CS5242 : Neural Networks and Deep Learning

Lecture
Visualizing Convolutional Neural Networks

Semester 1 2021/22

Xavier Bresson

https://twitter.com/xbresson

Department of Computer Science
National University of Singapore (NUS)

Outline

- ConvNet features
- Domain of dependence of features
- Visualizing learned filters
- Hierarchical filters
- Improving ConvNets with visualization
- Conclusion

Outline

- **ConvNet features**
- Domain of dependence of features
- Visualizing learned filters
- Hierarchical filters
- Improving ConvNets with visualization
- Conclusion

ConvNet features

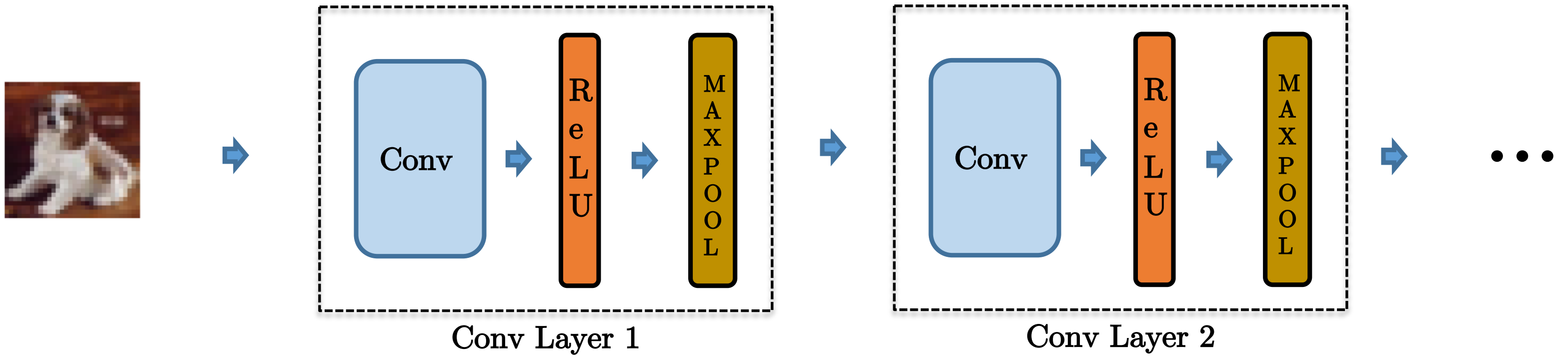
- ConvNets are extremely efficient at extracting compositional statistic patterns in large-scale and high-dimensional image datasets.
- Can we visualize the features/patterns learned that perform the classification task?
  - Some filters may have some human interpretations.
  - But most of them may not.
- However, visualizing some features can help to understand what ConvNet is doing.

Outline

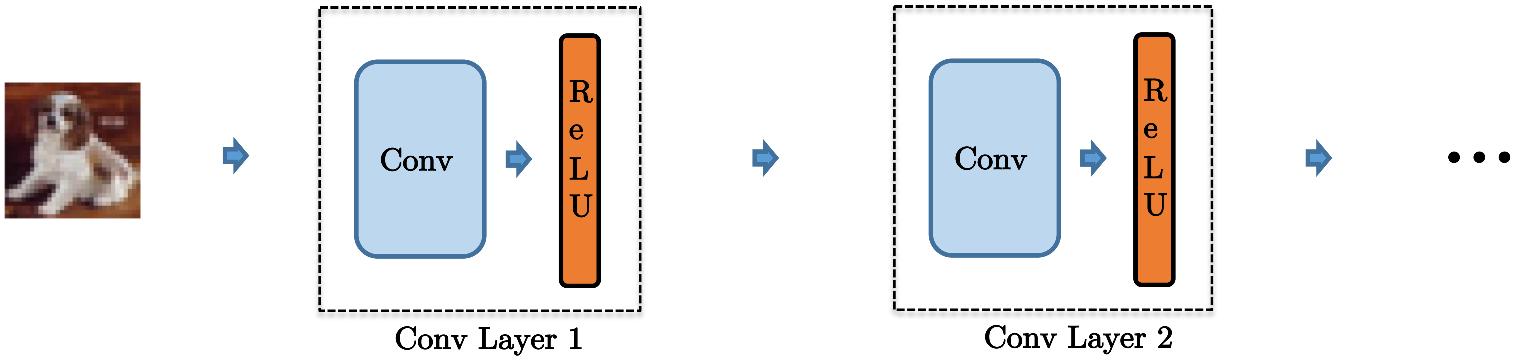
- ConvNet features
- **Domain of dependence of features**
- Visualizing learned filters
- Hierarchical filters
- Improving ConvNets with visualization
- Conclusion

Domain of dependence

- Consider a L-layer ConvNet and 3x3 filters :

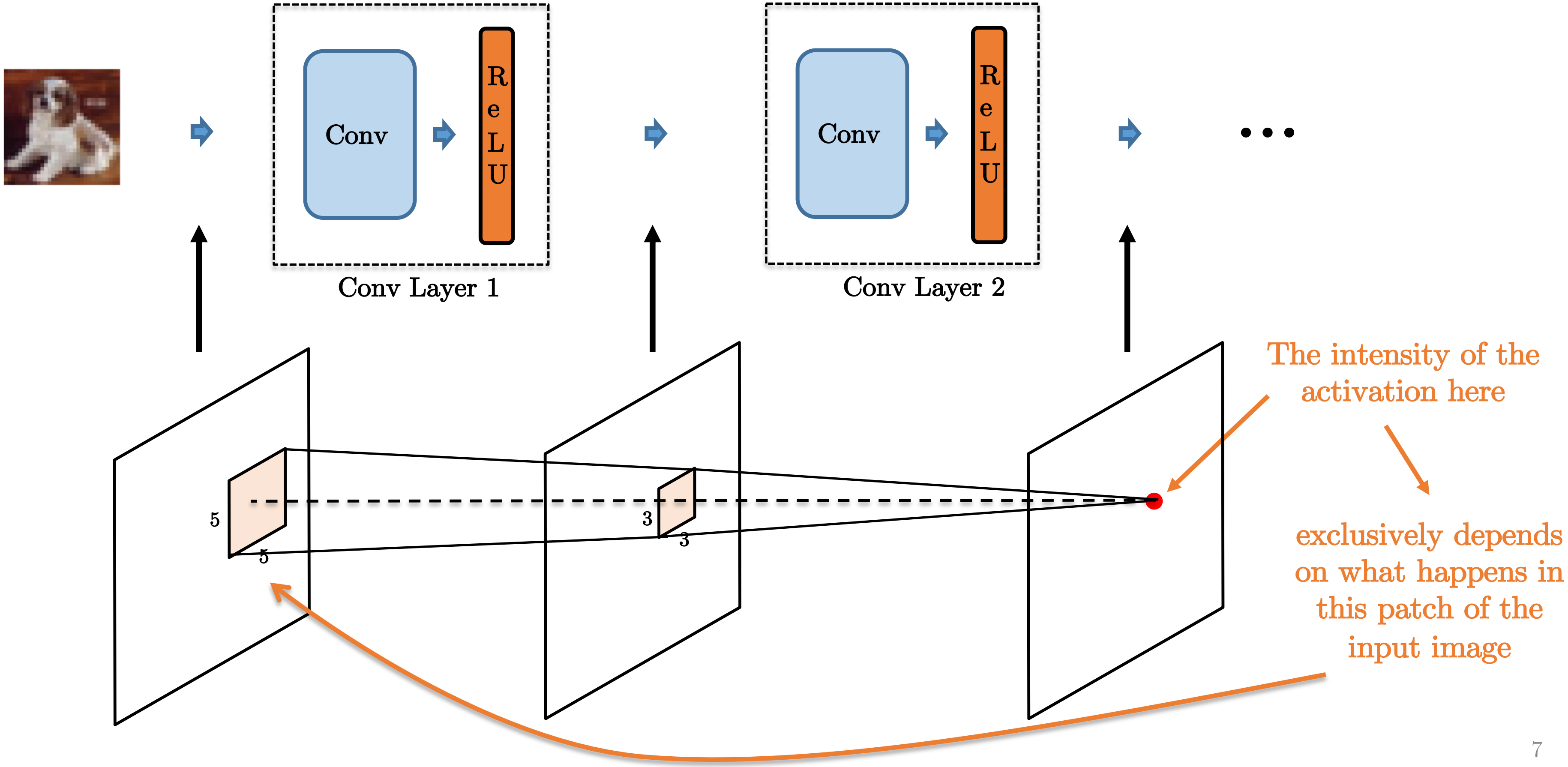


- For simplicity, suppose we do not use the Max Pooling layer :

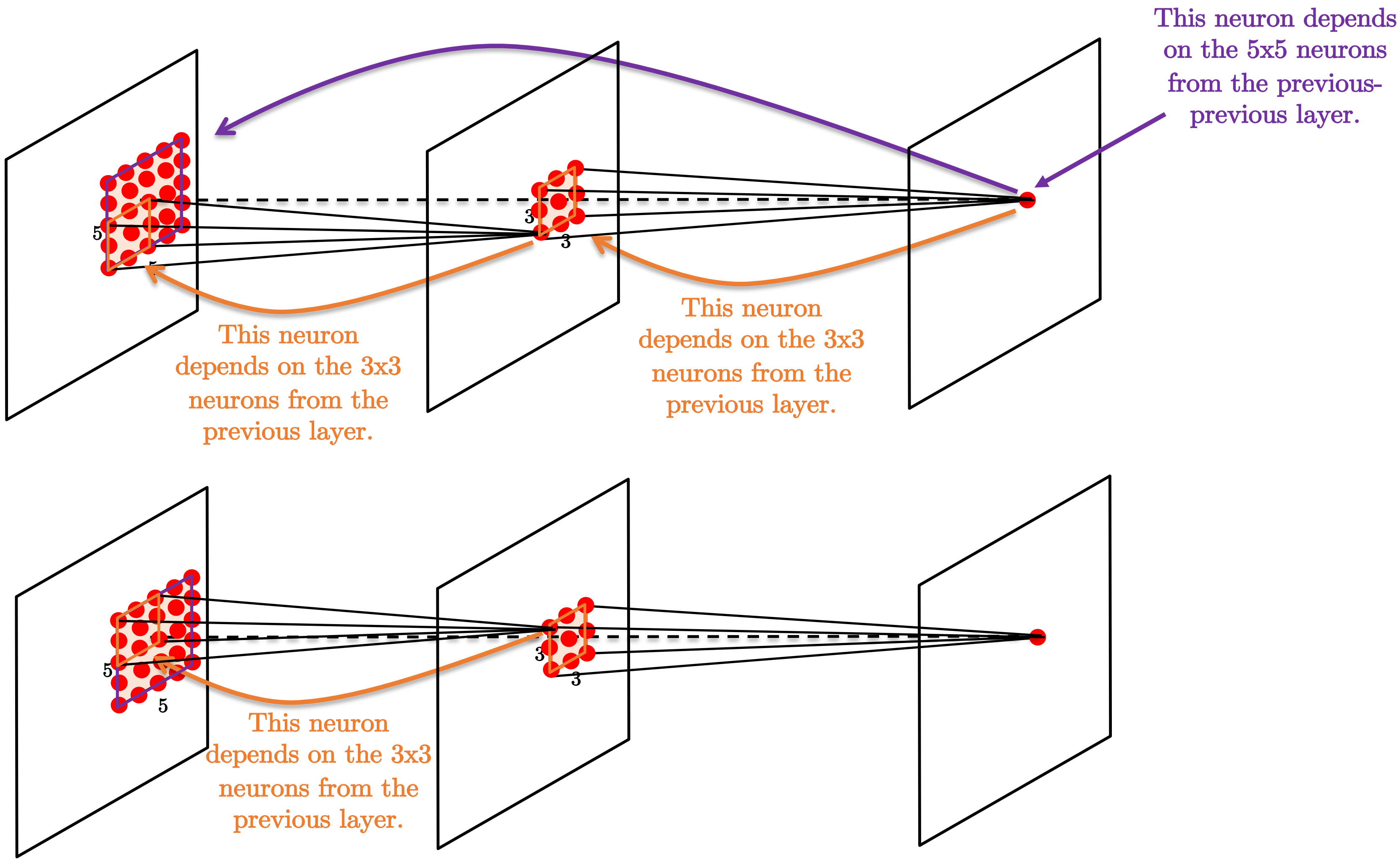


Domain of dependence

- Consider a L-layer ConvNet and 3x3 filters :



Domain of dependence



Domain of dependence

- When considering 3x3 convolutional filters :
  - A neuron from the 1$^{\text{st}}$ layer has access to 3 x 3 patches of the input image.
  - A neuron from the 2$^{\text{nd}}$ layer has access to 5 x 5 patches of the input image.
  - A neuron from the 3$^{\text{rd}}$ layer has access to 7 x 7 patches of the input image.
  - ...

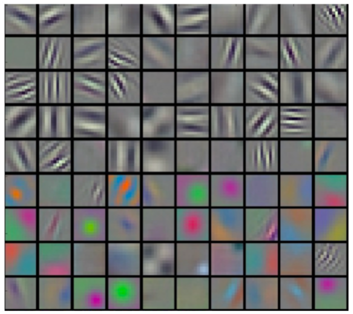
Outline

Visualizing filters learned with ConvNets

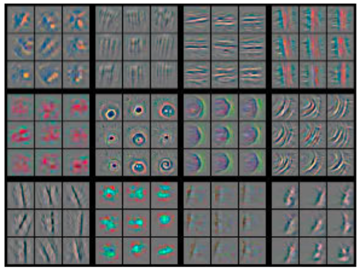
- Can we visualize the filters/patterns learned in the 1^{st} layer, 2^{nd} layer, 3^{rd} layer, etc ?
- Intuition: Hierarchical filters/features learned in the layers :
  - First layers learn low-level features like edges, corners, small textures.
  - Intermediate layers learn medium-level features like eyes, noise, etc.
  - Deep layers learn high-level features like faces, cars, cats, etc.
- Can we check this intuition?

DeConvNets

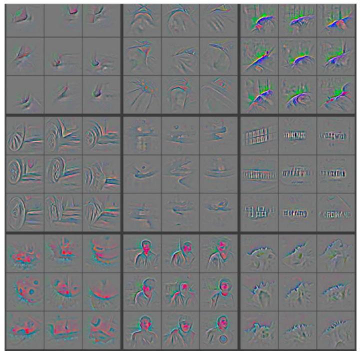
- Deconvolutional neural networks introduced by Zeiler and Fergus, 2013 :
  - Technique to map activations from deep layers back to input.
  - DeConvNets are like ConvNets, but in reverse order.
  - Can be used for visualizing filters learned with ConvNets.
  - It is not a learning technique.



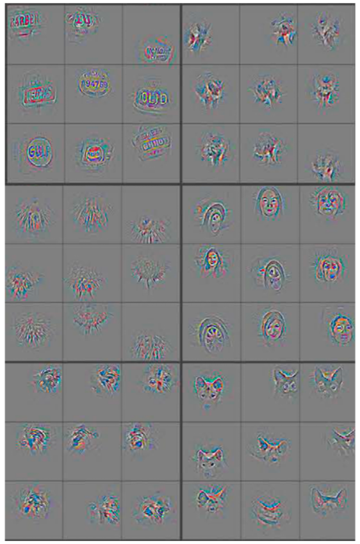
Filters learned 1^{st} layer



Filters learned 2^{nd} layer

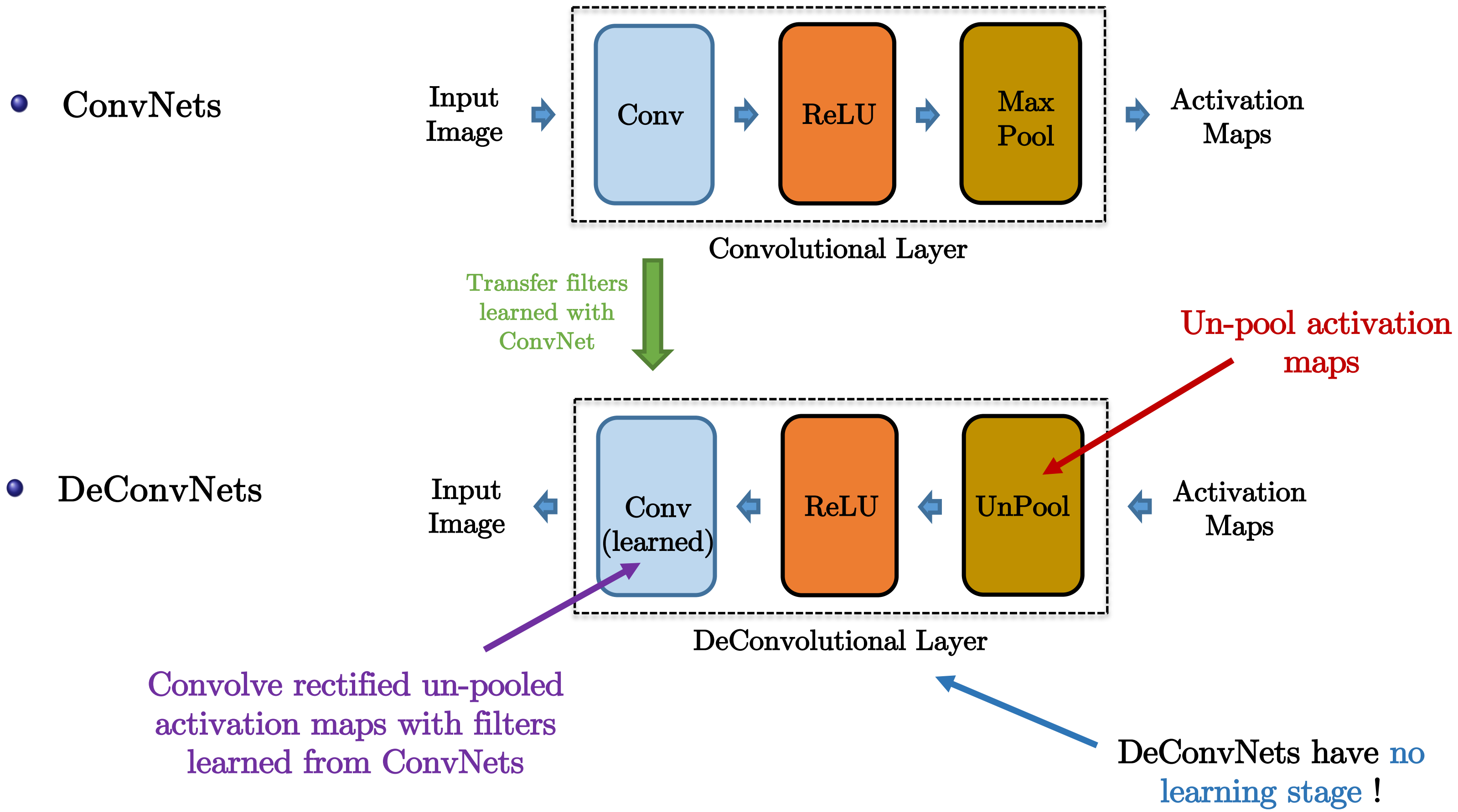


Filters learned 3^{rd} layer



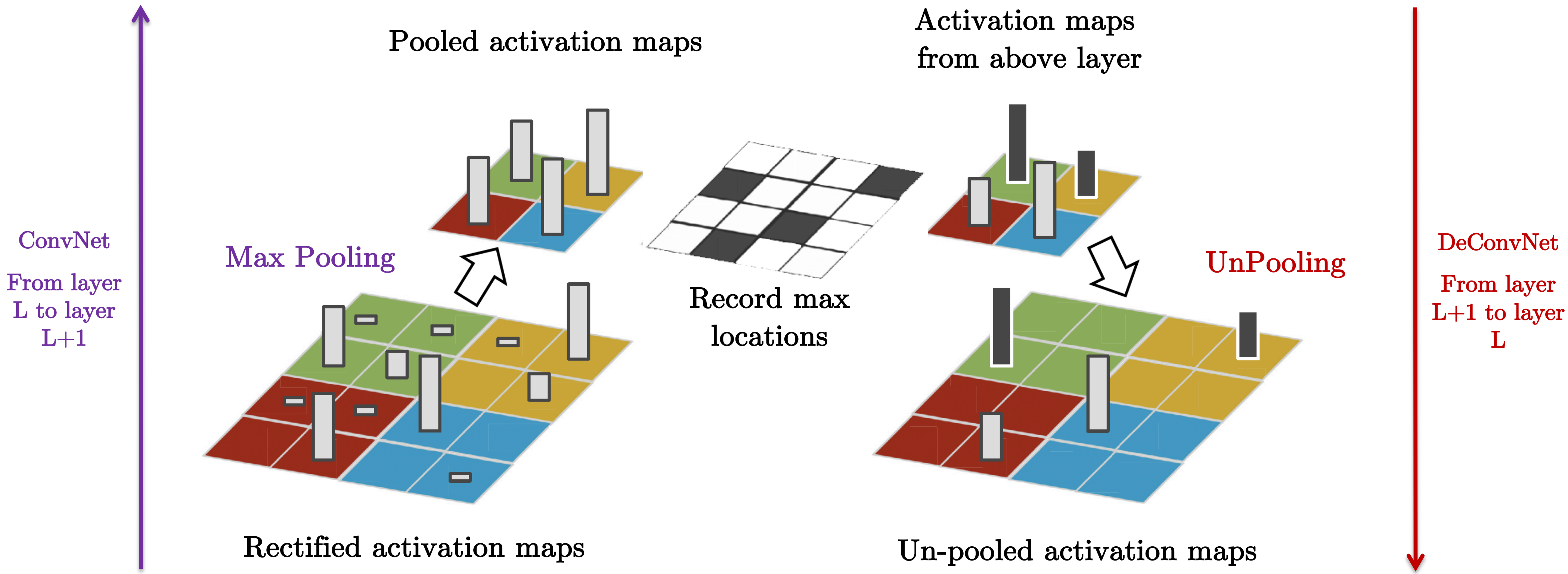
Filters learned 4^{th} layer

ConvNets vs DeConvNets



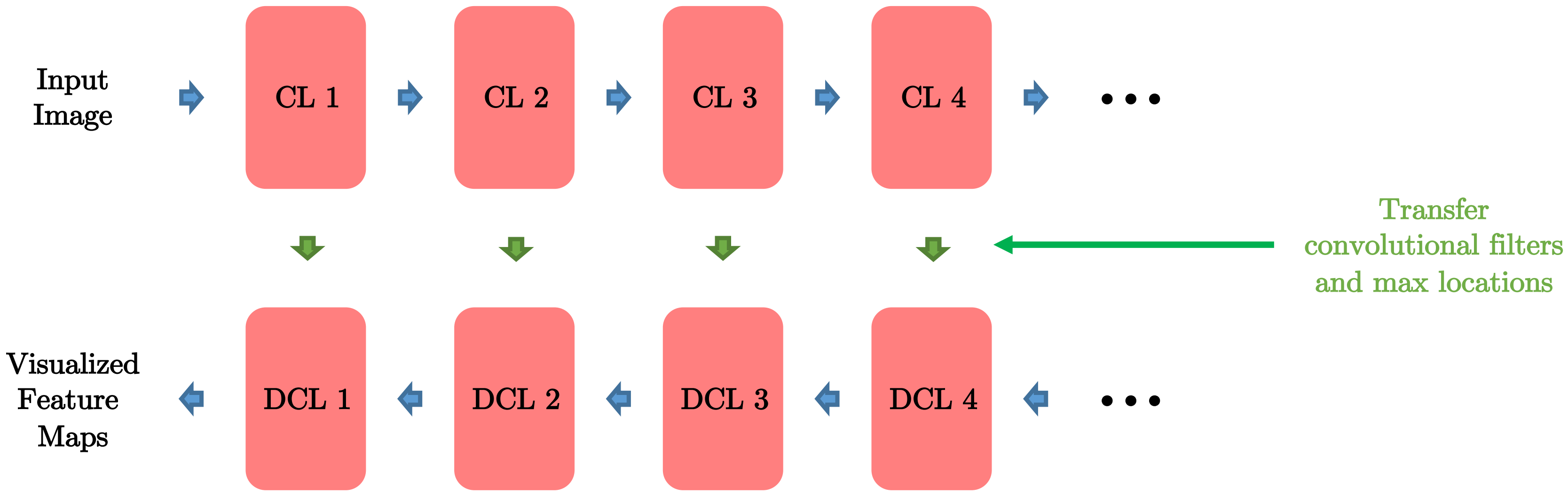
Un-pooling

- Reverse Max Pooling:



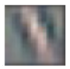
Visualization architecture

- Visualization architecture couples ConvNets and DeConvNets together.
  - Visualization of ConvNet features is done after training.

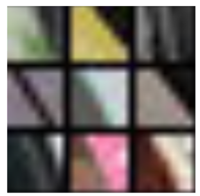


Visualization process

- It works as follows:
  - For the 1^{st} layer, extract all 7x7 patches in the image and feed them to ConvNet.
  - Suppose we have 100 7x7 convolutional filters in the 1^{st} layer.
    - Select 1 filter, find the top-9 7x7 patches with the maximum activation maps.



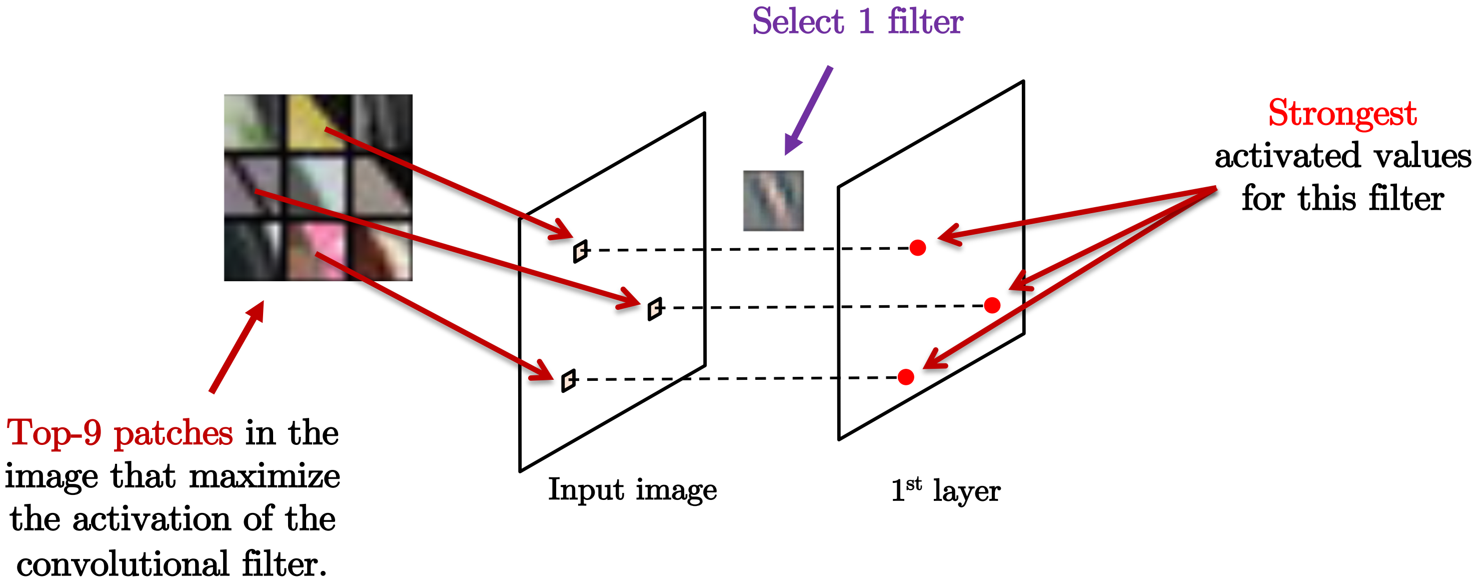
One convolutional filter



Top-9 patches
that maximize the
activation of the
convolutional filter.

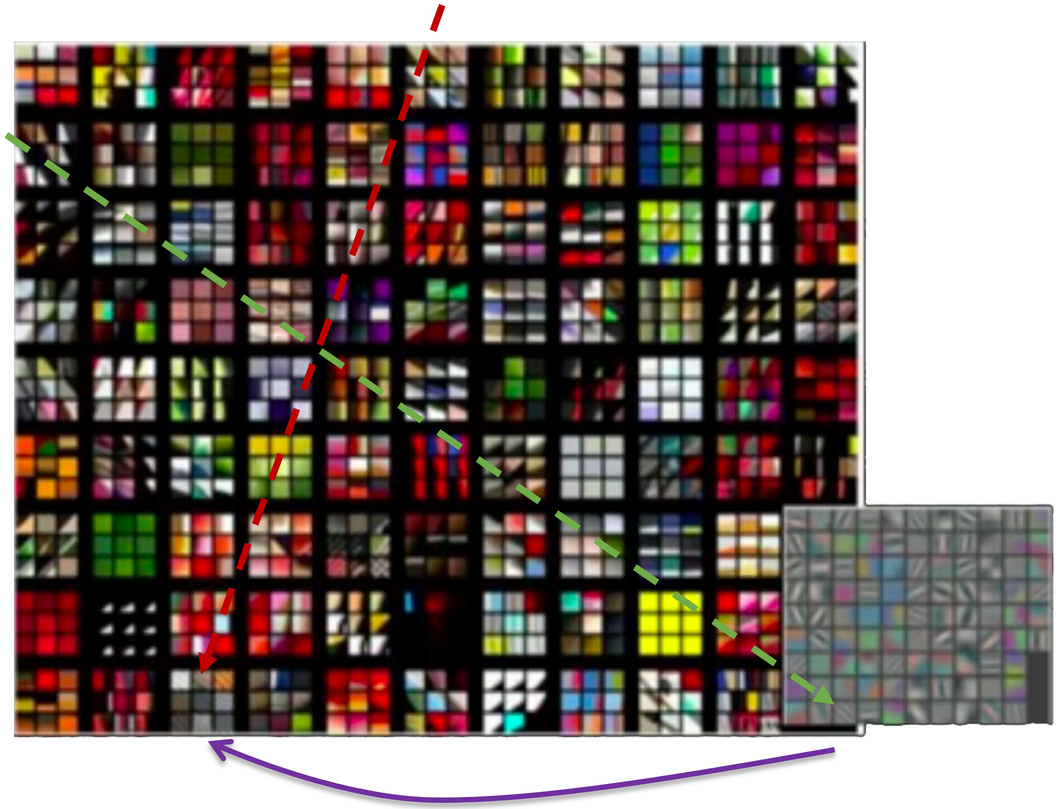
Visualization process

- For the 1^{st} layer, the 7x7 filters learned by ConvNet can be directly visualized.
  - There is no need to use for DeConvNet.



Visualization process

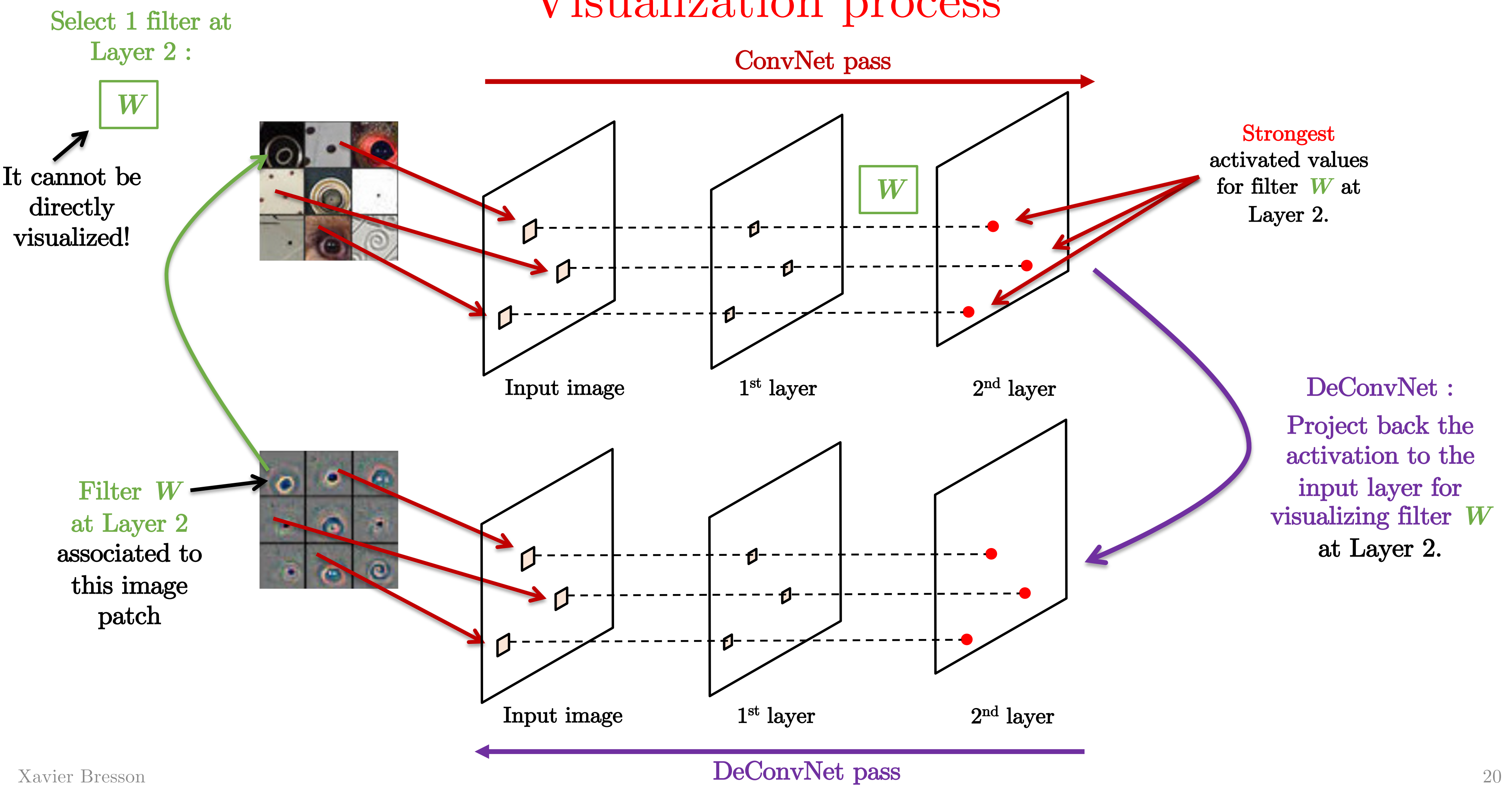
- For the 1st layer :
  - Filters are simply visualized with ConvNet.
  - For each filter, also visualize the associated top-9 7x7 patches.



Visualization process

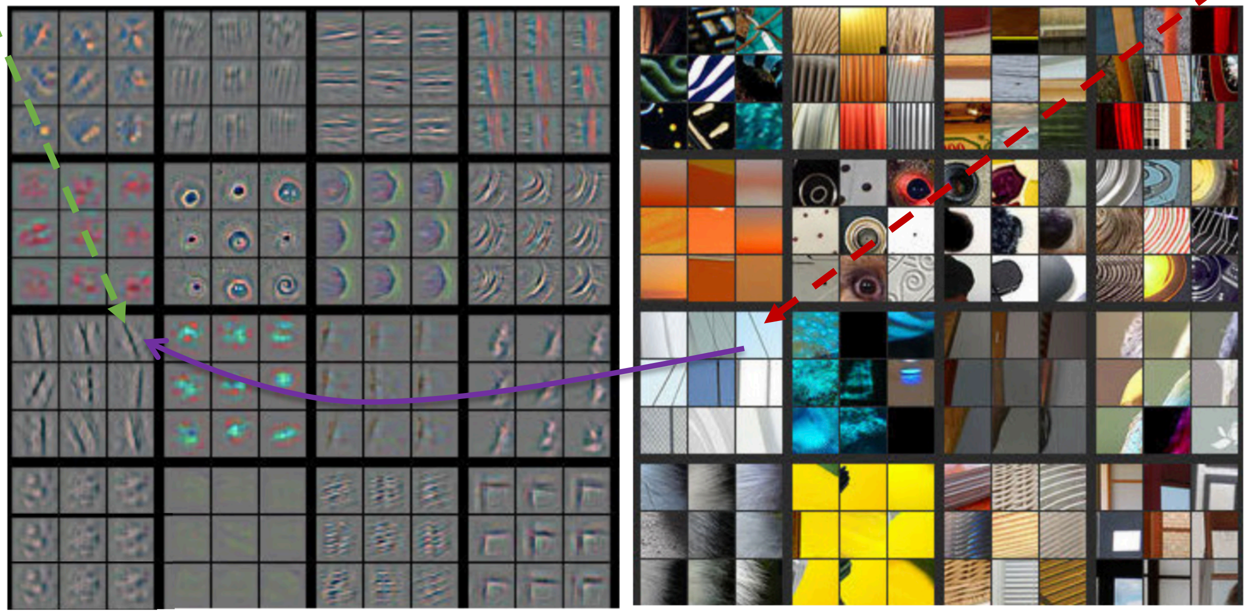
- For the 2^{nd} layer :
  - We cannot directly visualized the filters learned with ConvNets.
  - We proceed as follows:
    - Extract all 11x11 patches in the image and feed them to (learned) ConvNet.
    - Suppose we have 100 7x7 convolutional filters in the 2^{nd} layer.
      - Select 1 filter, find the top-9 11x11 patches with the maximum activation maps.
      - Use DeConNet to visualize the filters associated with these top-9 patches.

Visualization process



Visualization process

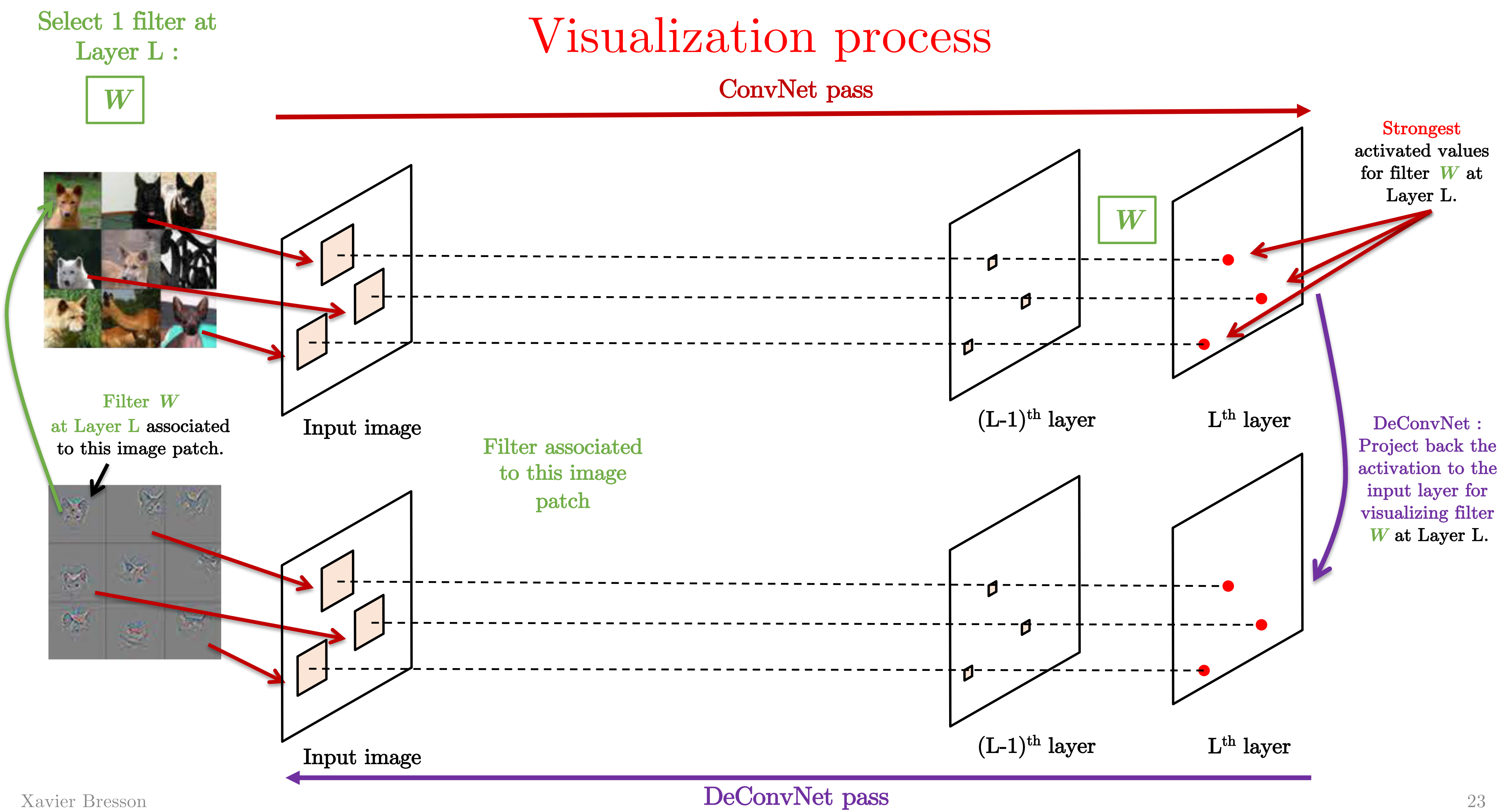
- For the 2^{nd} layer :
  - Filters are visualized with DeConvNet.
  - For each filter and each maximum activation, visualize the associated top-1 11x11 patch.



Visualization process

- For the L^{th} layer :
  - Extract all 21x21 patches in the image and feed them to (learned) ConvNet.
  - Suppose we have 100 7x7 convolutional filters in the L^{th} layer.
    - Select 1 filter, find the top-9 21x21 patches with the maximum activation maps.
    - Use DeConNet to visualize the filters associated with these top-9 patches.

Visualization process



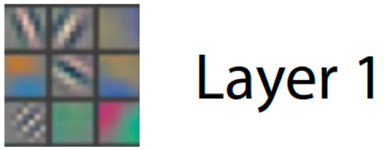
Outline

- ConvNet features
- Domain of dependence of features
- Visualizing learned filters
- **Hierarchical filters**
- Improving ConvNets with visualization
- Conclusion

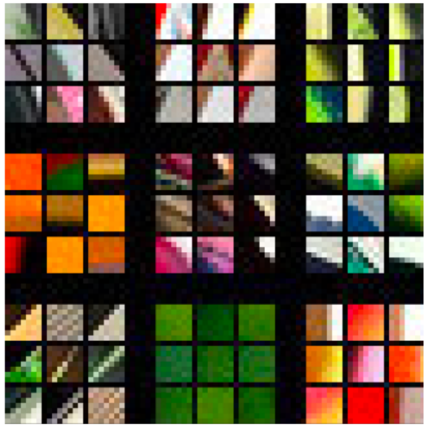
Hierarchical filters

- Let us check if ConvNet learns hierarchical filters.
  - The neurons in the 1^{st} layer respond to orientated edges and colors :

9 filters from the first layer



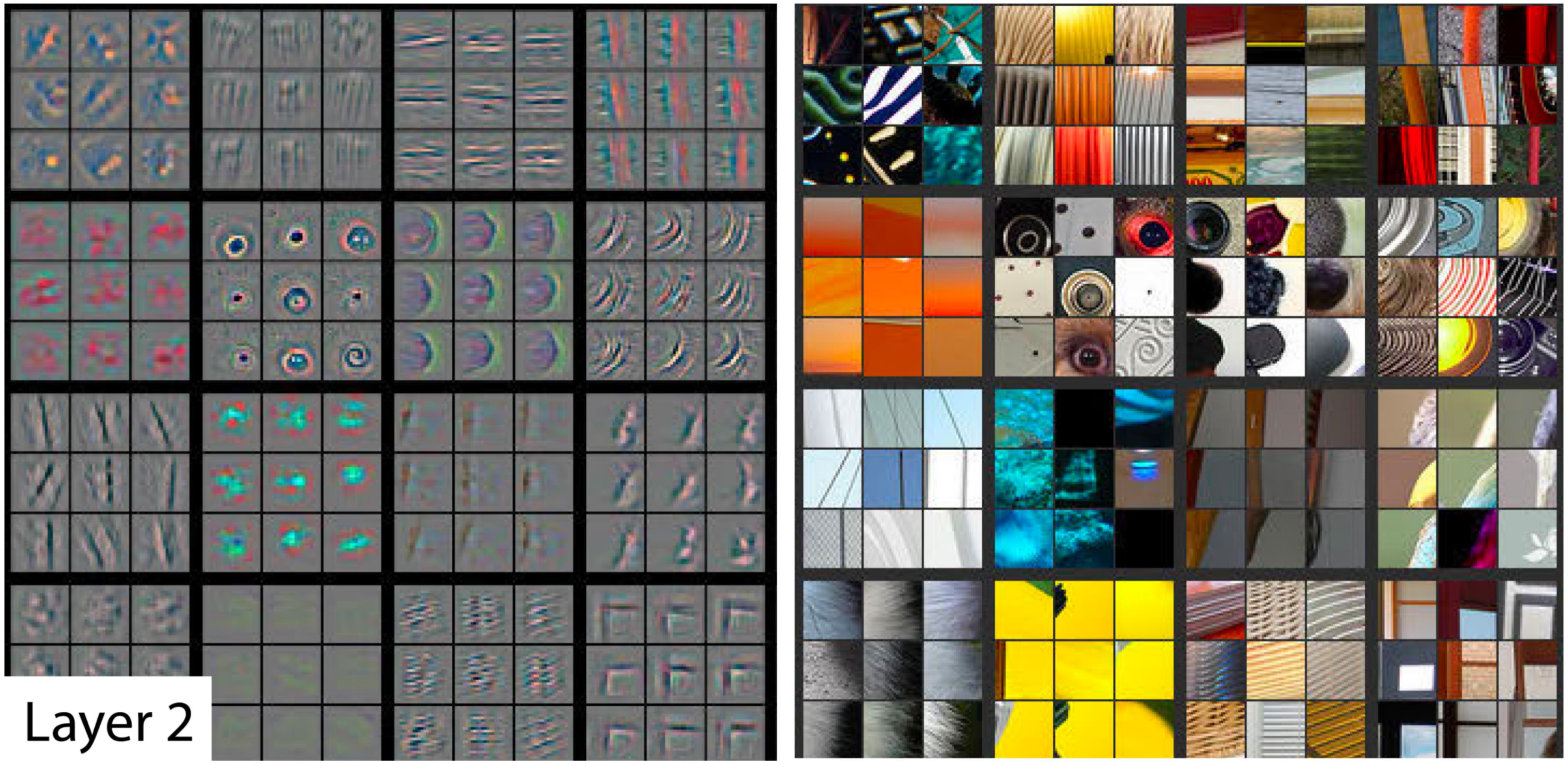
These patches of image activate the corresponding filters



For each filter, the top-9 patches are shown.

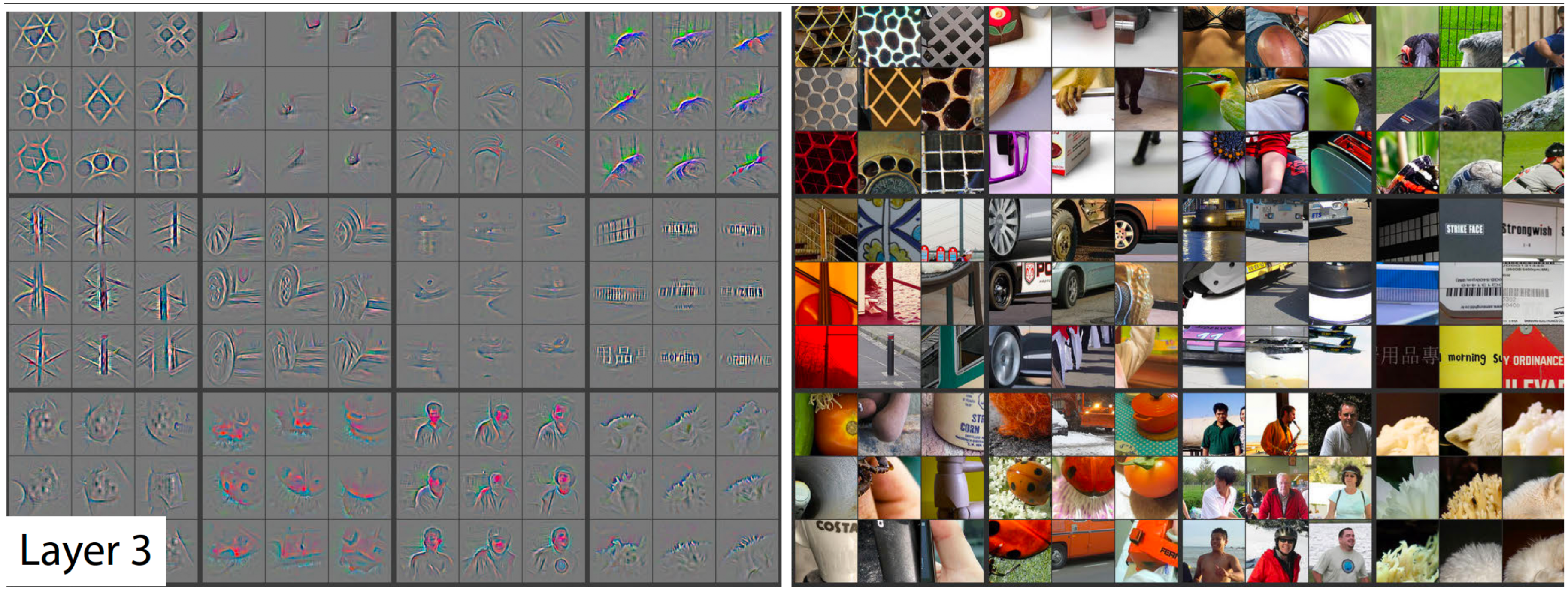
Hierarchical filters

- The 2^{nd} layer responds to corners and other edge/color conjunctions :



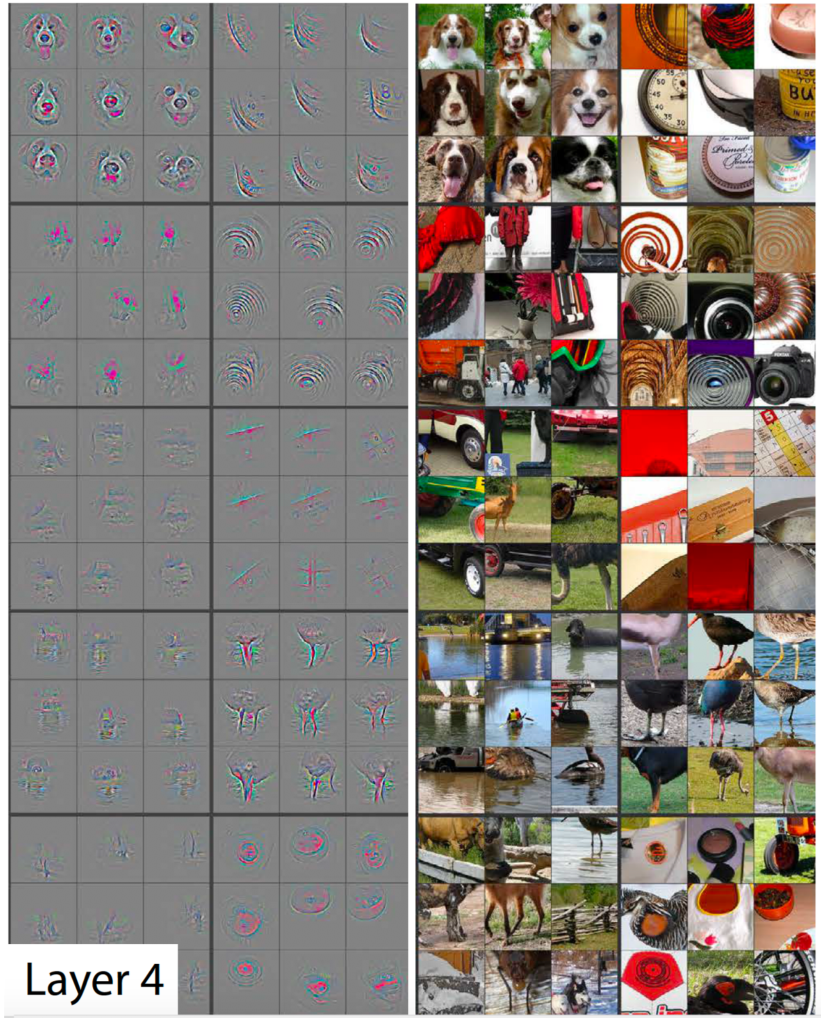
Hierarchical filters

- The 3rd layer responds to more complex patterns: mesh, wheel, text, etc...



Hierarchical filters

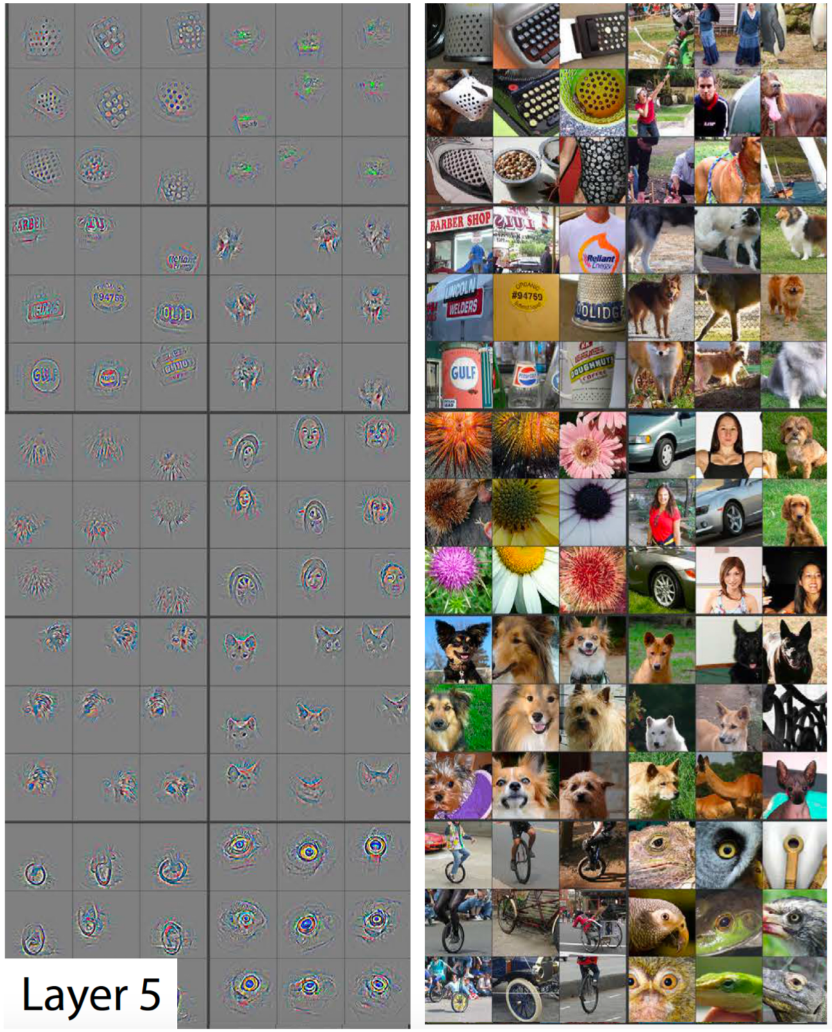
- The 4^{th} layer gets even more complex : Dog faces, birds' legs, ...



Hierarchical filters

- The 5^{th} layer shows entire objects : Full dog, keyboard, ...

The deeper we go in the network, the more complex and abstract are the detected patterns !

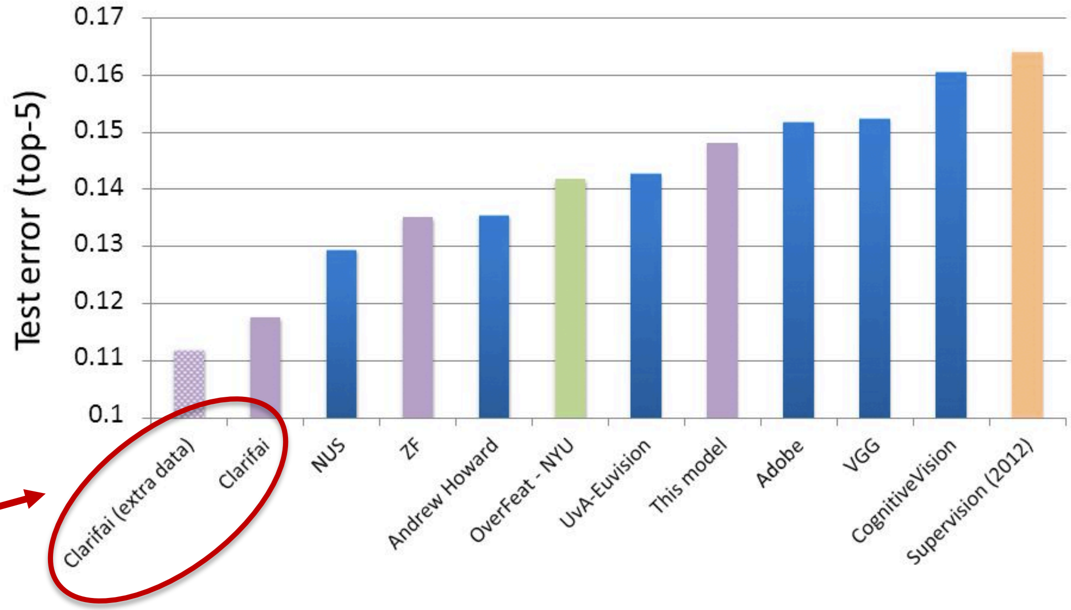


Outline

- ConvNet features
- Domain of dependence of features
- Visualizing learned filters
- Hierarchical filters
- **Improving ConvNets with visualization**
- Conclusion

ImageNet 2013 contest

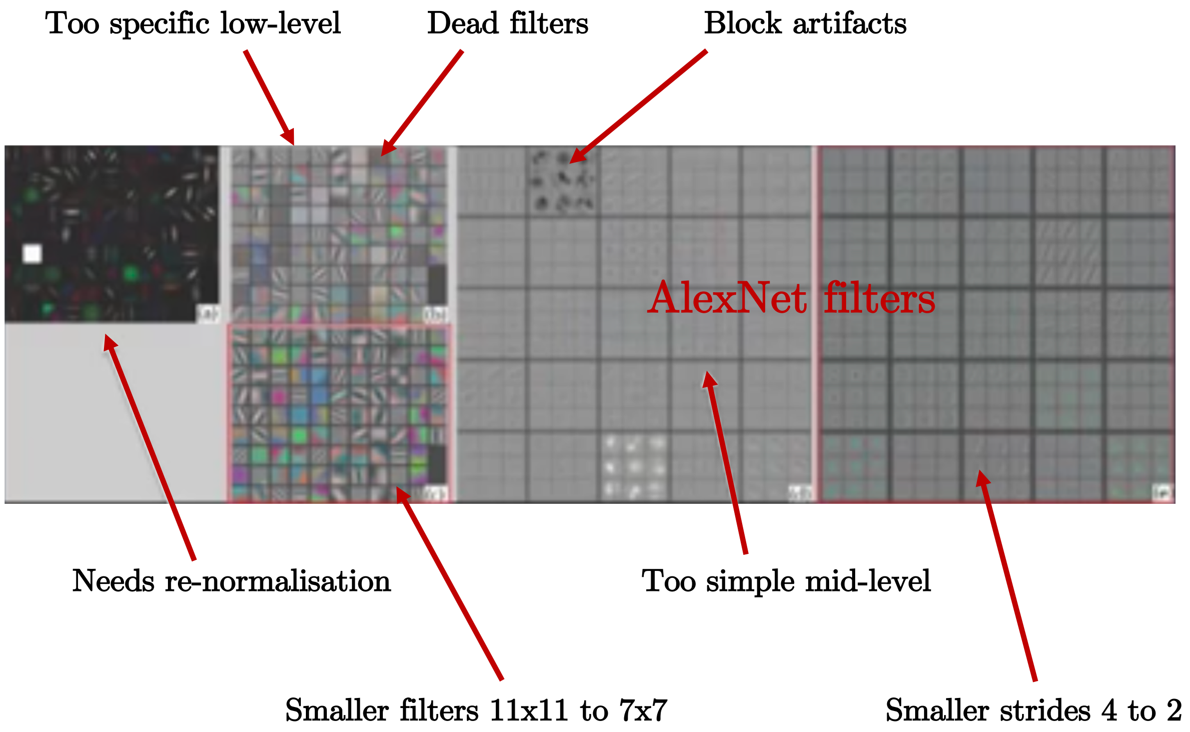
- 1st rank in ImageNet 2013 :
  - Visualization helped to boost the performance on ImageNet classification by 2%, and won the 2013 contest !



Zeiler and Fergus

ImageNet 2013 contest

- Visualization helped to improve the 2012 winner architecture.
- AlexNet architecture have some limitations:
  - Filters in the 1^{st} layer are not normalized ⇒ Re-normalization is needed.
  - Filters is the 1^{st} layer are too large 11x11 and learn too specific low-level filters and some dead filters ⇒ Smaller 7x7 filters fix the issues.
  - There are some block artifacts in the 2^{nd} layer and very simple mid-level filters ⇒ Smaller strides fix these problems.



Outline

- ConvNet features
- Domain of dependence of features
- Visualizing learned filters
- Hierarchical filters
- Improving ConvNets with visualization
- Conclusion

Conclusion

- Why ConvNet works so well?
  - It learns strong discriminative features for visual tasks.
  - Theses features are compositional (from simple to abstract representations).

- Warning :
  - It may also be an anthropomorphic explanation of ConvNets.

- Mathematical reasons of success may be
  - Global and local translation invariance
  - Deep representation
  - SGD is the right tool to find these features.

Questions?