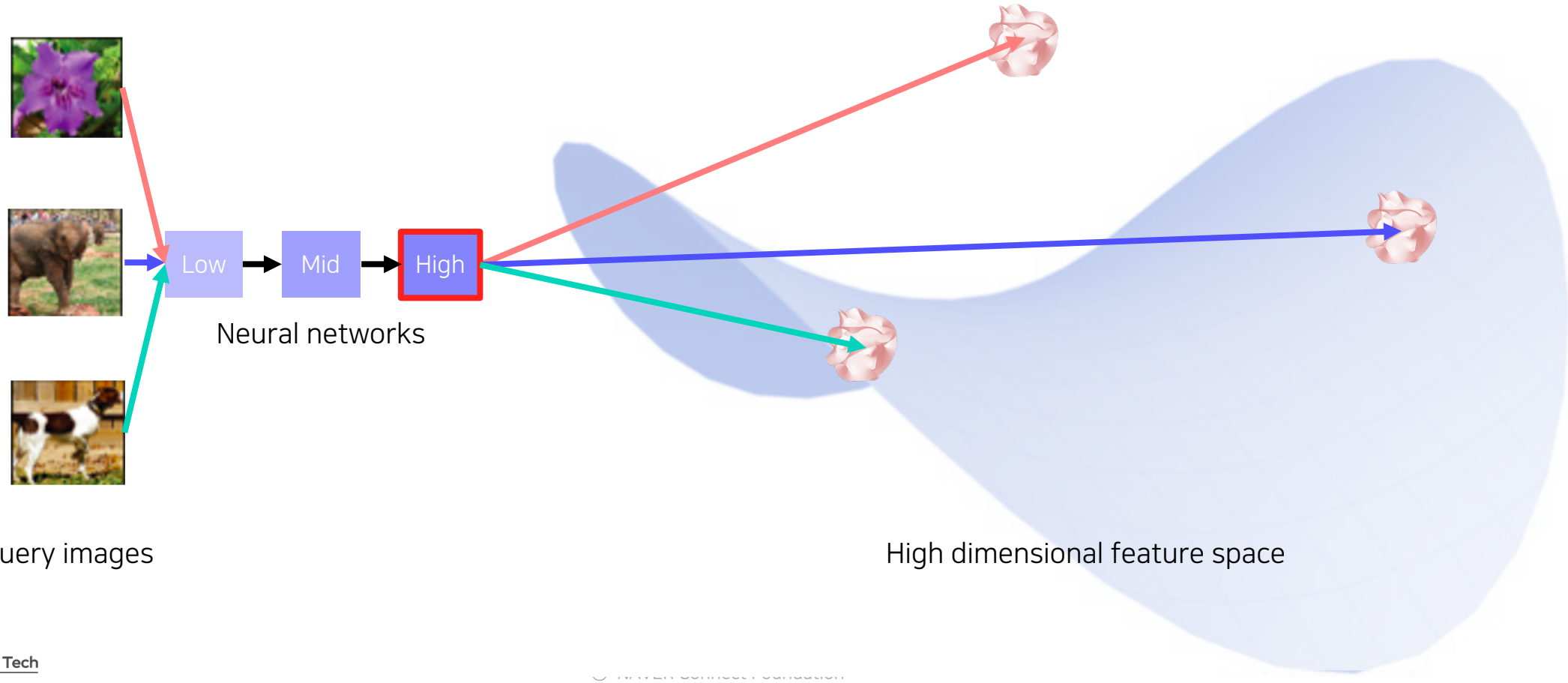
[https://www.h-its.org/projects/hyperbolic-polyspheriicons/, https://www.pngegg.com/en/png-tktnn]

2.1 Embedding feature analysis 1



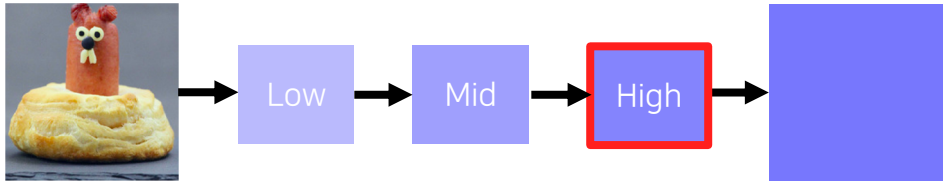
Nearest neighbors in a feature space

[Krizhevsky et al., NIPS 2012]



[https://www.h-its.org/projects/hyperbolic-polyspheriicons/, https://www.pngegg.com/en/png-tktnn]

2.1 Embedding feature analysis 1



Nearest neighbors in a feature space – Search the nearest neighbors of features from DB

[Krizhevsky et al., NIPS 2012]

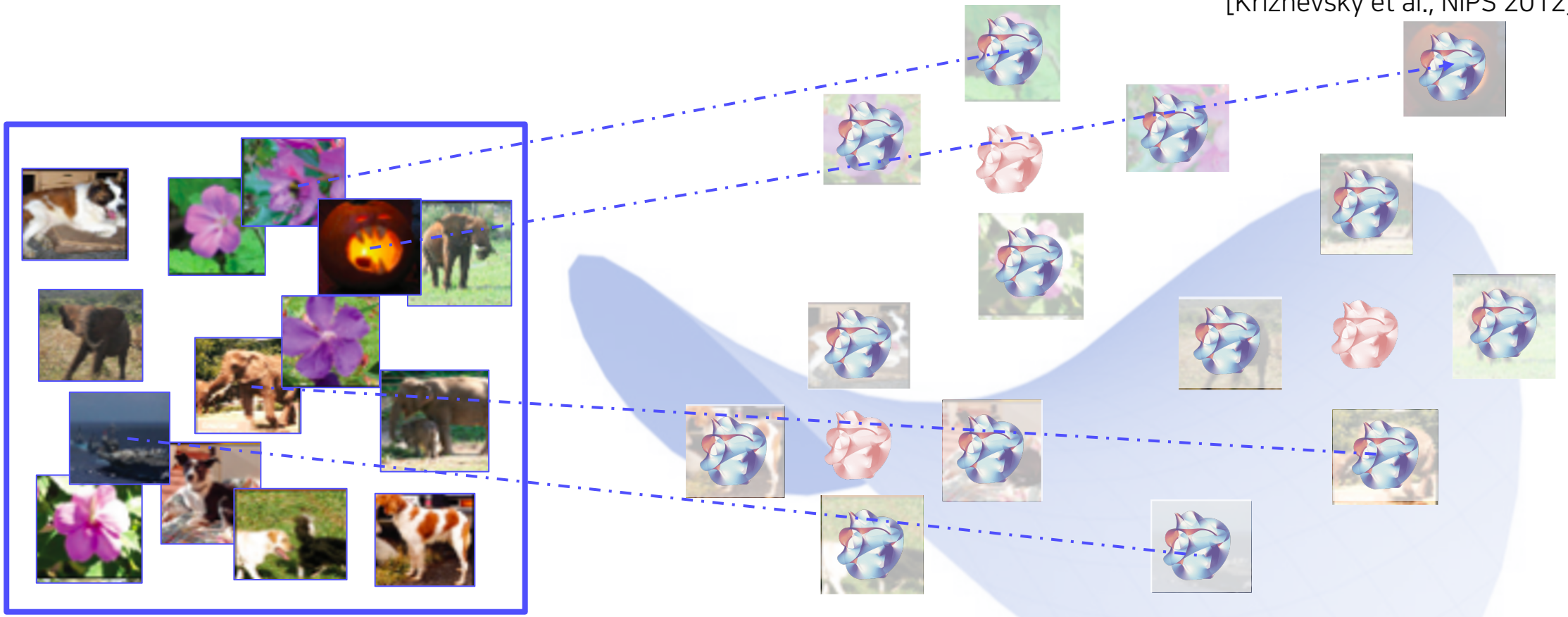
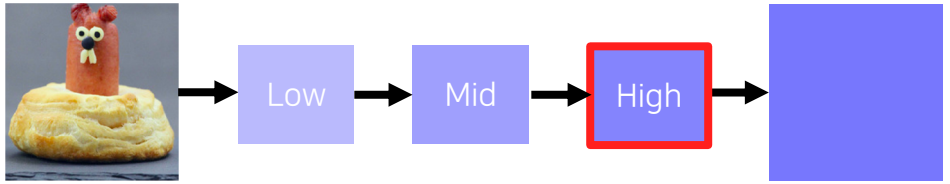


Image database

High dimensional feature space

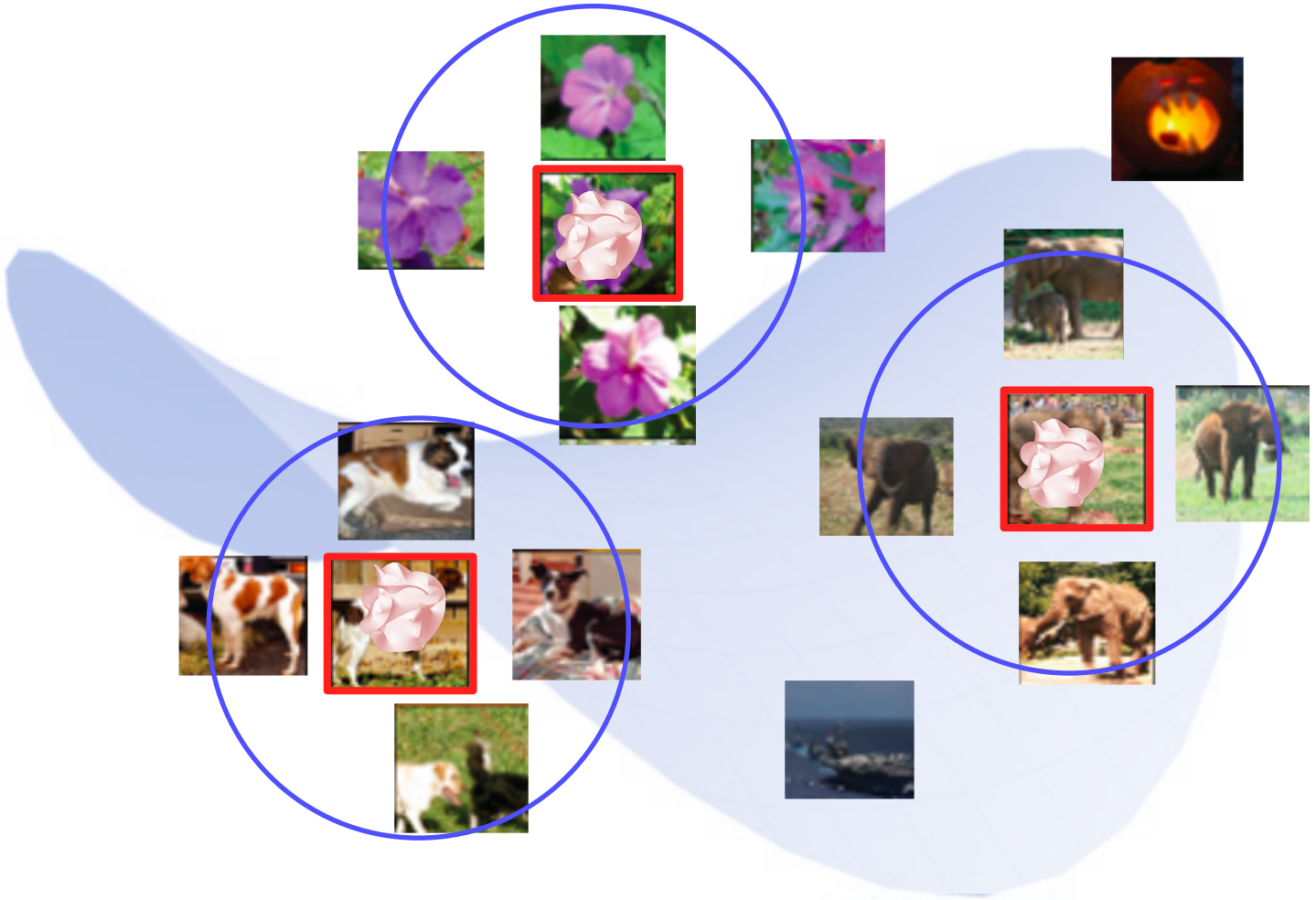
[https://www.h-its.org/projects/hyperbolic-polyspheriicons/, https://www.pngegg.com/en/png-tktnn]

2.1 Embedding feature analysis 1



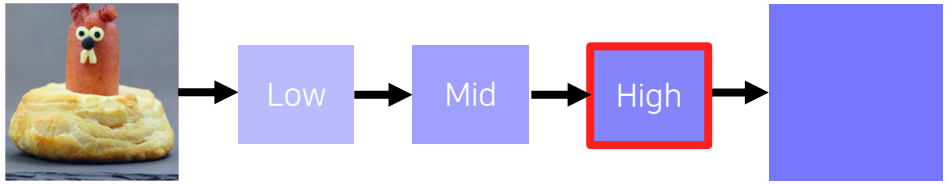
Nearest neighbors in a feature space – Analysis by searched neighbor images

[Krizhevsky et al., NIPS 2012]

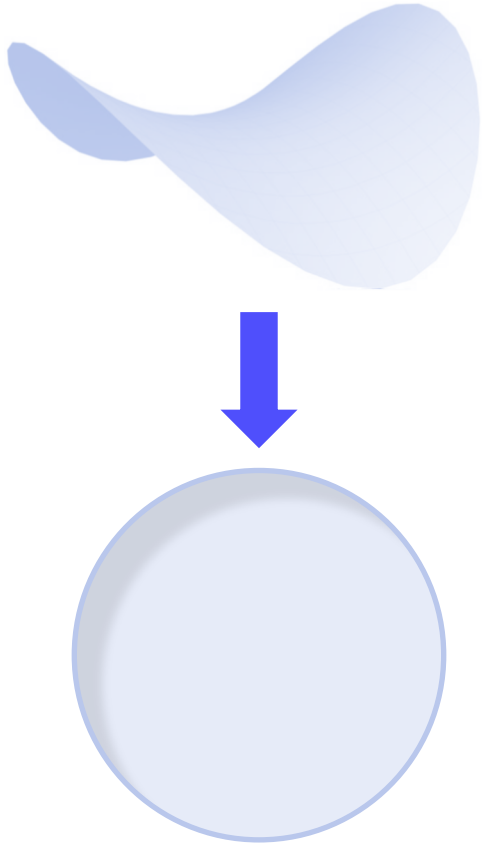


[https://www.h-its.org/projects/hyperbolic-polyspheric ons/, https://www.pngegg.com/en/png-tktnn]

2.1 Embedding feature analysis 2



Dimensionality reduction



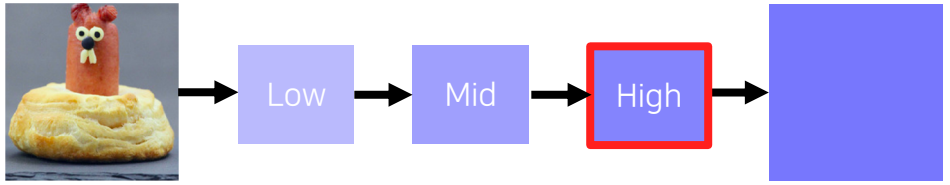
High dimensional space

Low dimensional space (better interpretable)

Illustration of dimension reduction of a high dim. feature space into a 2D space

[https://www.h-its.org/projects/hyperbolic-polyspheric ons/]

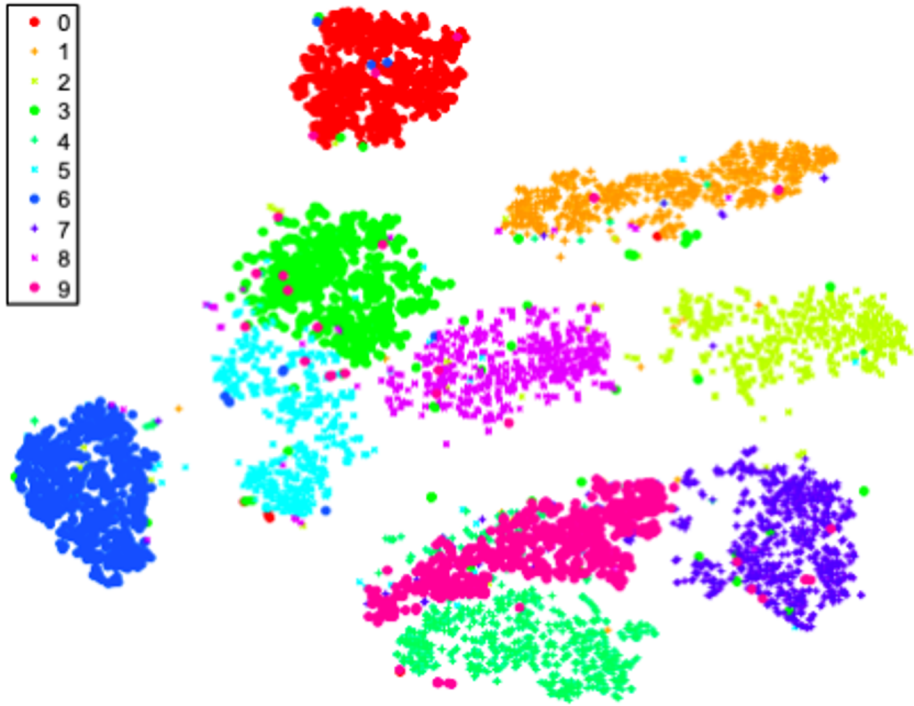
2.1 Embedding feature analysis 2



Dimensionality reduction

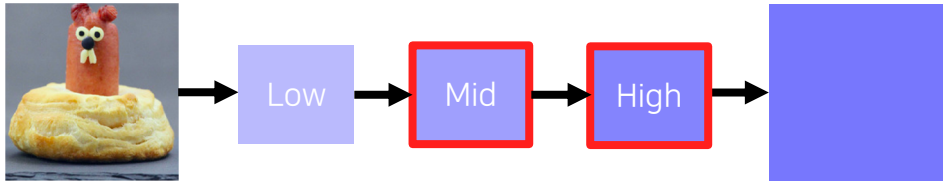
[Maaten and Hinton, JMLR 2008]

- t-distributed stochastic neighbor embedding (t-SNE)



Visualization on geometry of feature space
(experiment with MNIST dataset)

2.2 Activation investigation 1



Layer activation – Behaviors of mid- to high-level hidden units

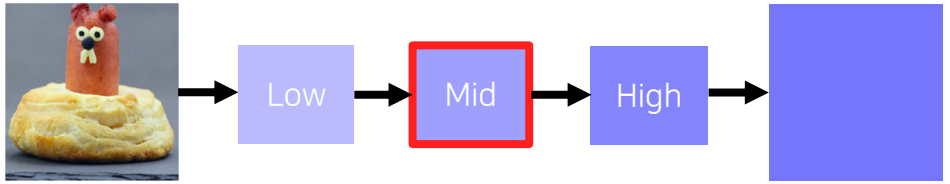
[Bau et al., CVPR 2017]

AlexNet-Places205
conv5 unit 138: heads

AlexNet-Places205
conv5 unit 53: stairways

High activation mask visualization
(The masks are generated by thresholding)

2.2 Activation investigation 2



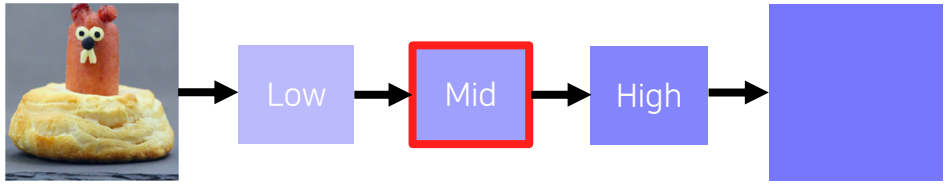
Maximally activating patches - Example

[Springenberg et al., ICLR 2015]

Hidden nodes

Image patches obtained from
the locations of maximum activations of each hidden node

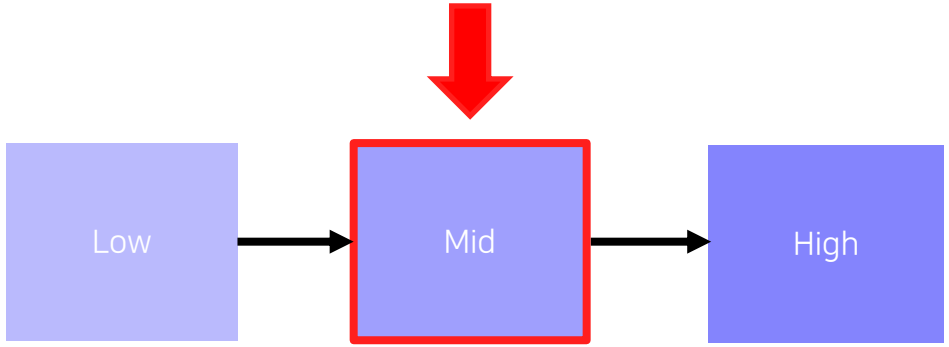
2.2 Activation investigation 2



Maximally activating patches – Patch acquisition

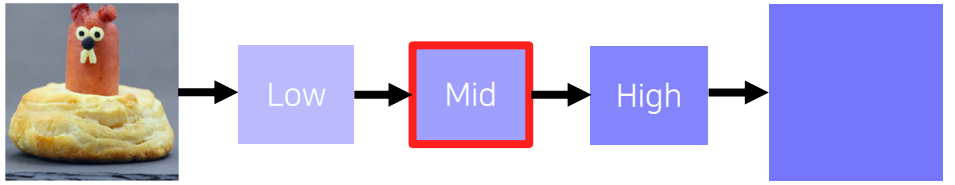
[Springenberg et al., ICLR 2015]

- 1) Pick a channel in a certain layer
- 2) Feed a chunk of images and record each activation value (of the chosen channel)
- 3) Crop image patches around maximum activation values



e.g., channel 14/256 of conv. 5 layer

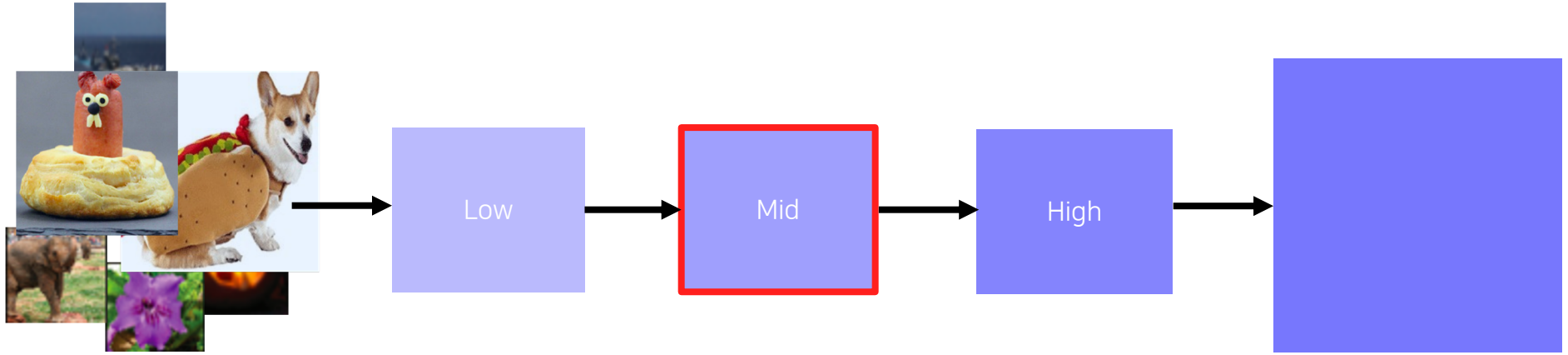
2.2 Activation investigation 2



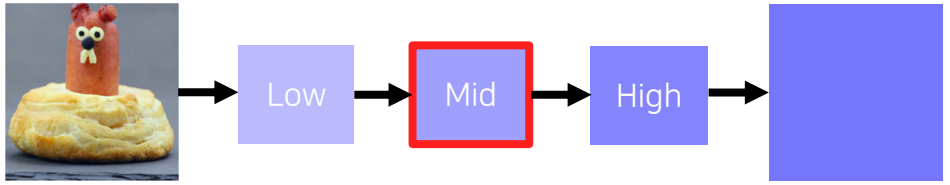
Maximally activating patches – Patch acquisition

[Springenberg et al., ICLR 2015]

- 1) Pick a channel in a certain layer
- 2) Feed a chunk of images and record each activation value (of the chosen channel)
- 3) Crop image patches around maximum activation values



2.2 Activation investigation 2

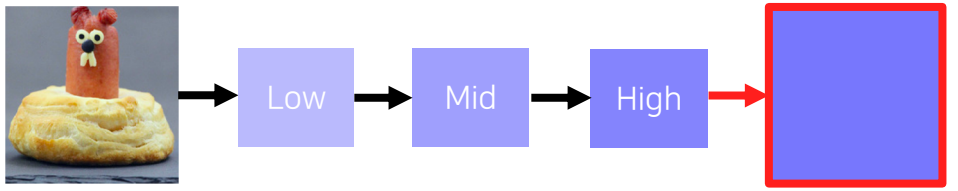


Maximally activating patches – Patch acquisition

[Springenberg et al., ICLR 2015]

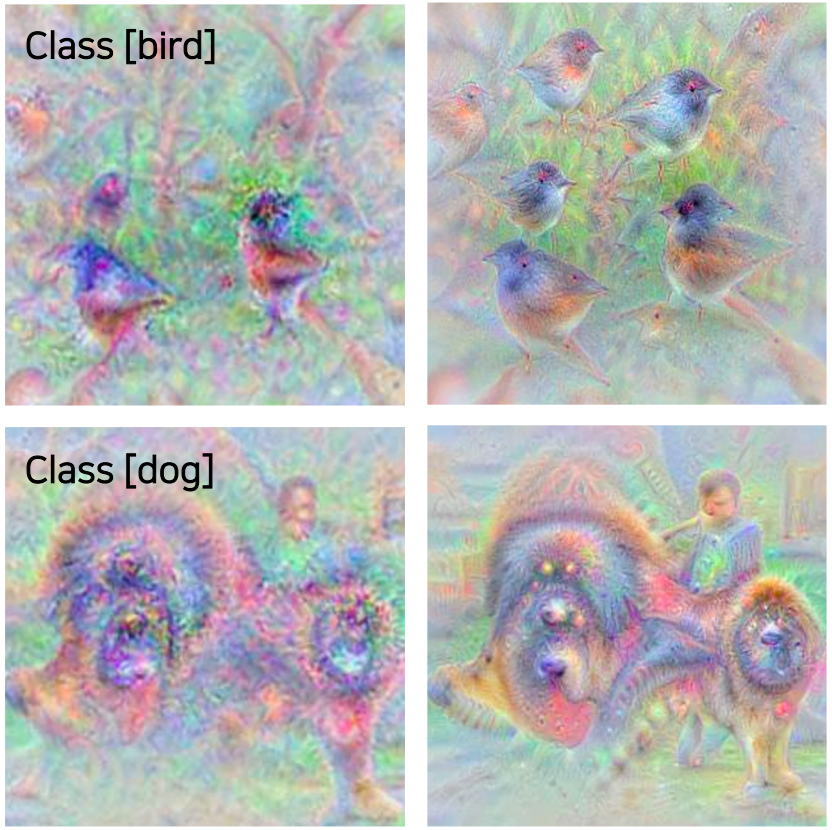
- 1) Pick a channel in a certain layer
- 2) Feed a chunk of images and record each activation map (of the chosen channel)
- 3) Crop image patches around maximum activation values

2.2 Activation investigation 3



Class visualization – Example

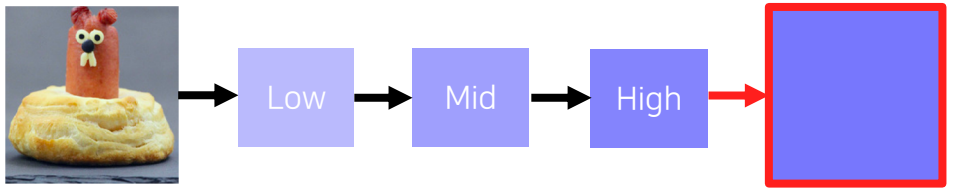
Training iteration



Visualized classes

[https://github.com/auduno/deepdraw/]

2.2 Activation investigation 3



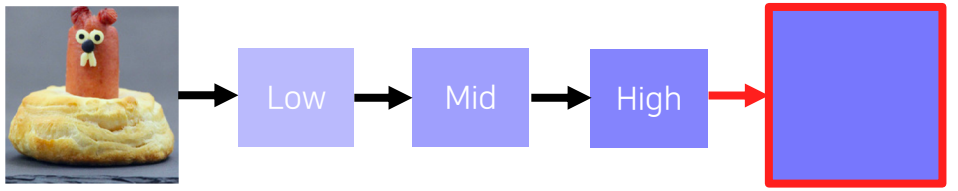
Class visualization – Gradient ascent

- Generate a synthetic image that triggers maximal class activation

$$I^* = \arg\max_I f(I) - Reg(I)$$

Regularization term

2.2 Activation investigation 3



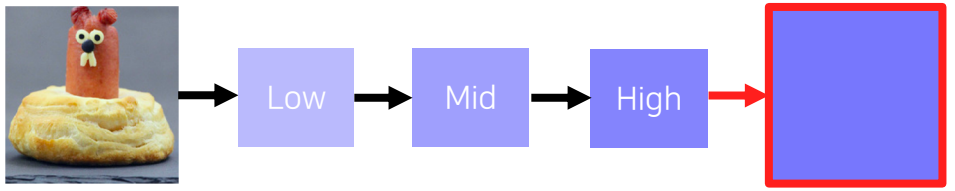
Class visualization – Gradient ascent

- Generate a synthetic image that triggers maximal class activation

$$I^* = \arg\max_I f(I) - \boxed{\lambda ||I||_2^2}$$

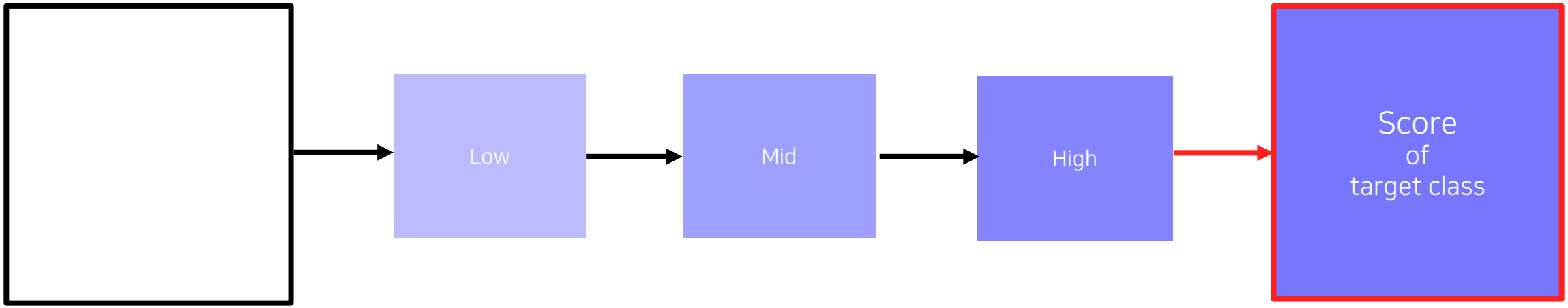
Regularization term

2.2 Activation investigation 3

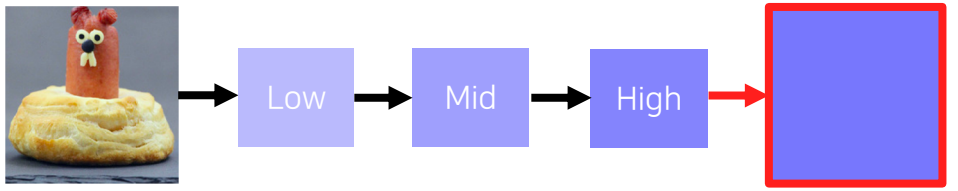


Gradient ascent – Image synthesis

- 1) Get a prediction score (of the target class) of a dummy image (blank or random initial)
- 2) Backpropagte the gradient maximizing the target class score w.r.t. the input image
- 3) Update the current image

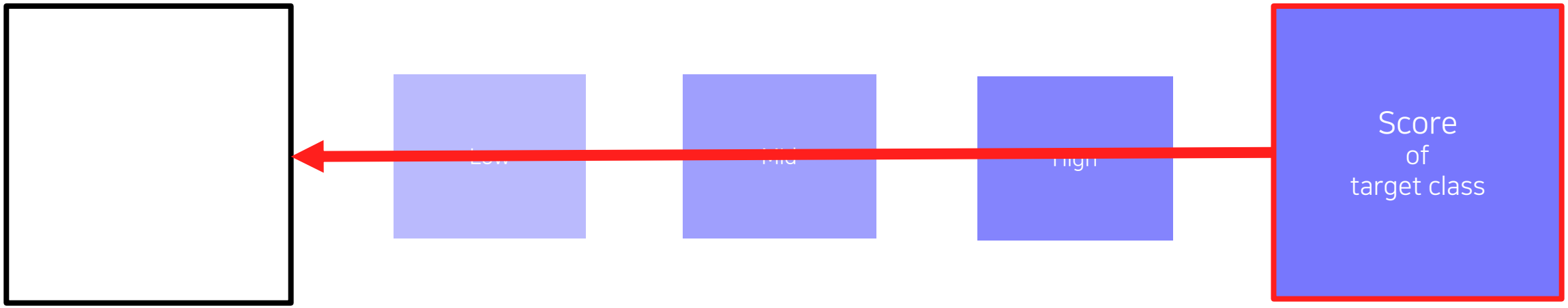


2.2 Activation investigation 3

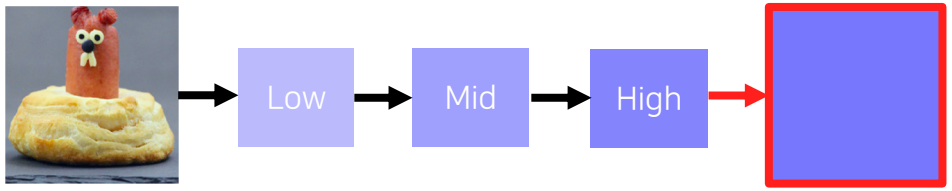


Gradient ascent – Image synthesis

- 1) Get a prediction score (of the target class) of a dummy image (blank or random initial)
- 2) Backpropagte the gradient maximizing the target class score w.r.t. the input image
- 3) Update the current image

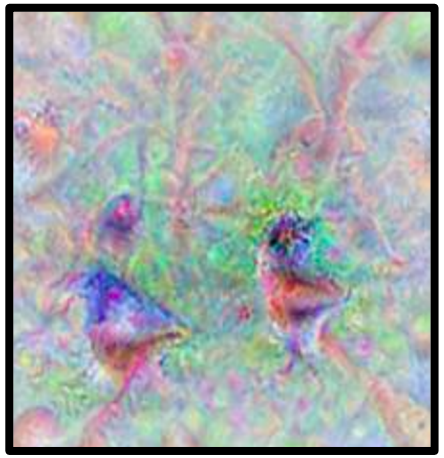


2.2 Activation investigation 3



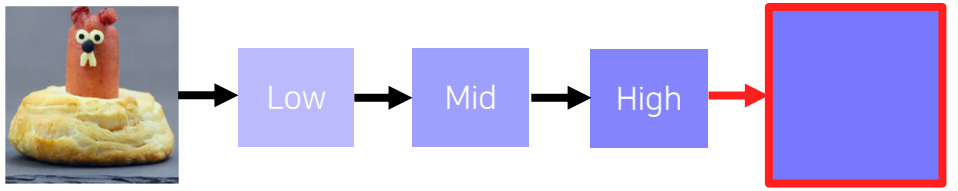
Gradient ascent – Image synthesis

- 1) Get a prediction score (of the target class) of a dummy image (blank or random initial)
- 2) Backpropagte the gradient maximizing the target class score w.r.t. the input image
- 3) Update the current image



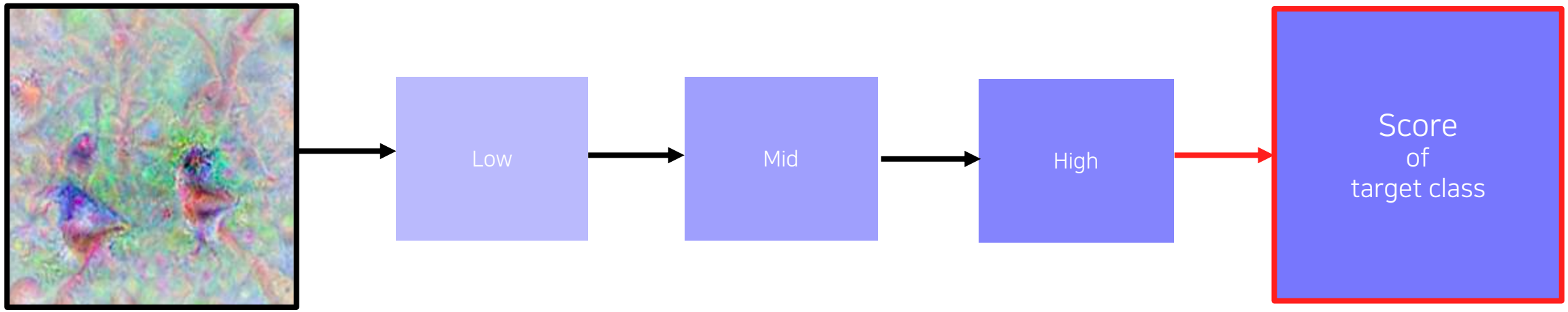
[https://github.com/auduno/deepdraw/]

2.2 Activation investigation 3



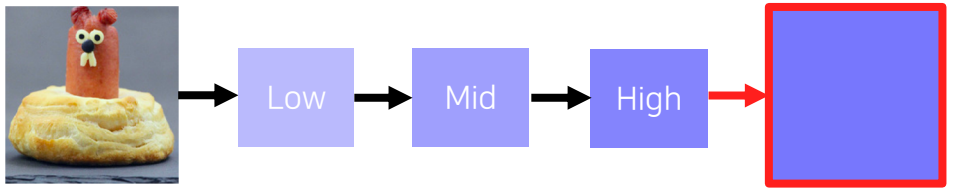
Gradient ascent – Image synthesis

- 1) Get a prediction score (of the target class) of **the current image**
- 2) Backpropagte the gradient maximizing the target class score w.r.t. the input image
- 3) Update the current image



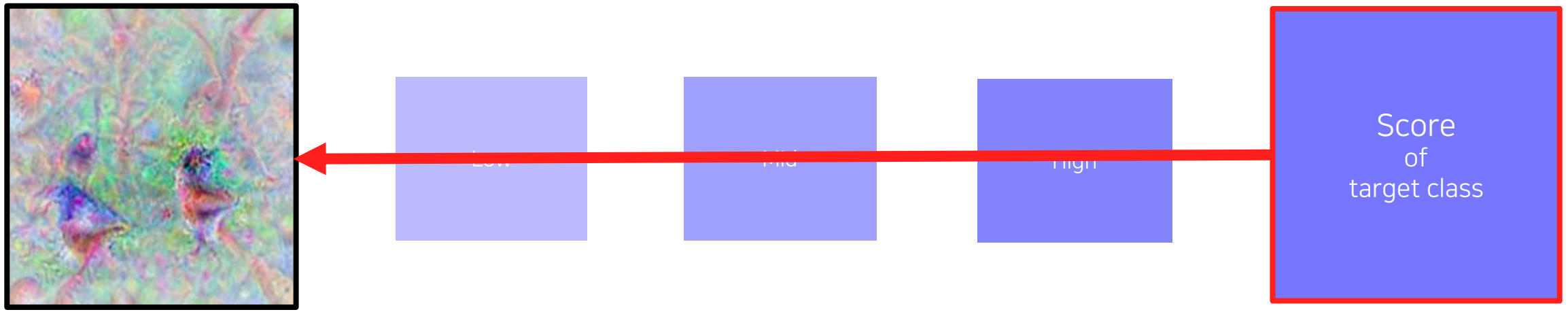
[https://github.com/auduno/deepdraw/]

2.2 Activation investigation 3



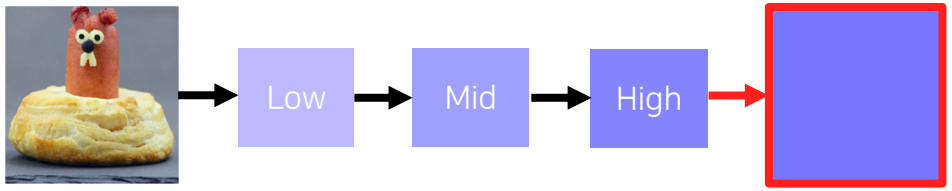
Gradient ascent – Image synthesis

- 1) Get a prediction score (of the target class) of the current image
- 2) Backpropagte the gradient maximizing the target class score w.r.t. the input image
- 3) Update the current image



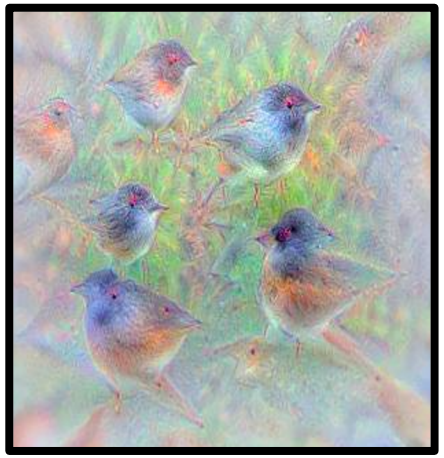
[https://github.com/auduno/deepdraw/]

2.2 Activation investigation 3



Gradient ascent – Image synthesis

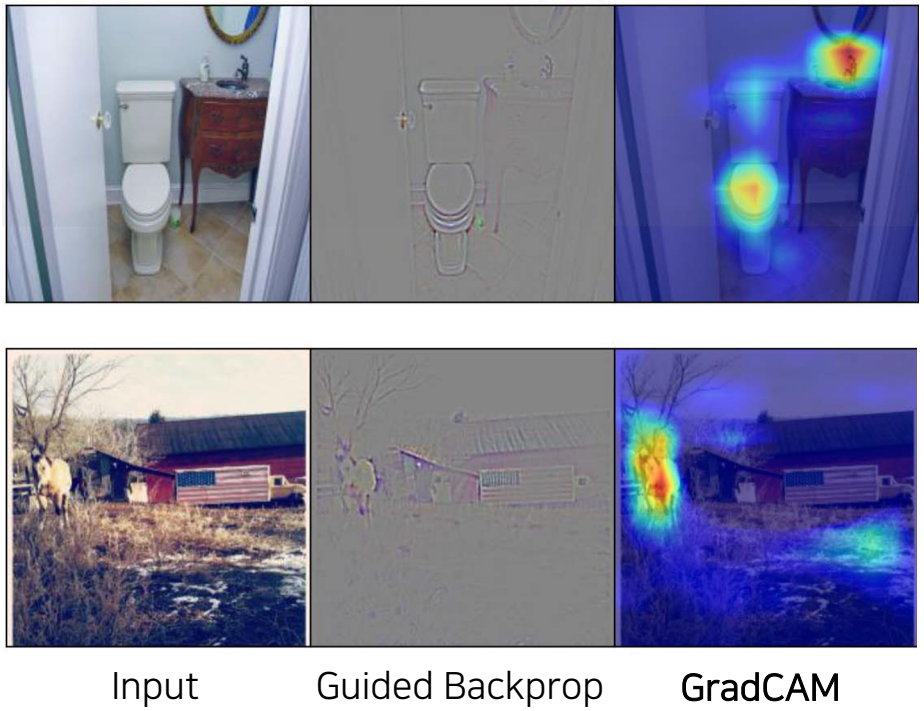
- 1) Get a prediction score (of the target class) of the current image
- 2) Backpropagte the gradient maximizing the target class score w.r.t. the input image
- 3) Update the current image



[https://github.com/auduno/deepdraw/]

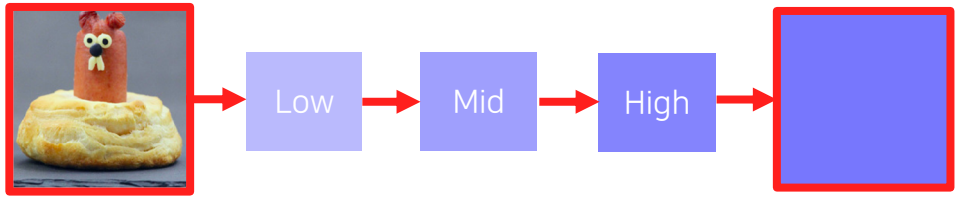
3.

Model decision explanation

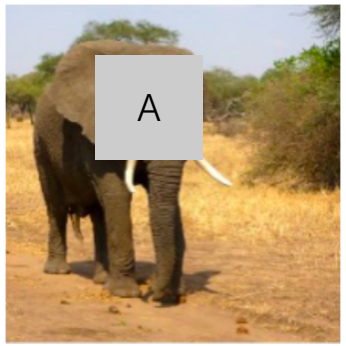


[V. Petsiuk et al., BMVC 2018]

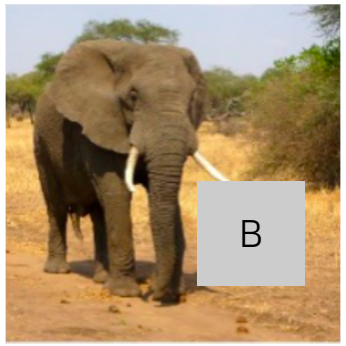
3.1 Saliency test 1



Occlusion map



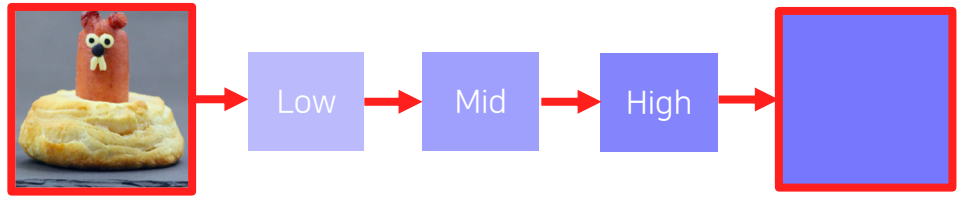
$$P(\text{Elephant} \mid \text{Occlusion A}) = 0.34$$



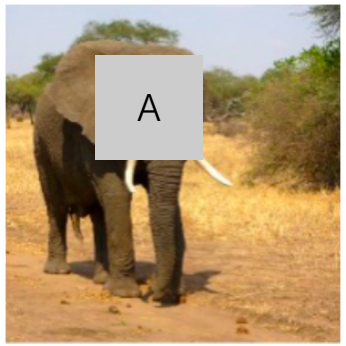
$$P(\text{Elephant} \mid \text{Occlusion B}) = 0.88$$

Prediction scores are changed according to the location of mask

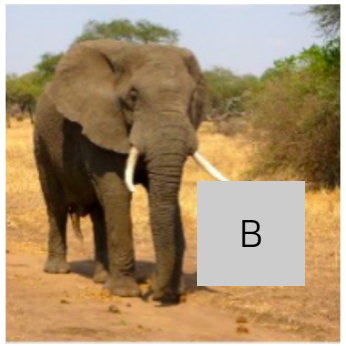
3.1 Saliency test 1



Occlusion map

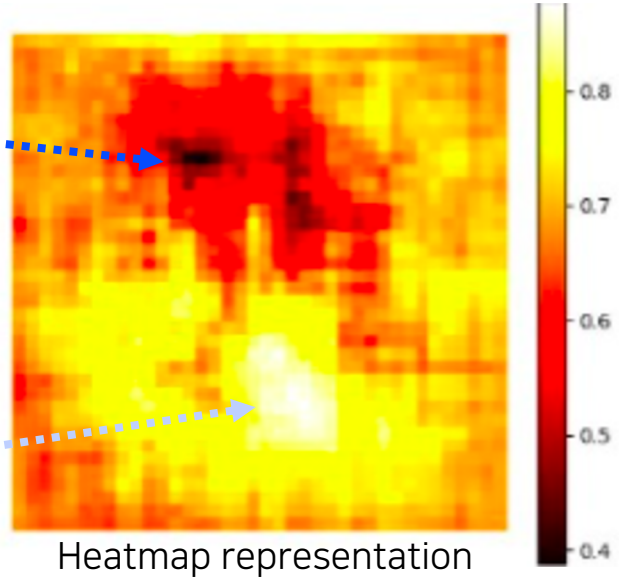


$P(\text{Elephant} \mid \text{Occlusion A}) = 0.34$



$P(\text{Elephant} \mid \text{Occlusion B}) = 0.88$

Salient parts

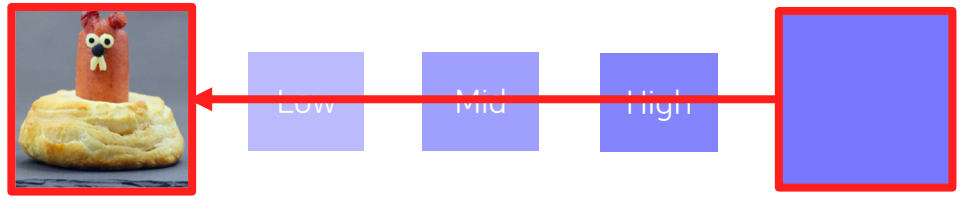


Heatmap representation

Prediction scores are changed according to the location of mask

Prediction scores drop drastically around the salient parts

3.1 Saliency test 2



via Backpropagation - Example

[Simonyan et al., CoRR 2013]

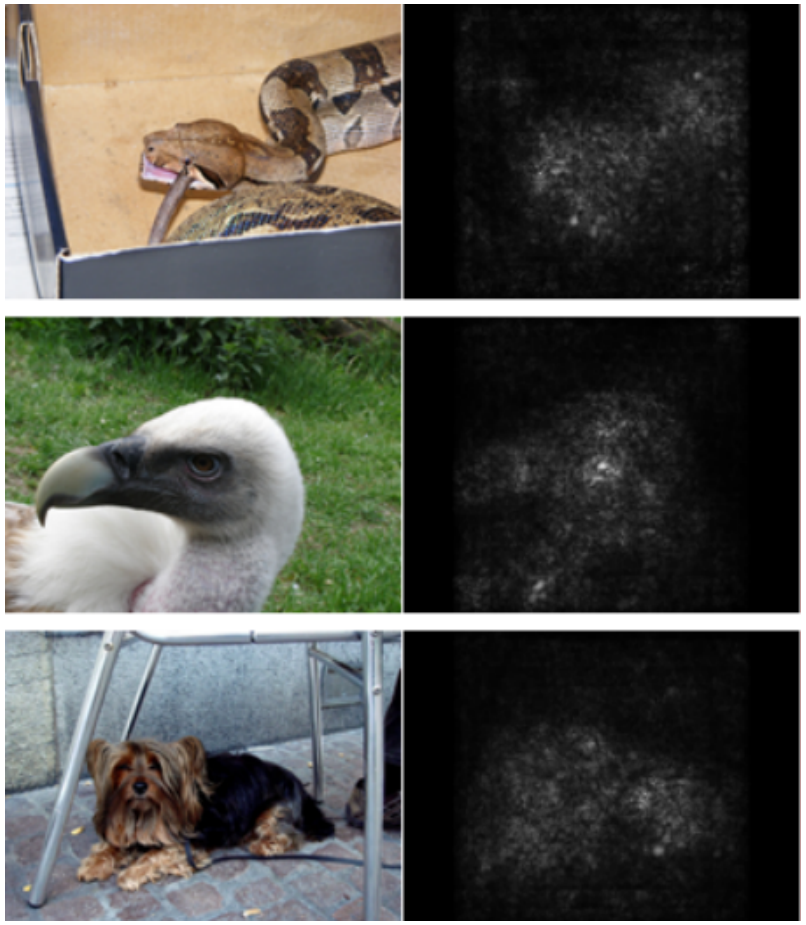
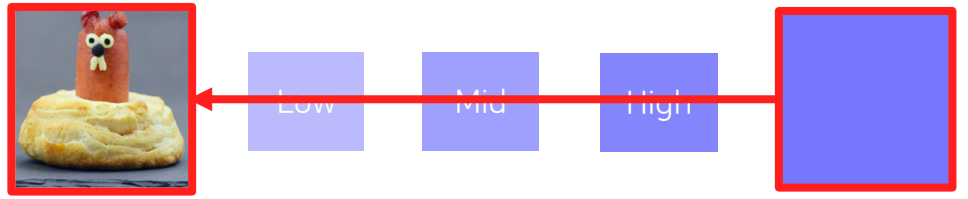


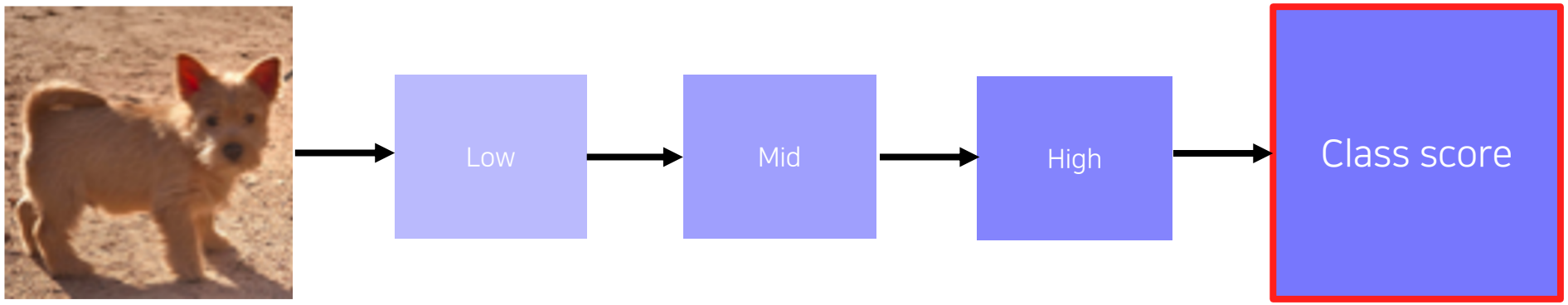
Image-specific class saliency maps

3.1 Saliency test 2

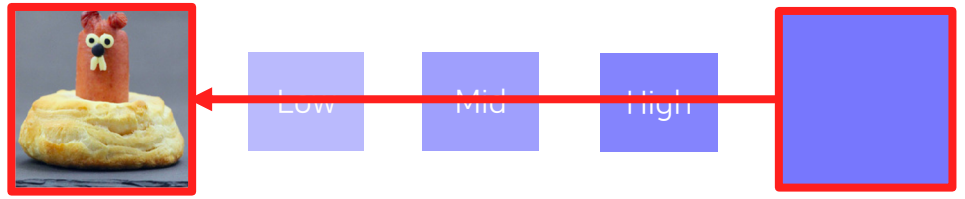


via Backpropagation - Derivatives of a class score w.r.t. input domain [Simonyan et al., CoRR 2013]

- 1) Get a class score of the target source image
- 2) Backpropagate the gradient of the class score w.r.t. input domain
- 3) Visualize the obtained gradient magnitude map (optionally, can be accumulated)

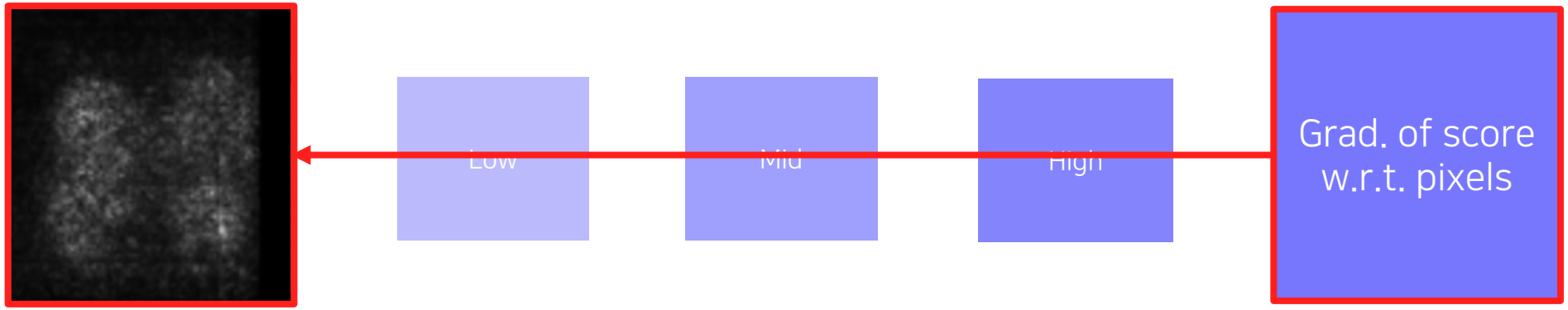


3.1 Saliency test 2

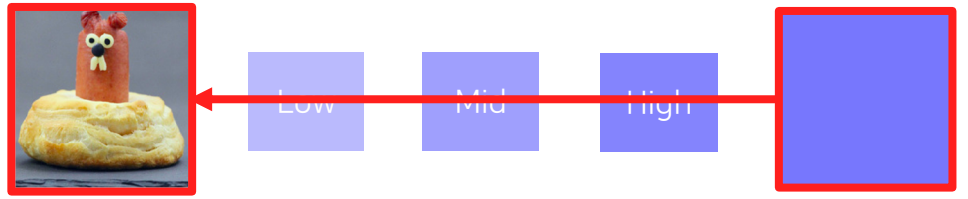


via Backpropagation - Derivatives of a class score w.r.t. input domain [Simonyan et al., CoRR 2013]

- 1) Get a class score of the target source image
- 2) Backpropagate the gradient of the class score w.r.t. input domain
- 3) Visualize the obtained gradient magnitude map (optionally, can be accumulated)

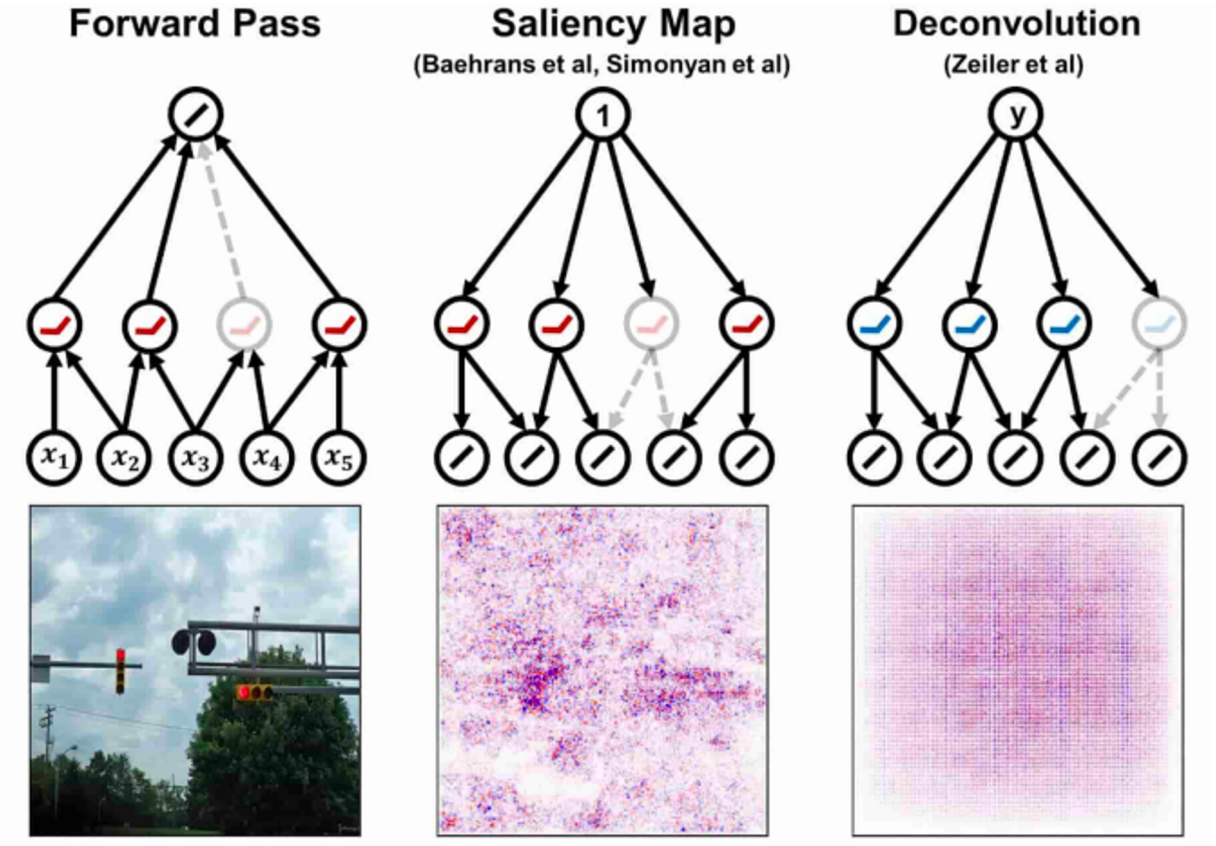


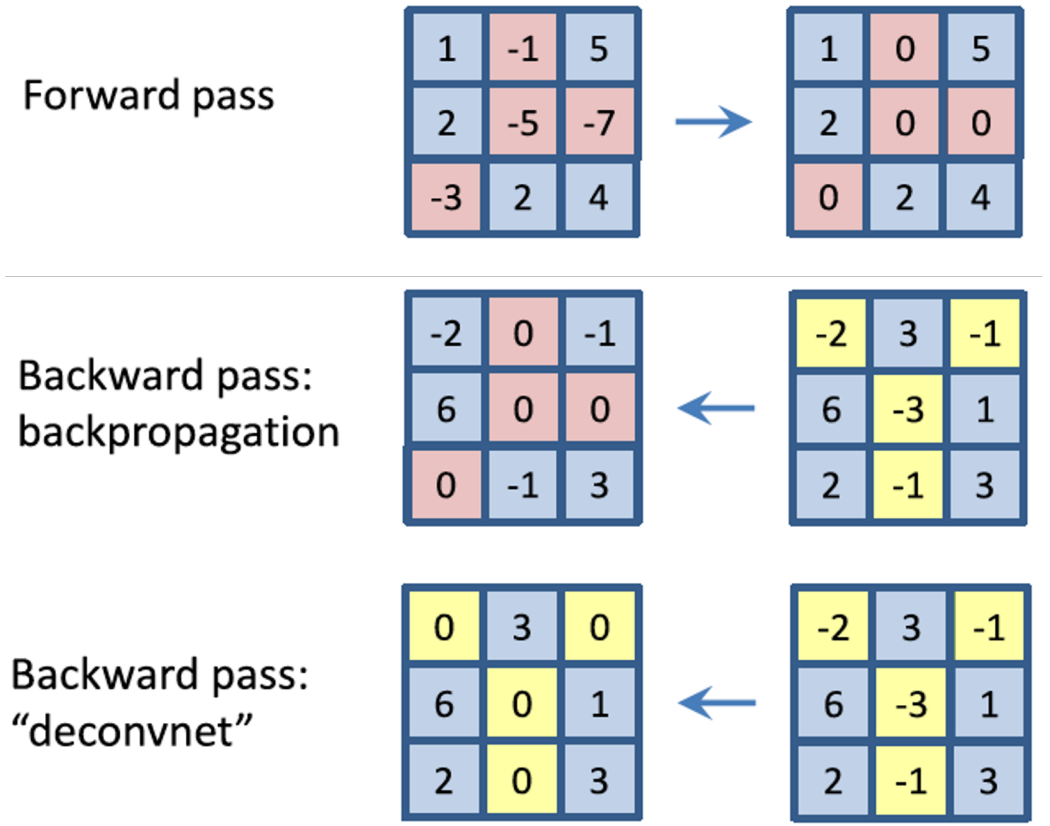
3.2 Backpropagate features



Rectified unit (backward pass)

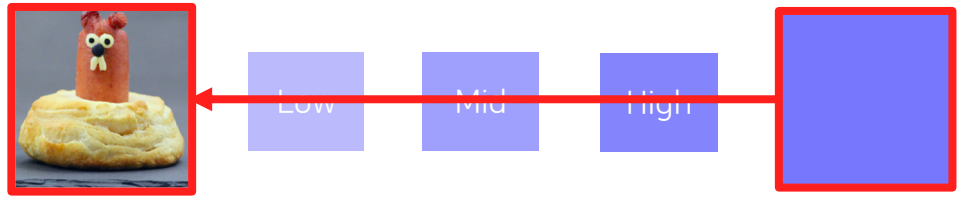
[Kim et al, ICCV 2019]





Images from [(C) Dhruv Batra and Zsolt Kira], [Kim et al, ICCV 2019]

3.2 Backpropagate features



Rectified unit (backward pass)

[Kim et al, ICCV 2019]

$$h^{l+1} = max(0, h^l)$$

$$\frac{\partial L}{\partial h^l} = [(h^l > 0)] \frac{\partial L}{\partial h^{l+1}}$$

$$\frac{\partial L}{\partial h^l} = [(h^{l+1} > 0)] \frac{\partial L}{\partial h^{l+1}}$$

Forward pass

1	-1	5
2	-5	-7
-3	2	4

→

1	0	5
2	0	0
0	2	4

Backward pass: backpropagation

-2	0	-1
6	0	0
0	-1	3

←

-2	3	-1
6	-3	1
2	-1	3

Backward pass: "deconvnet"

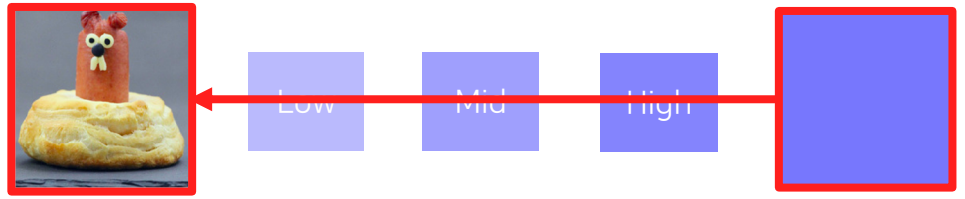
0	3	0
6	0	1
2	0	3

←

-2	3	-1
6	-3	1
2	-1	3

[(C) Dhruv Batra and Zsolt Kira]

3.2 Backpropagate features



Guided backpropagation

Backward pass:
guided
backpropagation

Forward pass

0	0	0
6	0	0
0	0	3

←

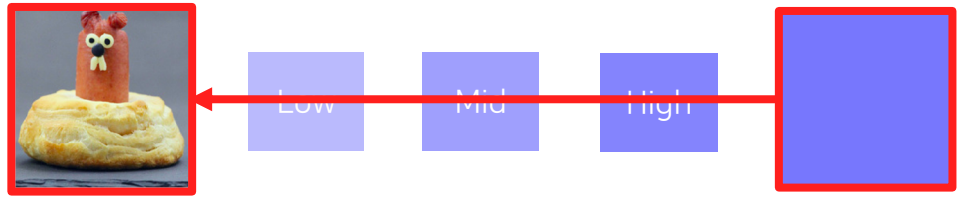
-2	3	-1
6	-3	1
2	-1	3

backpropagation

$$\frac{\partial L}{\partial h^l} = [(h^l > 0) \& (h^{l+1} > 0)] \frac{\partial L}{\partial h^{l+1}}$$

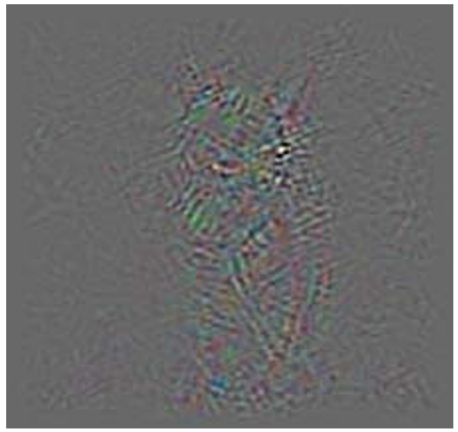
[(C) Dhruv Batra and Zsolt Kira]

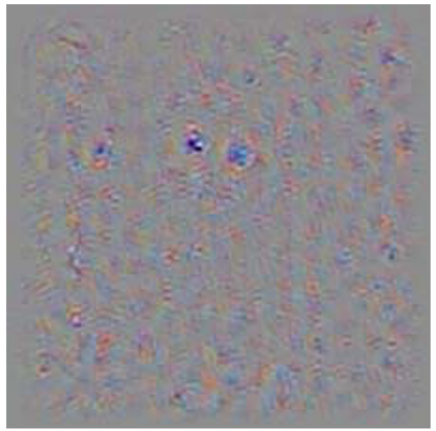
3.2 Backpropagate features



Guided backpropagation - Comparison

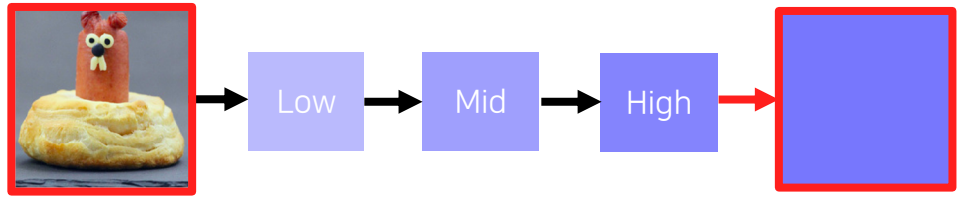

Input


Backprop


DeConv


Guided backprop

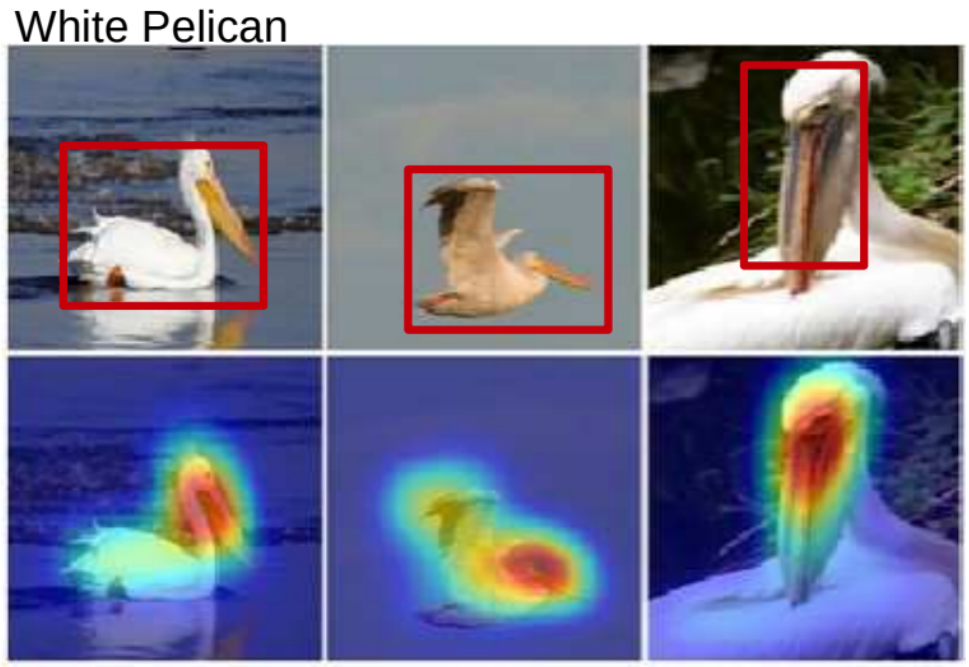
3.3 Class activation mapping

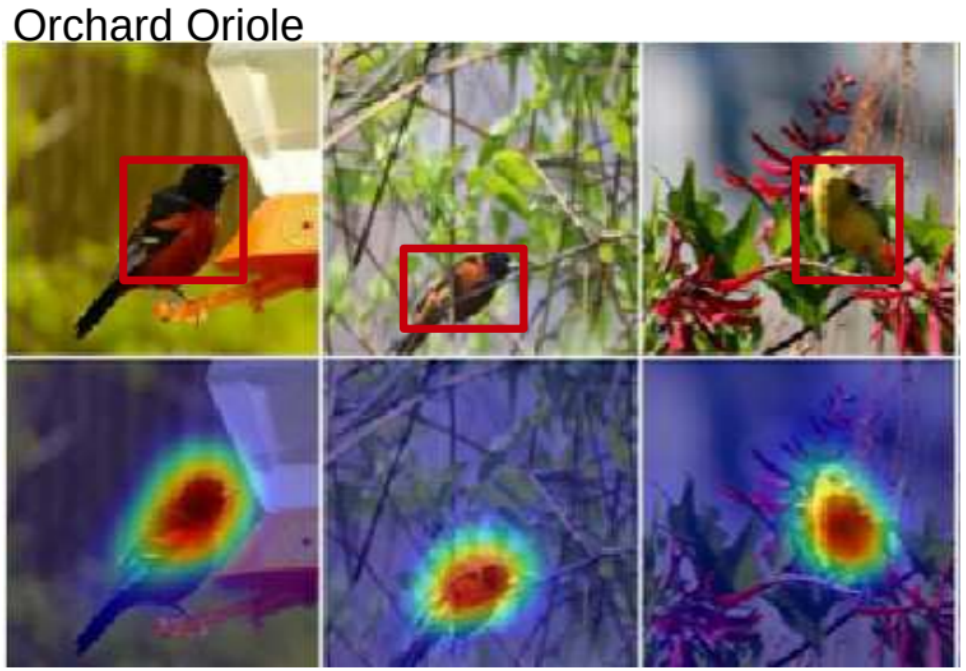


Class activation mapping (CAM) - Example

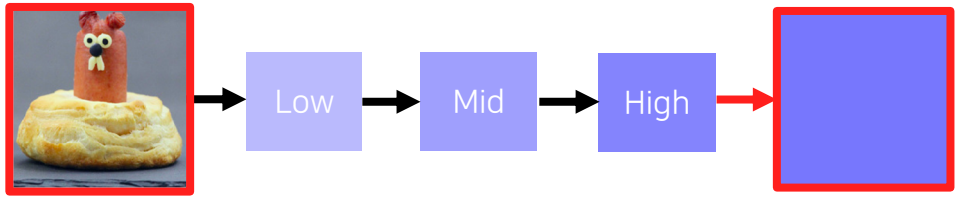
[Zhou et al., CVPR 2016]

- Visualize which part of image contributes to the final decision





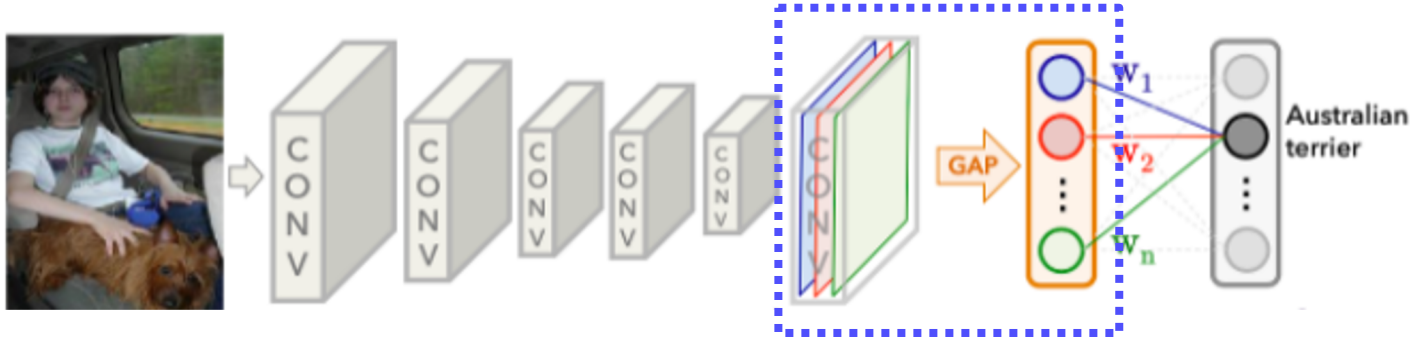
3.3 Class activation mapping



Class activation mapping (CAM)

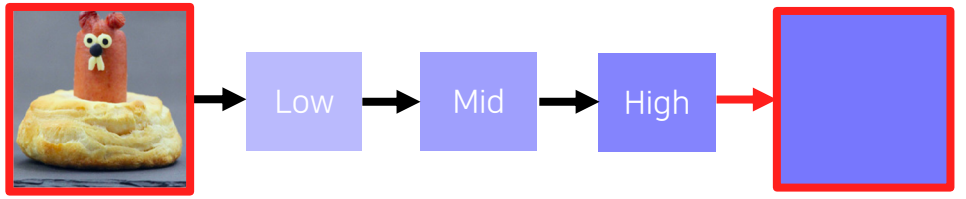
[Zhou et al., CVPR 2016]

- Global average pooling (GAP) layer instead of the FC layer



CAM architecture with GAP

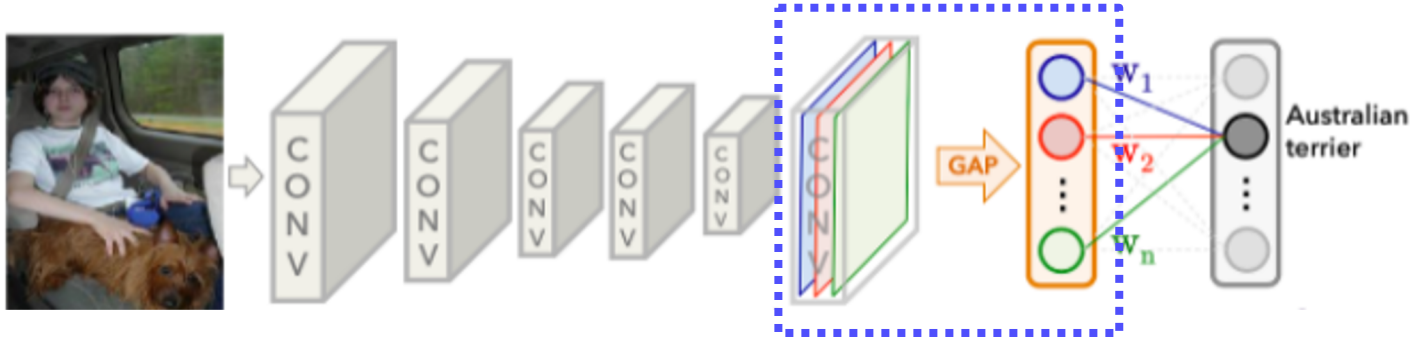
3.3 Class activation mapping



Class activation mapping (CAM)

[Zhou et al., CVPR 2016]

- Derivation of CAM: Changing the order of the operations



Channels

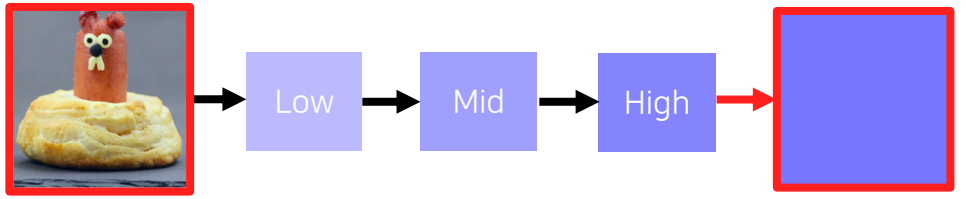
$$S_c = \sum_k w_k^c F_k$$

Score of the class c — GAP feature

$$\overset{\text{GAP}}{=} \sum_k w_k^c \sum_{(x,y)} f_k(x,y)$$

Feature map before GAP

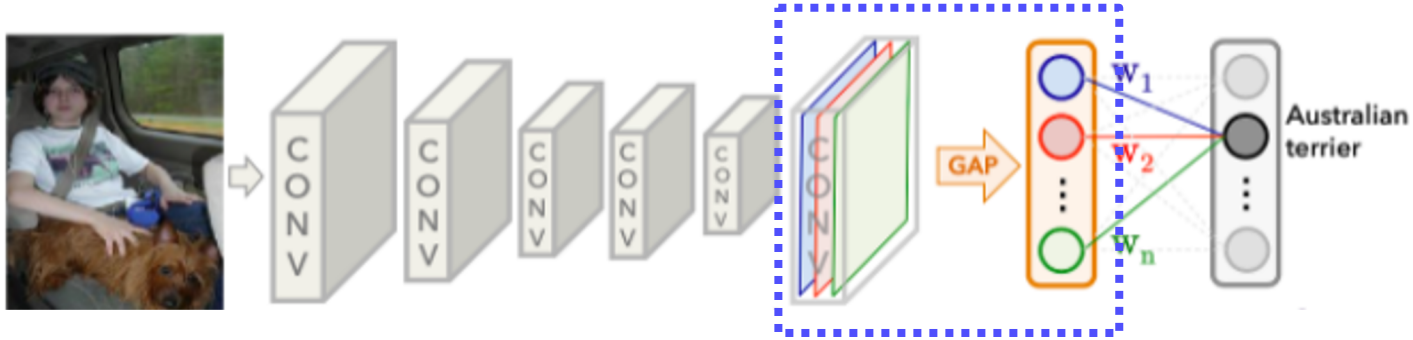
3.3 Class activation mapping



Class activation mapping (CAM)

[Zhou et al., CVPR 2016]

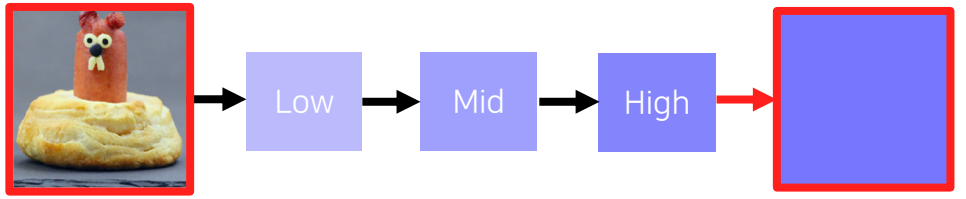
- Derivation of CAM: Changing the order of the operations



$$S_c = \sum_k w_k^c F_k$$

$$\overset{\text{GAP}}{=} \sum_k w_k^c \sum_{(x,y)} f_k(x,y) = \sum_{(x,y)} \overbrace{\sum_k w_k^c f_k(x,y)}^{CAM_c(x,y)}$$

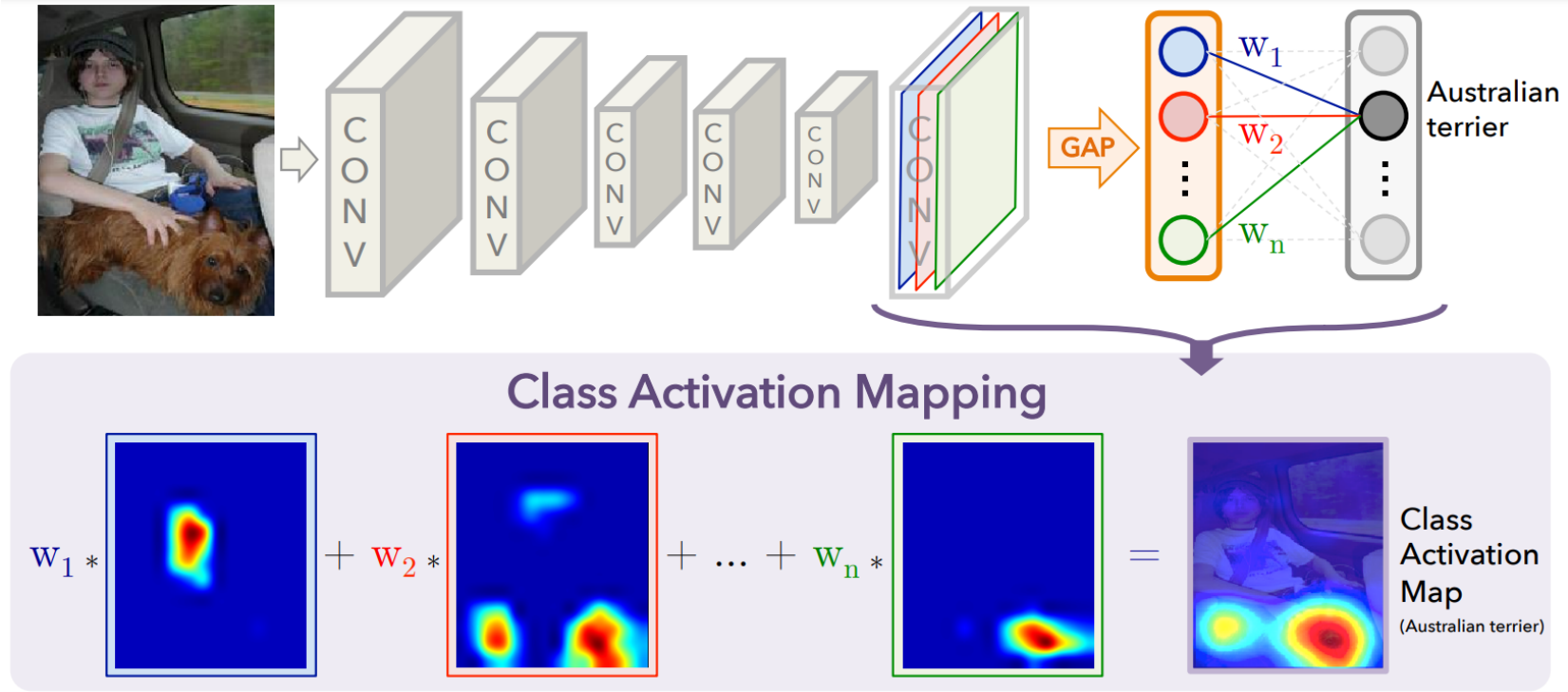
3.3 Class activation mapping



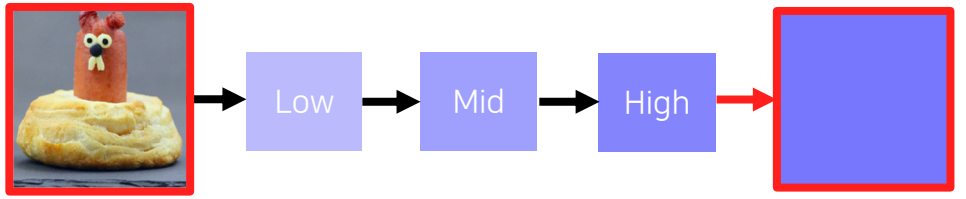
Class activation mapping (CAM)

[Zhou et al., CVPR 2016]

- By visualizing CAM, we can interpret why the network classified the input to that class
- GAP layer enables localization without supervision



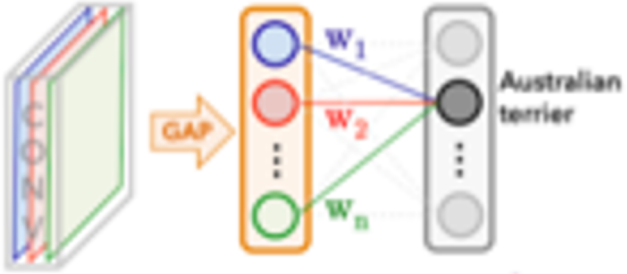
3.3 Class activation mapping



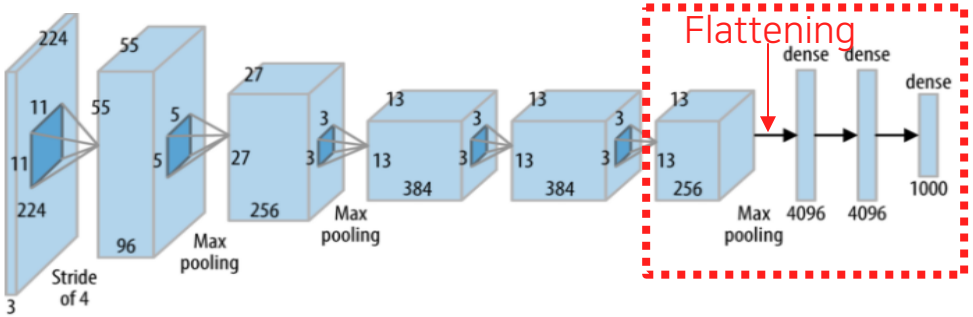
Class activation mapping (CAM)

[Zhou et al., CVPR 2016]

- Requires a modification of the network architecture and re-training
- ResNet and GoogLeNet already have the GAP layer

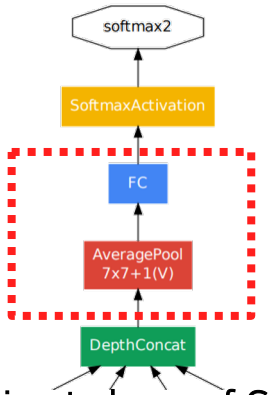


CAM



AlexNet architecture

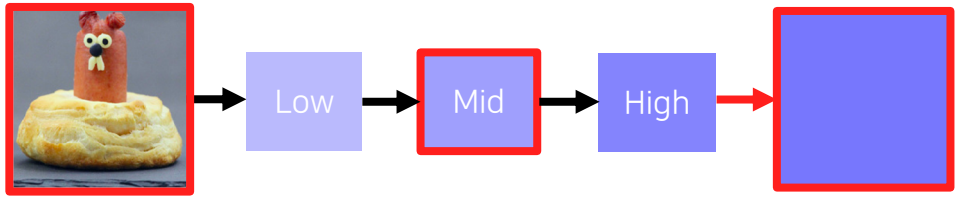
To be modified to CAM, the flattening layer should be changed to GAP



The penultimate layer of GoogLeNet

[Szegedy et al., CVPR 2014; https://aqeel-anwar.medium.com/]

3.3 Class activation mapping



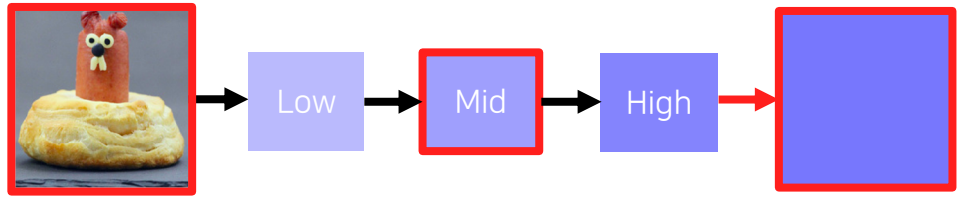
Grad-CAM - Example

[Selvaraju et al., ICCV 2017]

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

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

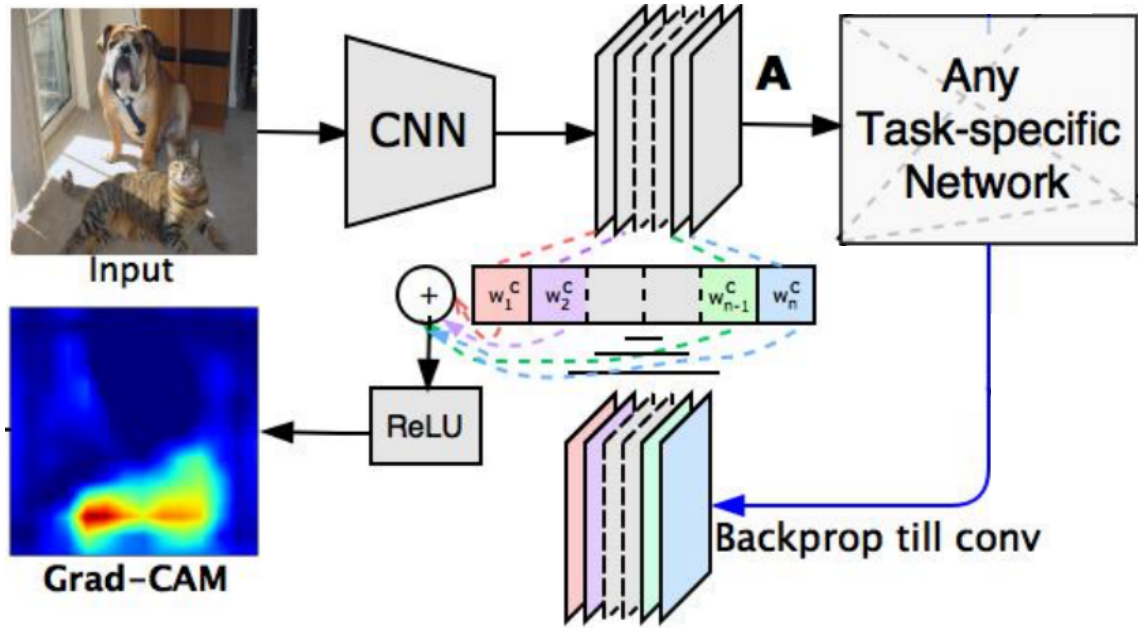
3.3 Class activation mapping



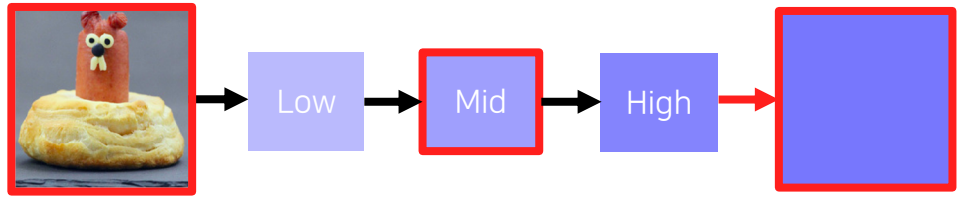
Grad-CAM

[Selvaraju et al., ICCV 2017]

- Get the CAM result without modifying and re-training the original network



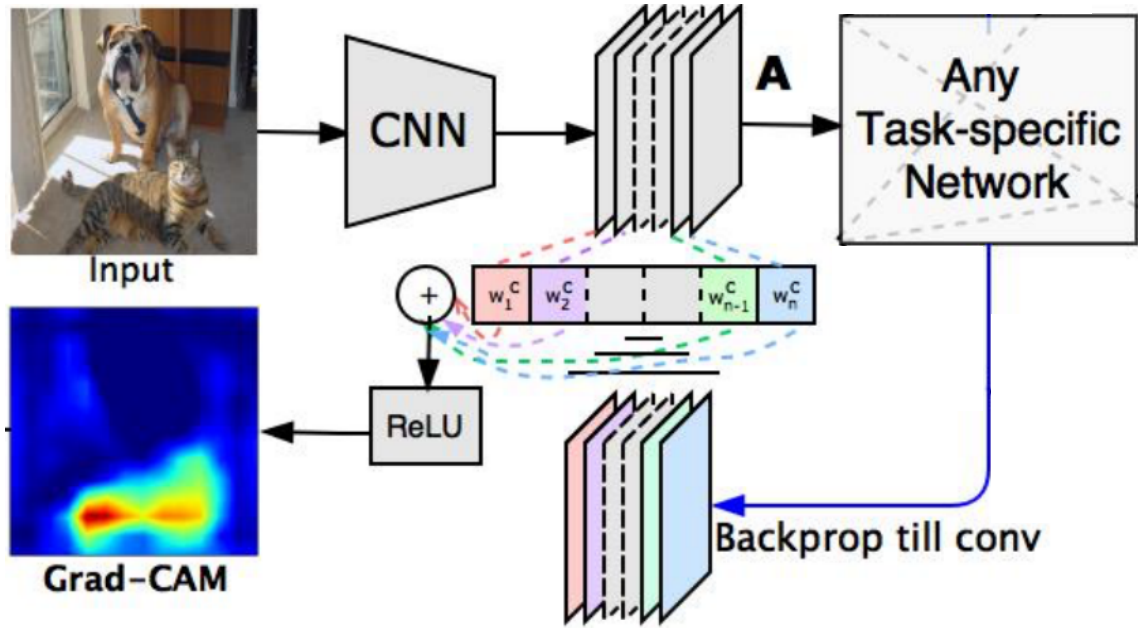
3.3 Class activation mapping



Grad-CAM

[Selvaraju et al., ICCV 2017]

- Get the CAM result without modifying and re-training the original network

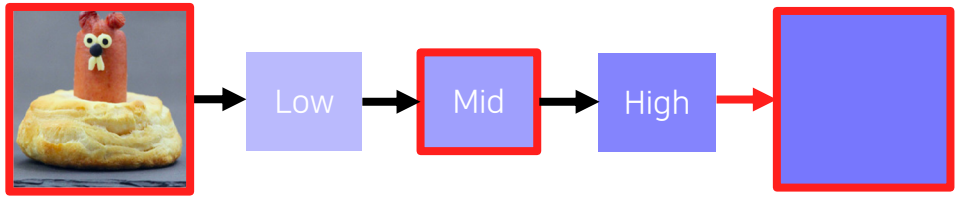


$$CAM_c(x,y)$$

$$\sum_k w_k^c f_k(x,y)$$

Key idea:
How to obtain the importance weights

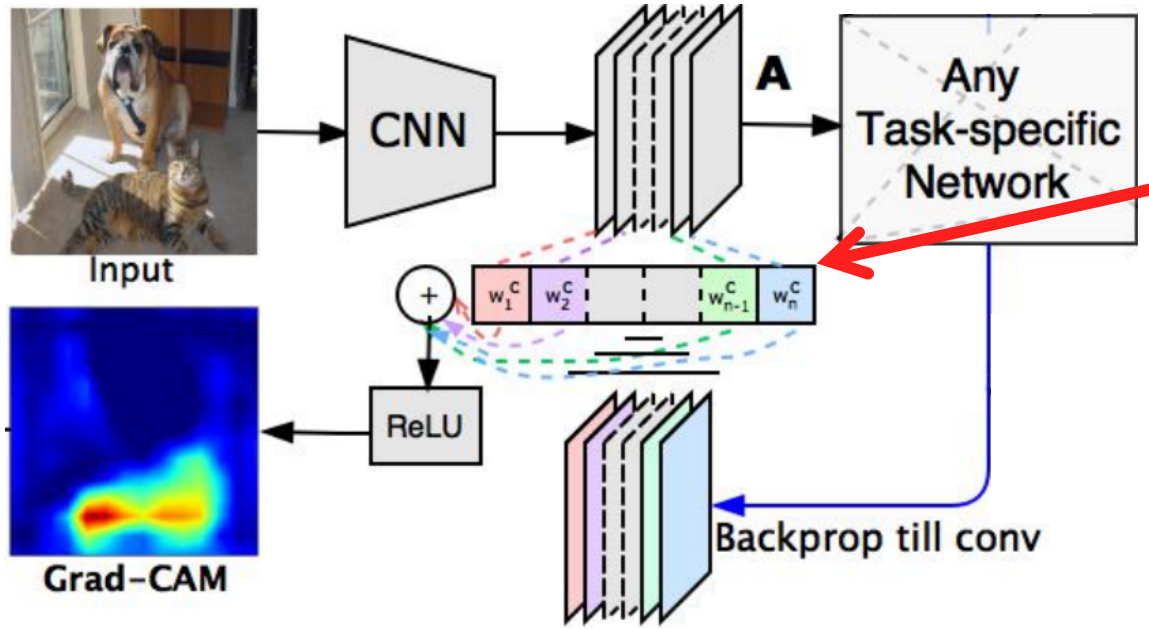
3.3 Class activation mapping



Grad-CAM

[Selvaraju et al., ICCV 2017]

- Measure magnitudes of gradients as neuron importance weights



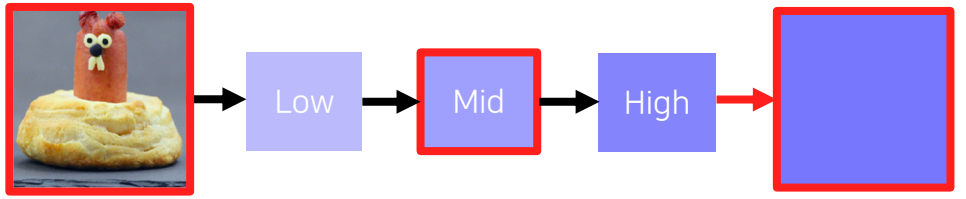
Global average pooling

$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k}$$

Importance weight of the k-th feature map w.r.t. the class c

Gradients via backprop

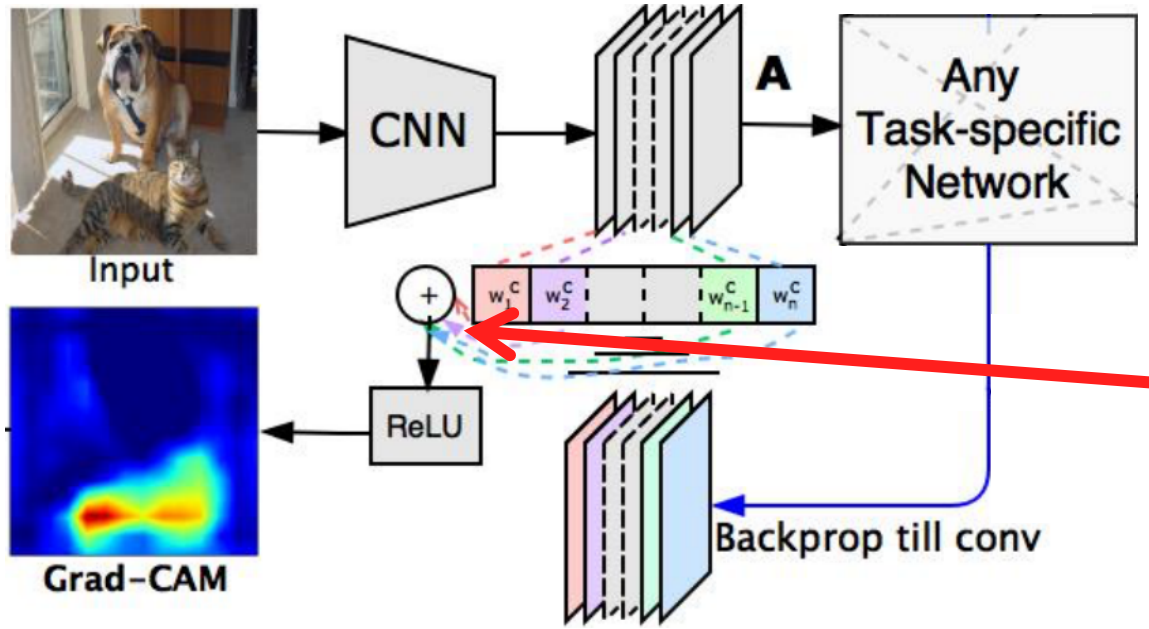
3.3 Class activation mapping



Grad-CAM

[Selvaraju et al., ICCV 2017]

- With ReLU, we focus on the positive effect only

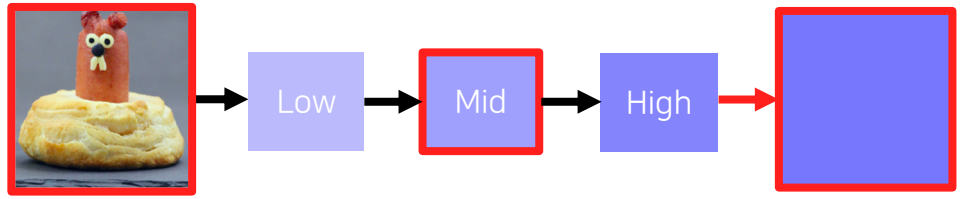


$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k}$$

Linear combination

$$L_{Grad-CAM}^c = ReLU\left(\sum_k \alpha_k^c A^k\right)$$

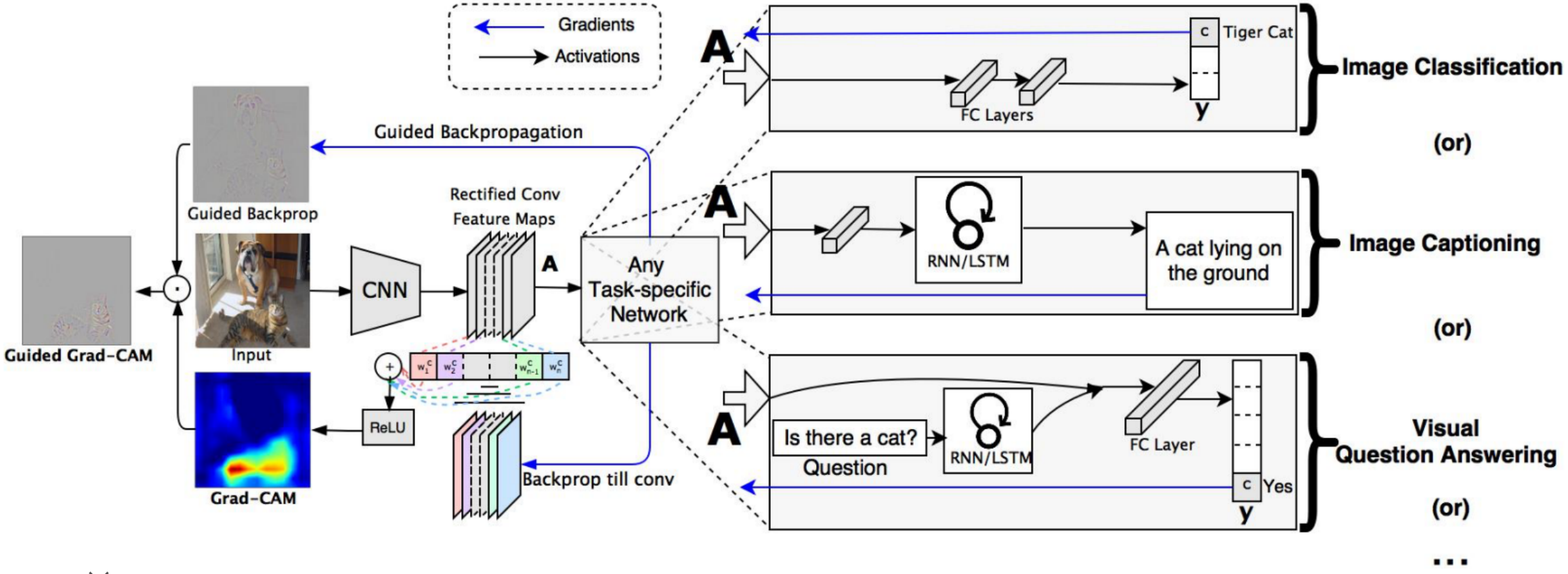
3.3 Class activation mapping



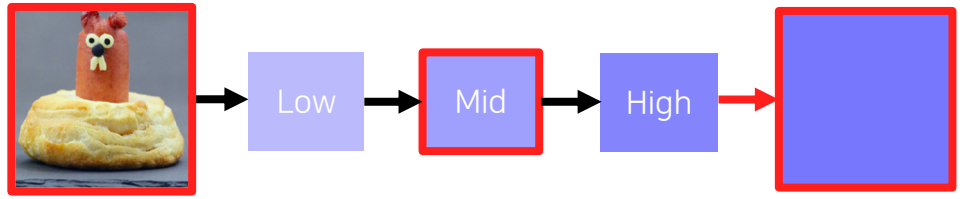
Grad-CAM

[Selvaraju et al., ICCV 2017]

- Grad-CAM works with any task head



3.3 Class activation mapping

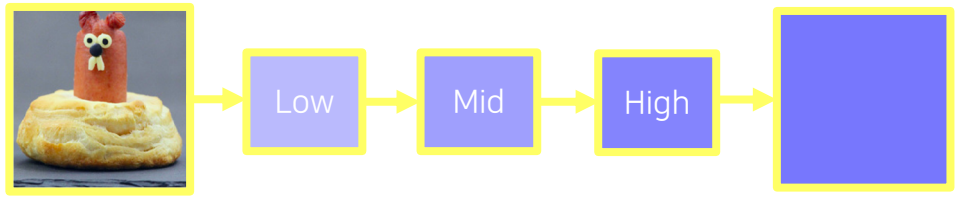


SCOUTER - Example

[Li et al., arXiv 2020]

Scouter tells "why the image is of a certain category" or "why the image is not of a certain category."

Conclusion.



GAN dissection

[Bau et al., ICLR 2019]

- Spontaneously emerging interpretable representation during training
- Not only for analysis but also for manipulation applications

Inserting door with GAN

Reference

1. Visualizing CNN

- Zeiler and Fergus, Visualizing and Understanding Convolutional Networks, ECCV 2014

2. Analysis on model behavior

- Krizhevsky et al., ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012
- Maaten and Hinton, Visualizing Data using t-SNE, JMLR 2008
- Bau et al., Network Dissection: Quantifying Interpretability of Deep Visual Representations, CVPR 2017
- Springenberg et al., Striving for Simplicity: The All Convolutional Net, ICLR 2015
- Goodfellow et al., Explaining and Harnessing Adversarial Examples, ICLR 2015
- Szegedy et al., Intriguing properties of neural networks, CVPR 2014

3. Model decision explanation

- Simonyan et al., Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps, CoRR 2013
- Li at el., Instance-Level Salient Object Segmentation, CVPR 2017
- Kim et al., Why are Saliency Maps Noisy? Cause of and Solution to Noisy Saliency Maps, ICCV 2019
- Zhou et al., Learning Deep Features for Discriminative Localization, CVPR 2016
- Selvaraju et al., Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization, ICCV 2017
- Bau et al., GAN DISSECTION: VISUALIZING AND UNDERSTANDING GENERATIVE ADVERSARIAL NETWORKS, ICLR 2019
- Li et al., SCOUTER: Slot Attention-based Classifier for Explainable Image Recognition, arXiv 2020

End of Document
Thank You.

상위 카테고리 입력란