Convolutional neural networks

E. Decencière

MINES ParisTech
PSL Research University
Center for Mathematical Morphology

Contents

Contents

A picture is worth a thousand words

Definition

- Classically, an image is a matrix of values belonging to $[0, \ldots, 255]$ (grey level images) or to $[0, \ldots, 255]^3$ (color images).
- More generally, an image is a q-dimensional array of values belonging to R^d.

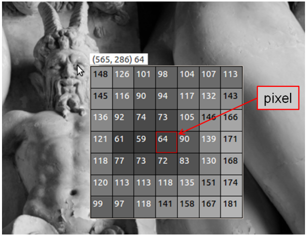


Figure: Grey level values around the left eye of the faun.

Designing computer vision systems that are able to extract semantic information from an image is a difficult task. How can we build systems that extract meaning from an image?

Extracting semantic information from an image

- Where is the phone? (localization task)
- How many mugs are there? (quantification task)
- Is there a window in the room?
- At what time of the day was the photograph taken?

Image processing approach

- Build a mathematical model for the objects you are interested in
- Implement this model using image processing operators

\+ This approach works correctly when the objects are not too complex.

\- If objects are difficult to model, machine learning methods can bring a solution.

Classical machine learning approach

- Compute features from the image
- Apply machine learning to those features

+ Works well when you engineer the right features
- An expert is required to define those features - and this can be a long process
- Annotated data is required

Deep learning approach

Modern neural networks approach

- Directly take as input the image pixels
- The network is supposed to build its own features

+ Good (impressive!) results

- A large amount of annotated data is required

Some accomplishments

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

2012: *AlexNet* [Krizhevsky et al., 2012] won this challenge by a large margin

The database contains more that 1 million training images, belonging to 1000 different classes (including 120 dog breeds!).

Some accomplishments (cont.)

- 2011: first super-human performance, IJCNN 2011 traffic sign recognition contest [Cireșan et al., 2011]
- 2012: visual object detection (Mitosis detection in breast cancer histology [Cireșan et al., 2013])
- 2012: segmentation competition (neuronal membranes in electron microscopy images [Ciresan et al., 2012])
- 2016: AlphaGo beats Lee Sedol, one of the best go players, in a 5-game match

Deep learning applications with images

- classification
- object localization
- semantic segmentation
- instance segmentation
- transformation (filtering, in-painting, editing, colorization...)
- quantification
- compression
- image caption generation
- 2D to 3D (stereo matching, 3D reconstruction, ...)
- motion estimation
- Style transfer
- Anomalous image detection
- Image generation

Convolutional neural networks in deep learning

- They are pivotal to many of the successes achieved by neural networks these recent years
- They are interesting for dealing with regular structured data, such as images (or board games!)

Acronyms

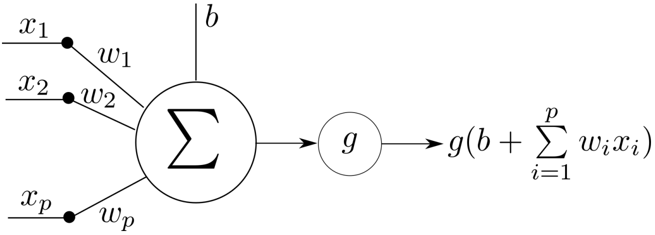
Two acronyms are used for convolutional neural networks in the literature: *CNN* and *ConvNet*.

Essential milestones

- 1979: Neocognitron (CNN architecture) [Fukushima, 1979, Fukushima, 1980]
- 1989: Backpropagation applied to CNNs [LeCun et al., 1989]
- 2006, 2010: GPU implementation [Chellapilla et al., 2006, Cireșan et al., 2010]

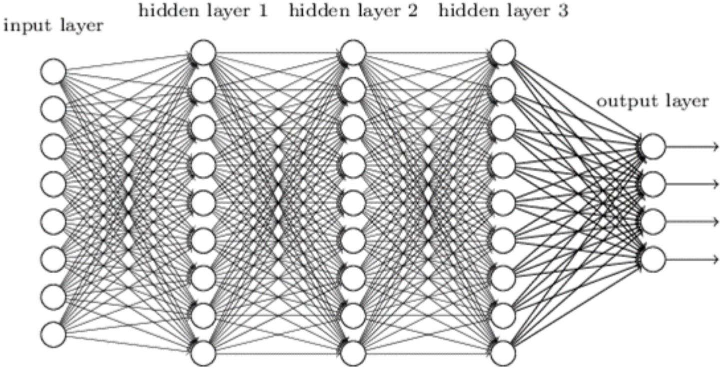
Contents

Artificial neuron



- $b, w_1, \ldots, w_n$ are the neuron parameters, to be learnt
- g is the activation or transfer function

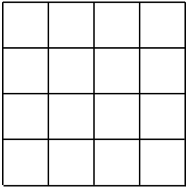
Neural network



(from http://www.jtoy.net)

Input image, input neurons

In the scalar case (single-valued images), each input pixel is considered as an input neuron.



Input image, input neurons

In the scalar case (single-valued images), each input pixel is considered as an input neuron.

Image classification problem

Classification problem:

- Input: image $\mathbf{x}$
- Output: class $y \in \{label_1, label_2, \dots, label_q\}$

Class coding

Often, classes are denoted by integers, but this is only a coding commodity. For instance, it would be meaningless to use a regression approach for this problem.

Class coding

If there are q possible classes, then a class will be coded as a vector $\mathbf{y}$ of length q. If its class is r then for $0 \leq i < q$:

$$\mathbf{y}[i] = \begin{cases} 1, & \text{if } i = r \\ 0, & \text{otherwise} \end{cases}$$

Example with 4 classes

- Label $0 \longmapsto [1, 0, 0, 0]$
- Label $1 \longmapsto [0, 1, 0, 0]$
- Label $2 \longmapsto [0, 0, 1, 0]$
- Label $3 \longmapsto [0, 0, 0, 1]$

Image classification with a neural network

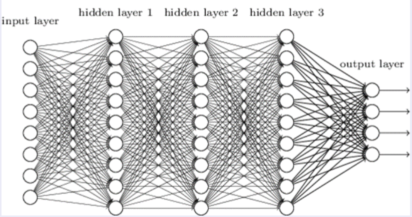
Input

Scalar image is transformed into a vector of length p.

Output

For q classes, the output will be a vector of length q.

Example: image of size 4×2, 4 possible classes



Activation of the last layer

Loss function for classification: cross-entropy

Conclusion on fully-connected networks for image classification

Fully connected layers:

- scale badly to large size images
- do not take into account the local structure of images

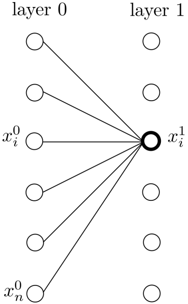
Today:

- Fully-connected networks are almost never used for image analysis.
- Fully-connected layers are only used in the middle (auto-encoders) or at the end (classification) of the pipeline.

Contents

Towards convolutional layers

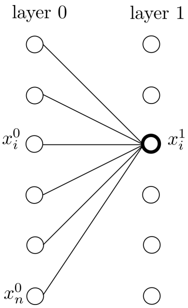
In the following slides, for illustration purposes, we will consider one-dimensional images



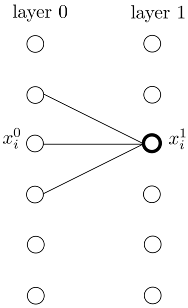
Fully connected layer: $n(s+1)$ weights

Towards convolutional layers

In the following slides, for illustration purposes, we will consider one-dimensional images



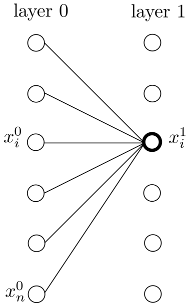
Fully connected layer: $n(s+1)$ weights



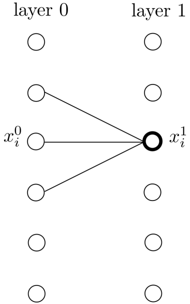
Locally conn. layer: $n(s+1)$ weights

Towards convolutional layers

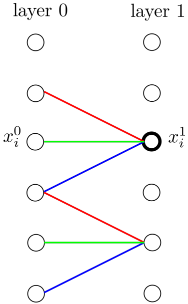
In the following slides, for illustration purposes, we will consider one-dimensional images



Fully connected layer: $n(s+1)$ weights



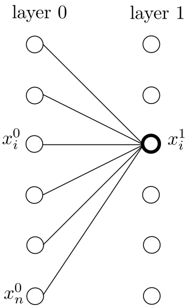
Locally conn. layer: $n(s+1)$ weights



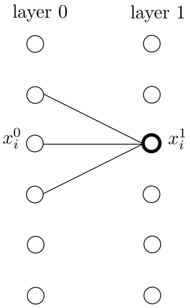
Weight replication: $s+1$ weights

Towards convolutional layers

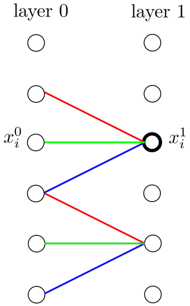
In the following slides, for illustration purposes, we will consider one-dimensional images



Fully connected layer: $n(s+1)$ weights



Locally conn. layer: $n(s+1)$ weights

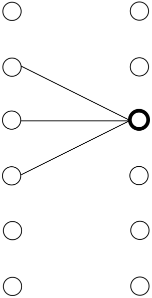


Weight replication: $s+1$ weights

A convolutional layer computes a convolution, plus a constant, of the precedent layer.

Stride

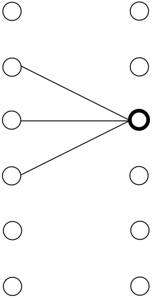
A convolutional layer can at the same time downsample the image by applying a sampling step, or *stride*.

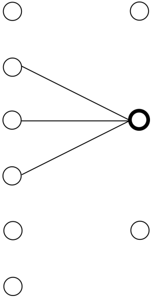


Stride 1

Stride

A convolutional layer can at the same time downsample the image by applying a sampling step, or *stride*.

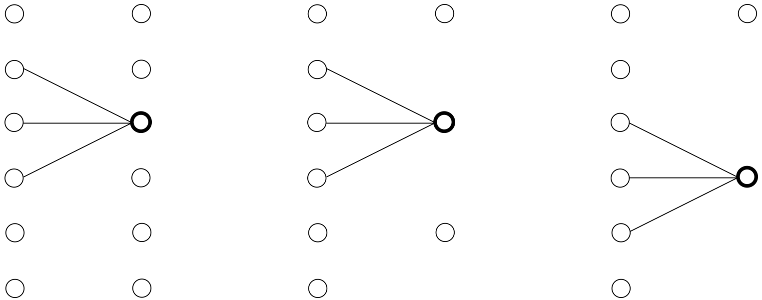




Stride 1 Stride 2

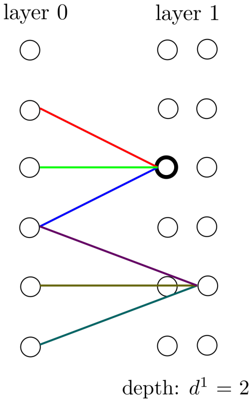
Stride

A convolutional layer can at the same time downsample the image by applying a sampling step, or *stride*.

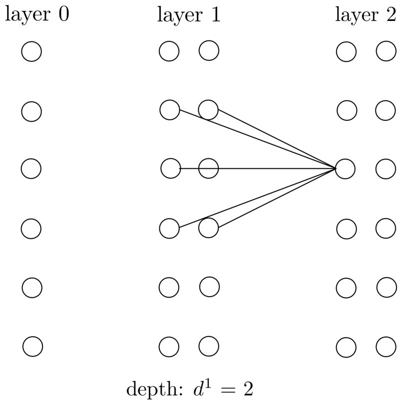


Stride 1 Stride 2 Stride 3

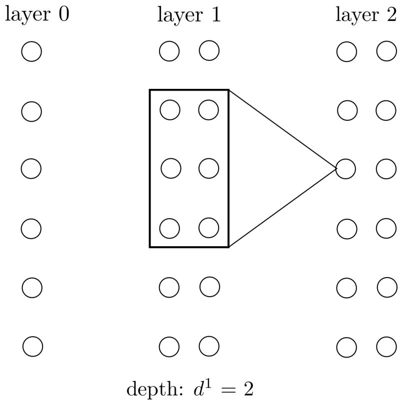
Several filters in the same convolutional layer



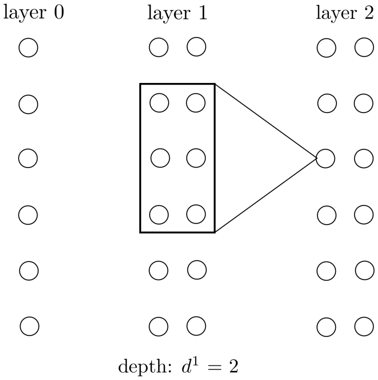
Several filters in the same convolutional layer



Several filters in the same convolutional layer

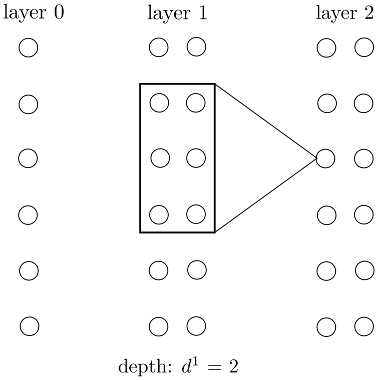


Consequences on the parameter number



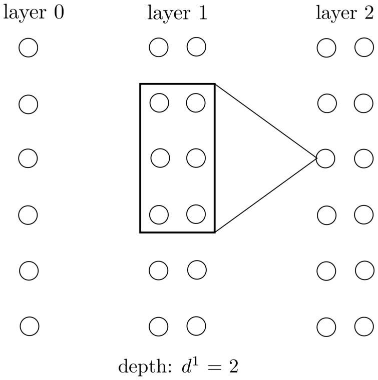
- How many parameters do we have in layer 1?

Consequences on the parameter number



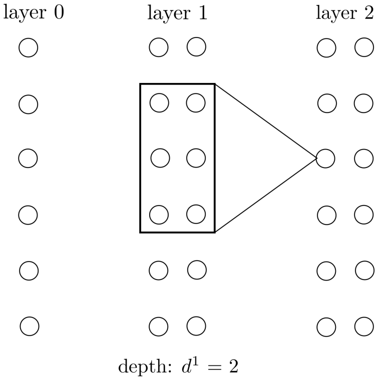
- How many parameters do we have in layer 1?
- $d^1 \times (s+1)$

Consequences on the parameter number



- How many parameters do we have in layer 1?
- $d^1 \times (s+1)$
- In layer 2?

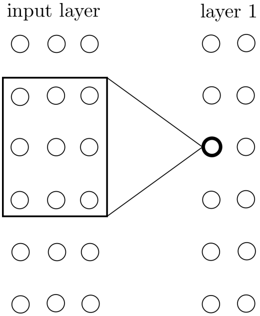
Consequences on the parameter number



- How many parameters do we have in layer 1?
- $d^1 \times (s+1)$
- In layer 2?
- $d^2 \times (d^1 \times s + 1)$

Multi-valued images

An input image with p channels (for instance à colour image with 3 channels) can be represented by an input layer of depth 3



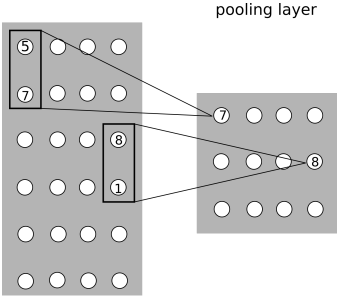
Some properties

- Translation invariance
- Efficient implementation using matrix operations and graphical processing units

Contents

Max pooling

- Convolutional networks often contain subsampling steps. These can be done using strides of 2 or more within convolutional layers or, as it is common practice today, using *max pooling* layers.
- Sampling is only applied along the spatial dimensions, not along the dimension of the filters.



Skip connections

Main components of a convolutional neural network

Many successful architectures, especially for image classification, follow the same pattern:

1. Several iterations of: One or several convolutional layers, with increasing depth, followed by max pooling
2. A few fully connected layers

VGGnet

- Proposed by K. Simonyan and A. Zisserman from the University of Oxford [Simonyan and Zisserman, 2014]
- Runner-up in the ImageNet Large Scale Visual Recognition Competition (ILSVRC) in 2014.
- Number of parameters (VGG16): 138 million.

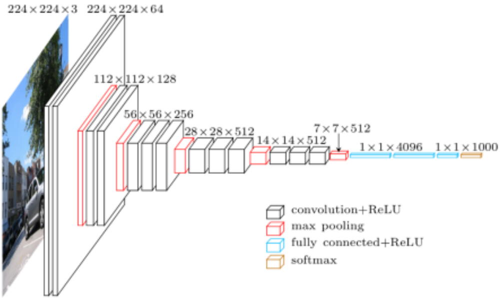


Figure: VGG16 (From https://www.cs.toronto.edu/ frossard/post/vgg16/)

GoogLeNet (a.k.a. Inception v1)

- Winner of the ImageNet Large Scale Visual Recognition Competition (ILSVRC) in 2014.
- Number of parameters: *only* 5 million.

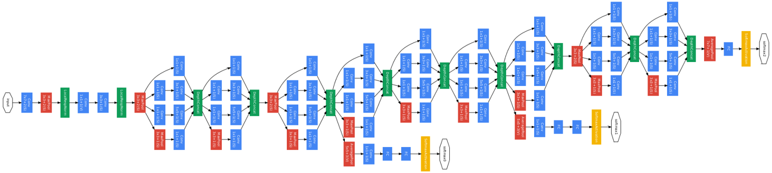


Figure: From [Szegedy et al., 2014]

ResNet won the following year...

Inception module

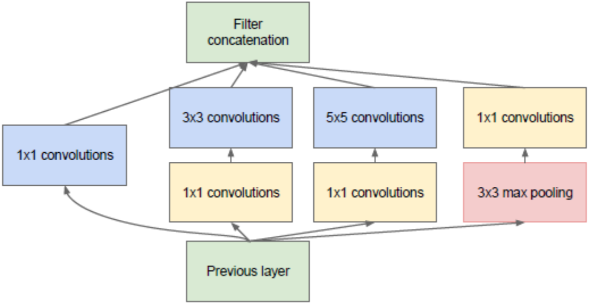


Figure: From [Szegedy et al., 2014]

ResNet

- Winner of the ImageNet Large Scale Visual Recognition Competition (ILSVRC) in 2015.
- The authors tested up to 1202 layers. They reported no training difficulties, but overfitting [He et al., 2015]

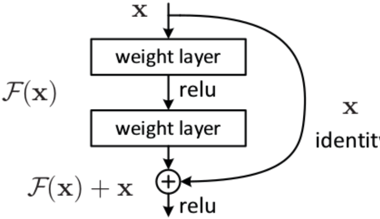


Figure: Residual learning block (from [He et al., 2015])

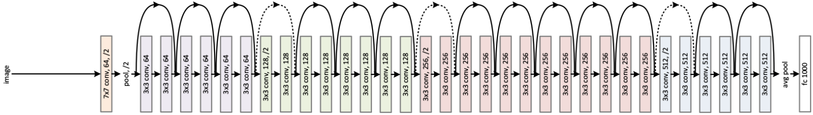


Figure: Residual network with 34 layers (from [He et al., 2015])

Current trends

- Small convolutions (3×3)
- Dimension reduction using 1×1 convolutions
- Increasing number of layers
- Skip connections

Contents

Designing a convolutional neural network

- Network architecture
- Hyper-parameter setting
- Optimization method and parameters

Learning, specially for complex networks, can be difficult. It is always time consuming.

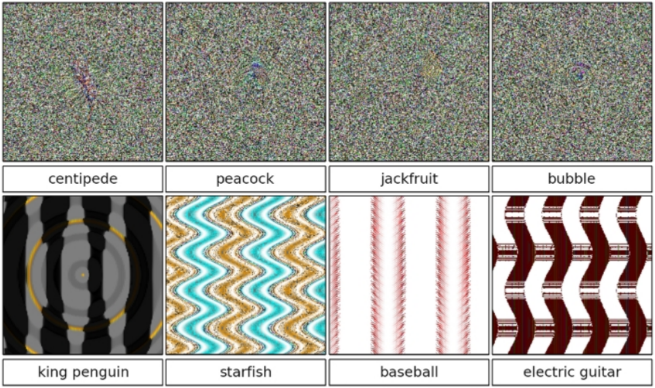
Using an existing network

- Replicate network architecture and hyper-parameters
- Load weights

With standard libraries this is very simple. Prediction time is usually fast.

ConvNets can be fooled

Deep learning can produce astonishing results [Nguyen et al., 2015]...



Main deep learning libraries

Deep learning is a very competitive domain, where code sharing is very common.

- Theano
- Torch
- Tensorflow
- Caffe
- MatConvNet

Keras

Keras is a very easy to use interface to Theano and Tensorflow.

Contents

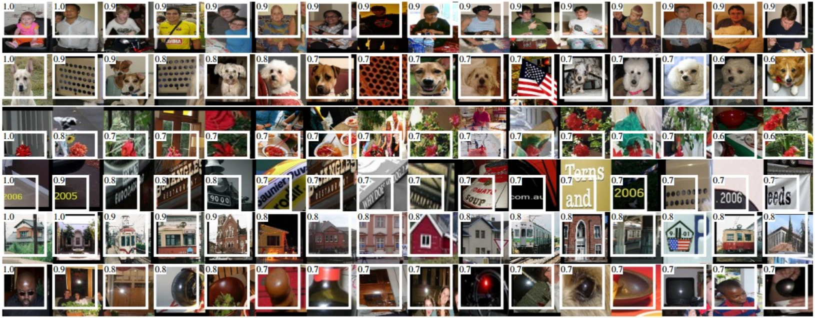
Visualizing at least what we don't understand

We do not fully understand today where the good performances of deep learning come from. But we can at least have a look at what is going on inside the networks.

- Neuron outputs (activations)
- Filter values
- "Important" pixels

Maximal neuron activation

Which images maximally activate a given neuron?
[Girshick et al., 2014]



Contents

A revolution in image analysis

- Deep learning has brought an undeniable break-through in image analysis (as in other fields)
- A significant part of research efforts in image analysis today is base on deep learning
- Its applications are ubiquitous

Limitations

For a deep-learning solution to work, you need:

- Enough annotated data
- A lot of fiddling (different architectures; hyper-parameters; optimization)
- One (or, even better, several) powerful GPUs

Moreover, these models lack interpretability.

References I

[Chellapilla et al., 2006] Chellapilla, K., Puri, S., and Simard, P. (2006). High Performance Convolutional Neural Networks for Document Processing. In *Tenth International Workshop on Frontiers in Handwriting Recognition*. Suvisoft.

[Ciresan et al., 2012] Ciresan, D., Giusti, A., Gambardella, L. M., and Schmidhuber, J. (2012). Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 2843–2851. Curran Associates, Inc.

[Cireșan et al., 2011] Cireșan, D., Meier, U., Masci, J., and Schmidhuber, J. (2011). A committee of neural networks for traffic sign classification. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 1918–1921. IEEE.

[Cireșan et al., 2013] Cireșan, D. C., Giusti, A., Gambardella, L. M., and Schmidhuber, J. (2013). Mitosis Detection in Breast Cancer Histology Images with Deep Neural Networks. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*, pages 411–418. Springer, Berlin, Heidelberg.

[Cireșan et al., 2010] Cireșan, D. C., Meier, U., Gambardella, L. M., and Schmidhuber, J. (2010). Deep, Big, Simple Neural Nets for Handwritten Digit Recognition. *Neural Computation*, 22(12):3207–3220.

References II

[Fukushima, 1979] Fukushima, K. (1979). Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position- Neocognitron. *ELECTRON. & COMMUN. JAPAN*, 62(10):11–18.

[Fukushima, 1980] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202.

[Girshick et al., 2014] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR 2014*, pages 580–587.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*. arXiv: 1512.03385.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

[LeCun et al., 1989] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551.

References III

[Nguyen et al., 2015] Nguyen, A., Yosinski, J., and Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[Szegedy et al., 2014] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going Deeper with Convolutions. *arXiv:1409.4842 [cs]*. arXiv: 1409.4842.