CNN: Convolutional Neural Networks

Diane Lingrand and many contributors

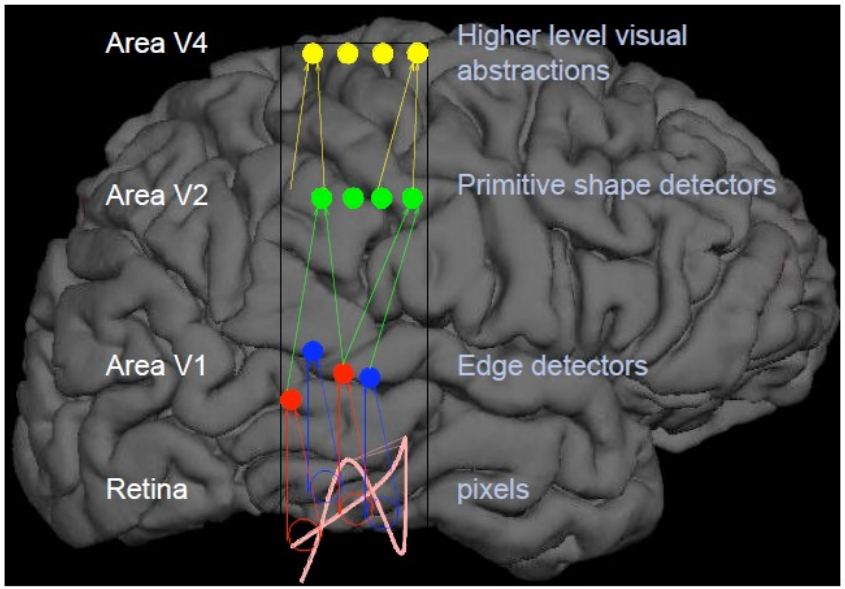
SI 4

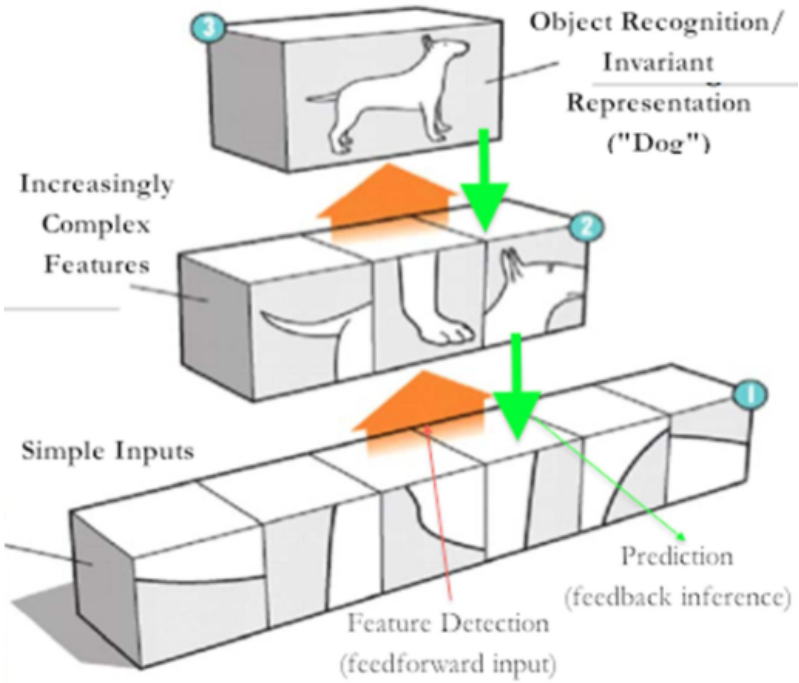
2018 - 2019

Outline

Outline

Deep Architecture in the Brain





Motivation

- image classification
  - neural network as a function for image classification
    - huge number of weights to learn (nb. layers x nb. neurons)
    - even with GPUs
  - inspired by old fashion image processing methods
    - 70's : Sobel, Laplace, Kirsch, Prewitt, Mean or Gaussian smoothing
    - 80's : power of two $\Rightarrow$ integer
    - 90's : SIFT, SURF, ...
    - all are based on convolution functions

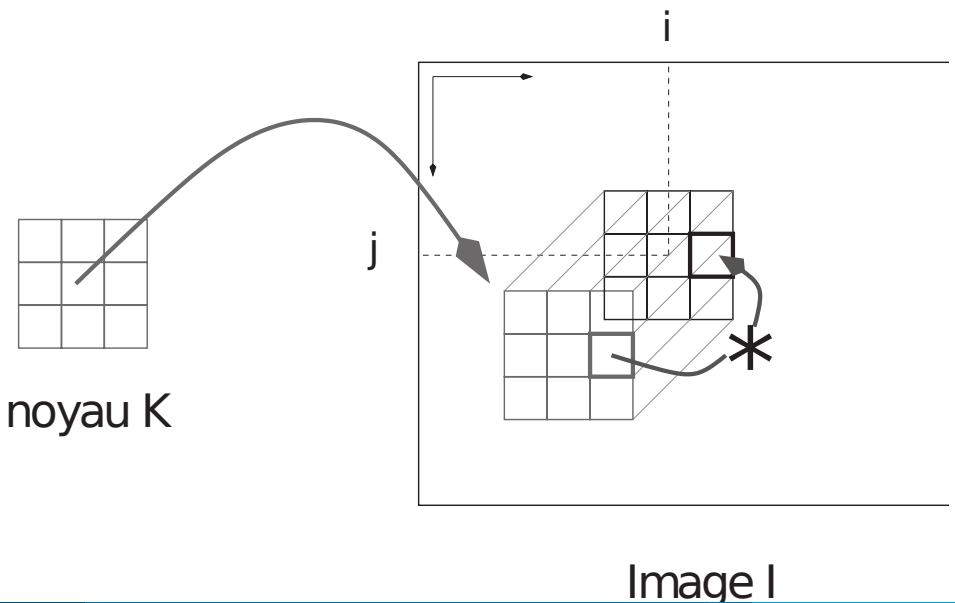
Back to convolution operator

- For f and g, discrete functions :

$$f * g(x, y) = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} f(x, y) g(u - x, v - y)$$

- Convolution of image I_1 by a kernel K of dimension $(2p + 1)x(2q + 1)$:

$$I_2[i][j] = \sum_{k=0}^{2p} \sum_{l=0}^{2q} I_1[i - k + p][j - l + q] K[k][l]$$



Idea of CNN : convolution layer

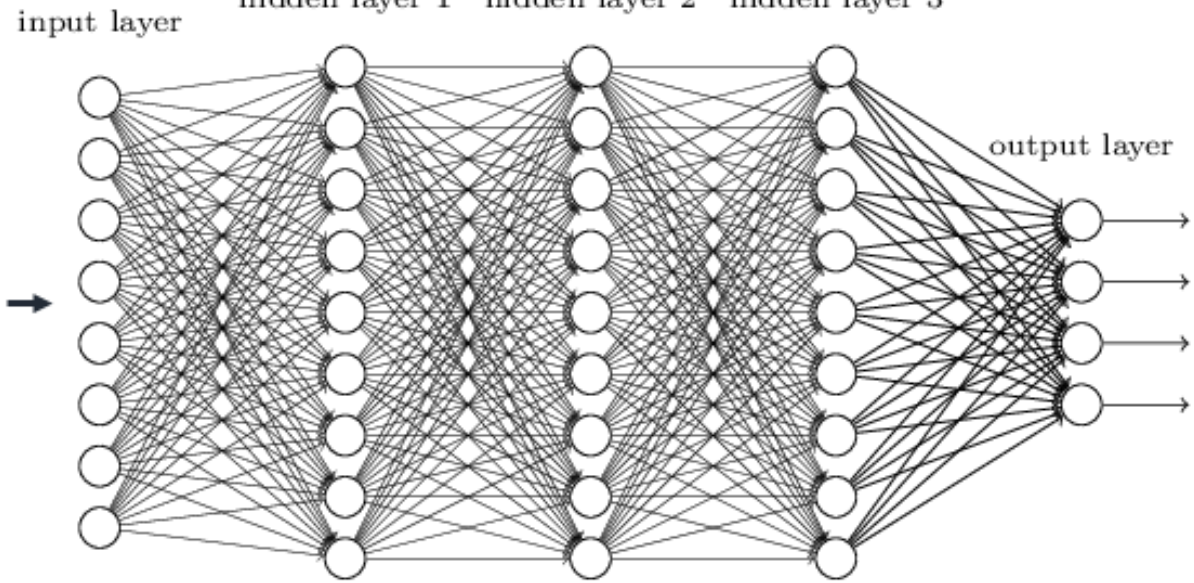
- learning the convolution filters
  - convolution : sum of weighted entries
  - 1 convolution filter $=$ 1 neuron
  - the weights are the filter coefficients
- sharing weights with different neurons
  - the same convolution is applied to every pixels in the image
  - less weights to learn !
- in keras : `Conv2D` instead of `Dense`
  - parameters : number of filters, size of filters (usually $k * k$)
  - input : size of image and number of channels (3 if RGB)
  - output : tensor of dimension nb. filters, dim of images, nb. channels
  - `Conv1D` and `Conv3D` also exists

CNN : Convolutional Neural Network

- many convolutionnal layers
  - in order to extract the characteritics of images
  - but not only convolutionnal layers :
    - pooling
  - RELU activation
- two hidden fully connected layers at the output
  - the function of classification

CNN : Representative layers

Deep neural networks learn hierarchical feature representations



Learning a CNN

- GPUs !
- "Baby sitting" you deep network
  - variations of loss function or metrics with respect to epochs
  - regularisation (L1, L2, Elastic)
  - drop out
  - batch normalisation
  - optimisation function (Momentum, RMSProp, Adam, ...)

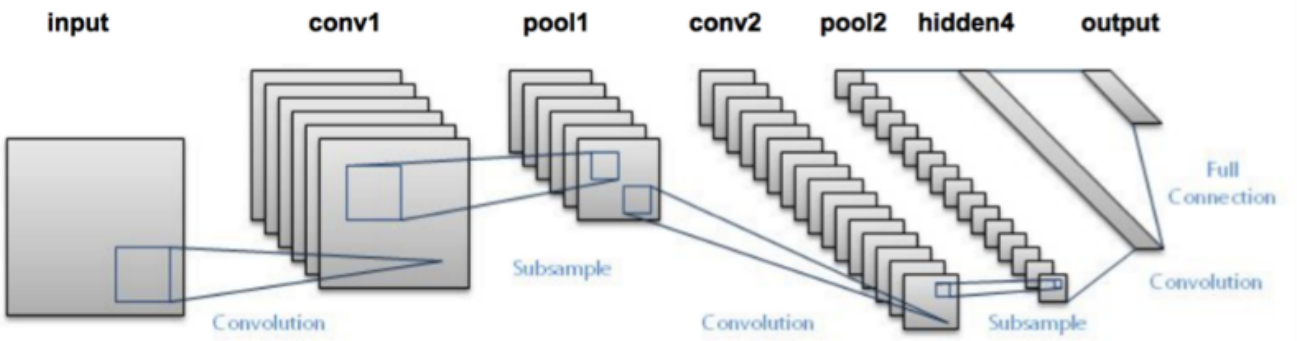
Outline

Architectures

from `https://medium.com/@sidereal/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-mor`

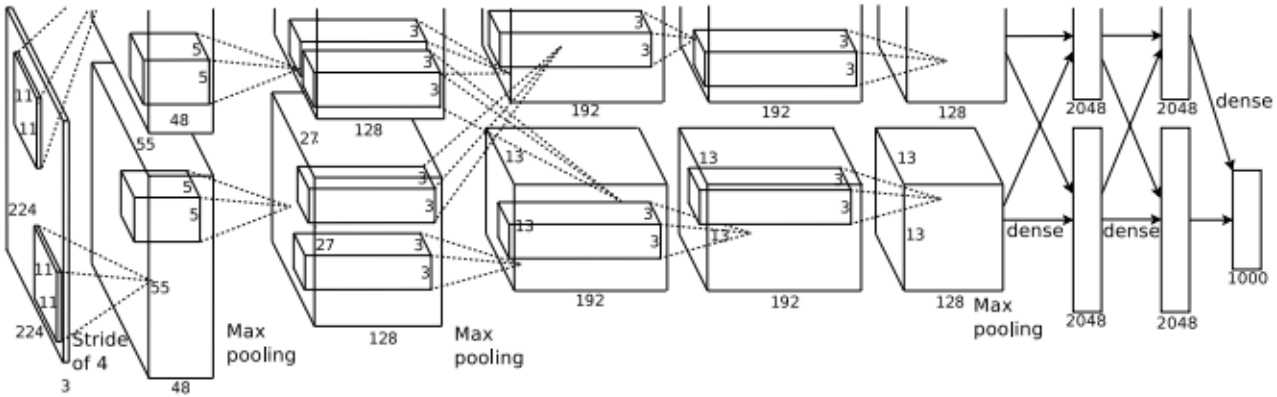
- LeNet 5 1998
  - by Le Cun *et al*
  - was applied to recognise hand-written numbers on checks (32x32 pixel grey images).
- AlexNet 2012 AlexNet2012.png
  - 11x11, 5x5 and 3x3 convolutions, max pooling, dropout, data augmentation, ReLU activations, SGD with momentum. ReLU activations after every convolutional and fully-connected layer.
- GoogLeNet Inception 2014
  - 22 layers
- VGGNet (2014)
  - 16 convolutional layers
- ResNet 2015
  - introduce residual connections

LeNet 5, 1998



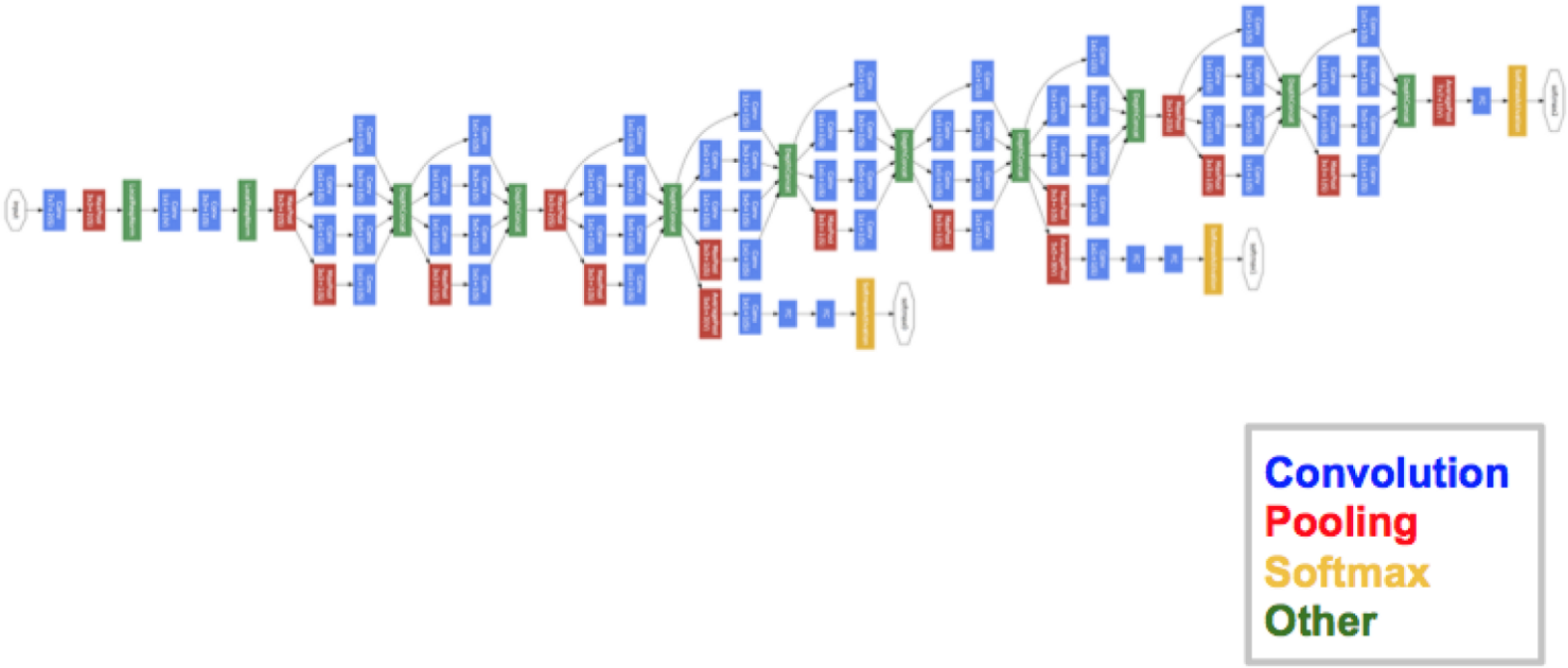
- historical
- by Le Cun et al
- was applied to recognise hand-written numbers on checks (32x32 pixel grey images).

AlexNet, 2012



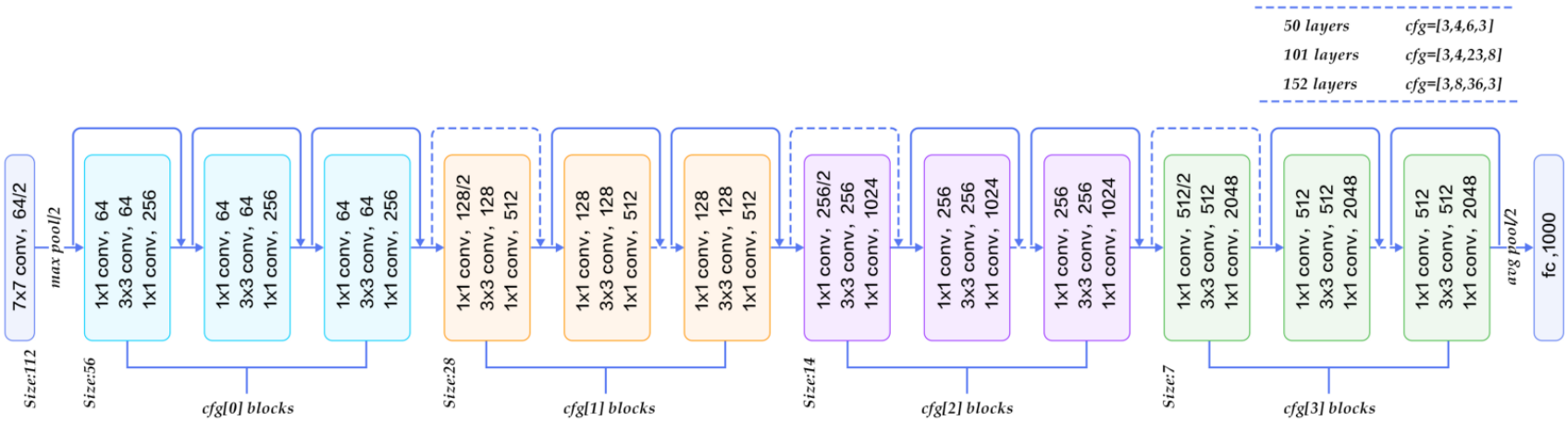
- 11x11, 5x5 and 3x3 convolutions, max pooling, dropout, data augmentation, ReLU activations, SGD with momentum. ReLU activations after every convolutional and fully-connected layer

GoogLeNet Inception v1, 2014



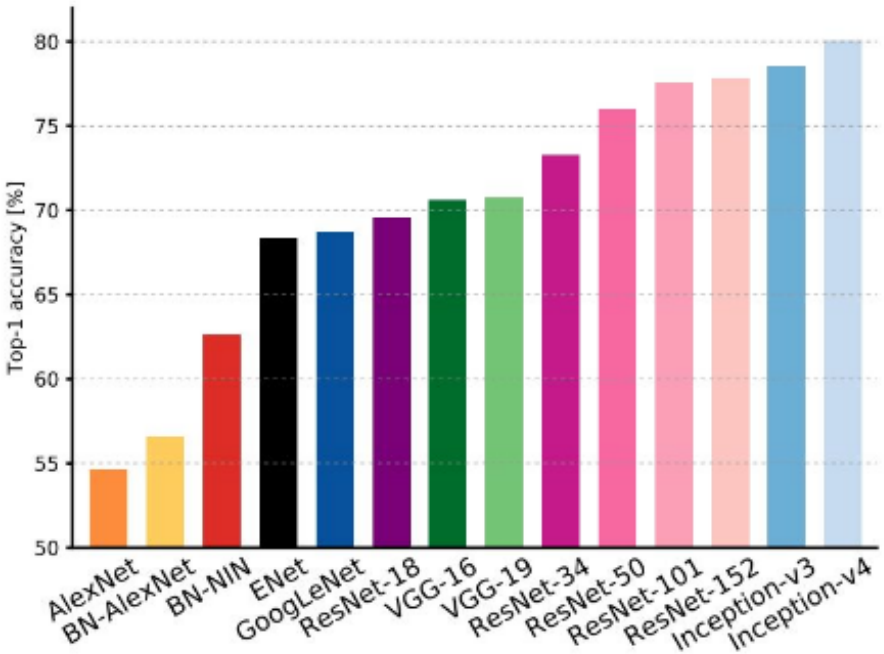
- 22 layers
- v2 and v3 published in 2015
  - `https://arxiv.org/pdf/1512.00567v3.pdf`

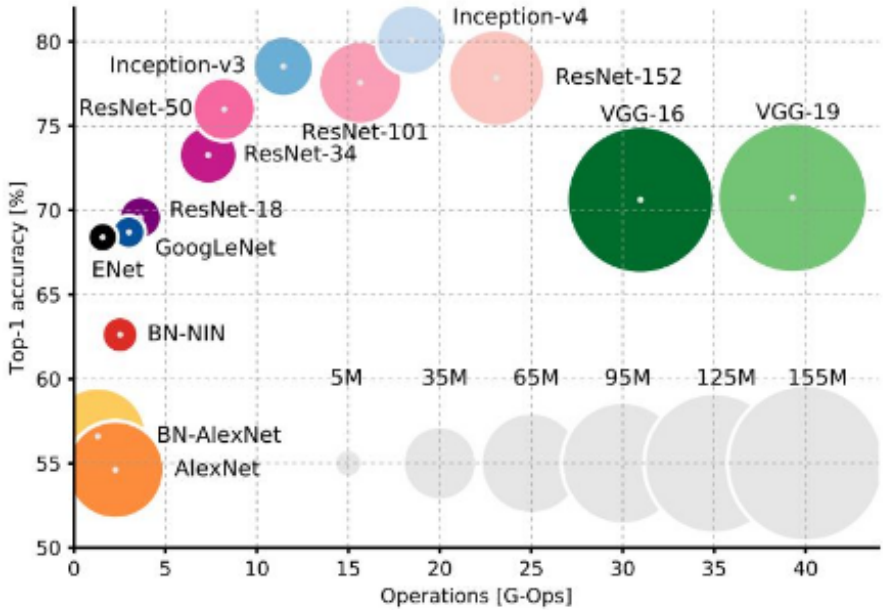
ResNet, 2015



- introduce residual connections

Different architectures



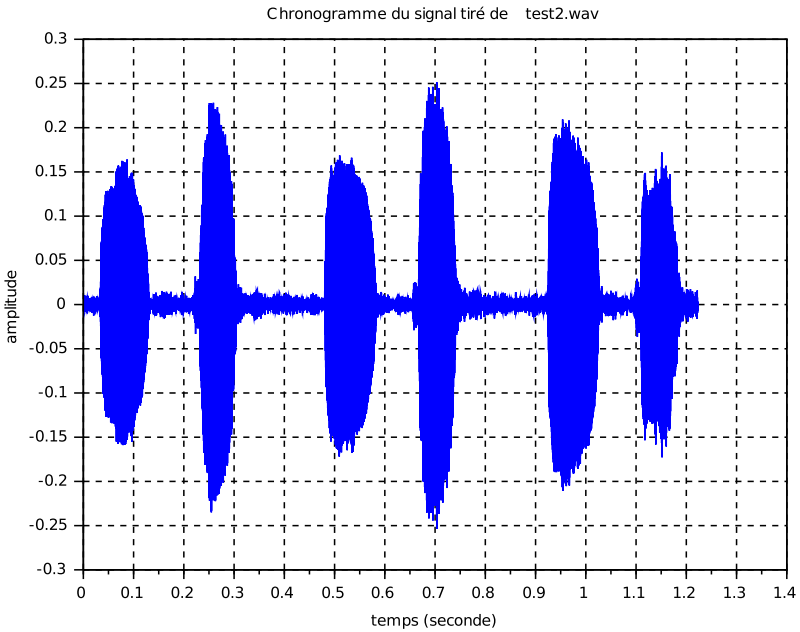


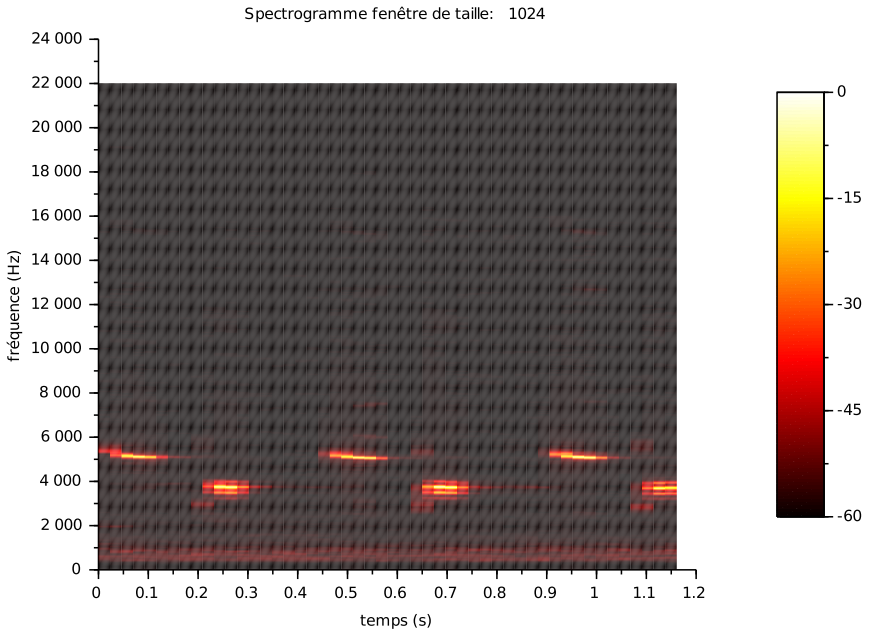
An Analysis of Deep Neural Network Models for Practical Applications, 2017.

Applications

- image classification
- image descriptor
  - output of convolution layer as a vector describing an image
- transfert learning
  - learning only the last fully connected layers for a new image classification task
- sound recognition
  - spectrogram of sound as an image
  - transfert learning for sound classification
- art style transfert
- the problem of adversarial examples

Spectrogram





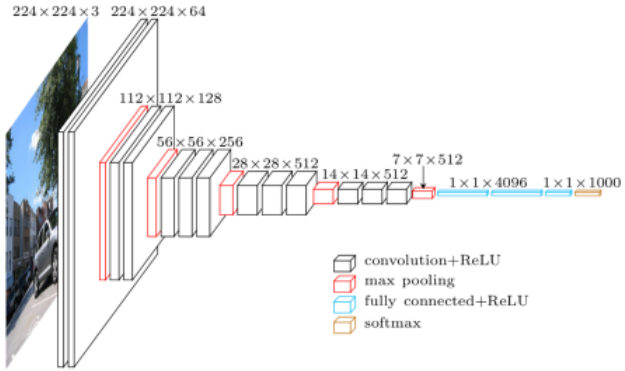
Outline

Art style transfert

- 2015 *A Neural Algorithm of Artistic Style* by Gatys *etal*
  - `https://arxiv.org/abs/1508.06576`
- `http://www.subsubroutine.com/sub-subroutine/2016/11/12/painting-like-van-gogh-with-convolutional-neural-networks`
- try it at `deepart.io`

Art style transfert : How does it work ?

- Based on a VGG16 CNN (Diagram reproduced from the Heuritech blog) :

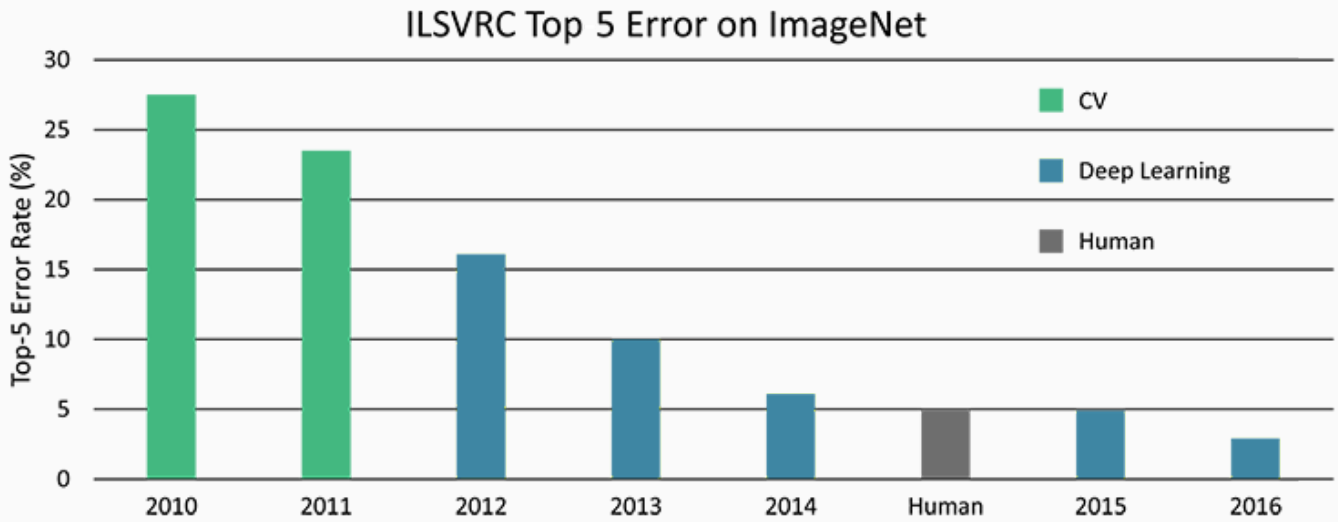


- Extract the content :
  - from layer 5_2 (2nd conv. layer from 5th conv. block) or 4_2
  - 2 images have same content if there output from layer 5_2 are similar (Euclidean distance)
- Capture the style :
  - from layers 1_1, 2_1, 3_1, 4_1 and 5_1.
  - Gram matrix : $F^T F$ where F is the output of the layers
- Algorithm :
  - start with a random image
  - compute the output of all layers up to 5_2
  - compute the cost (similar content + similar style)
  - backpropagate the gradient in order to **modify pixels**.

Outline

We did it!

- Deep Networks are as good as humans at recognition, identification...



How much does a deep network understands those tasks?

Adversarial examples

- images
- sounds
- synthesized 3D objects

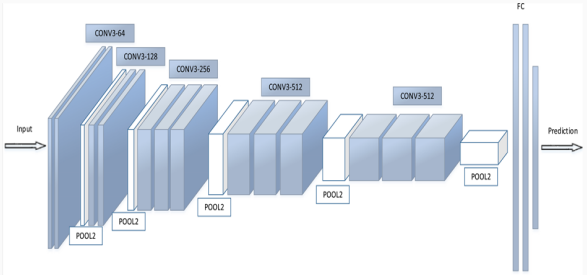
Adversarial images

- next slides from :
  `http://www.telecom-valley.fr/wp-content/uploads/2017/05/DEBARD.pdf`

A Simple Experiment: What we expected

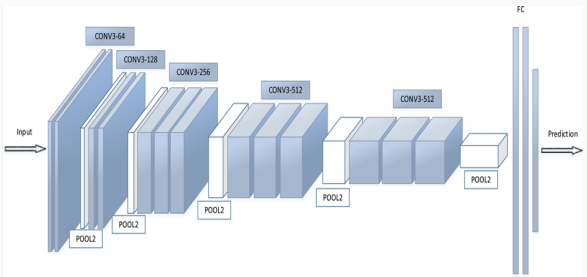
Input

Network's prediction

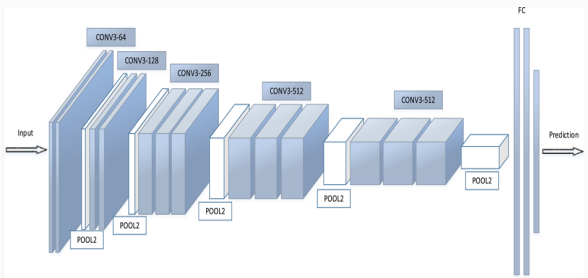


" This is a car !"

backpropagation to modify the pixels



changing the prediction

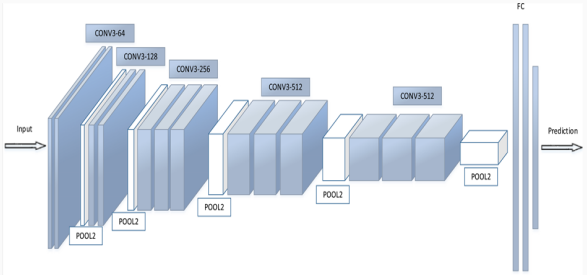


"This is a plane !"

A Simple Experiment: What really happened

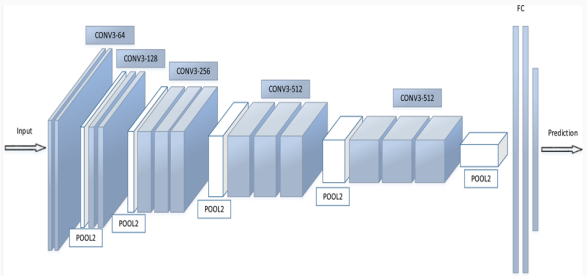
Input

Network's prediction

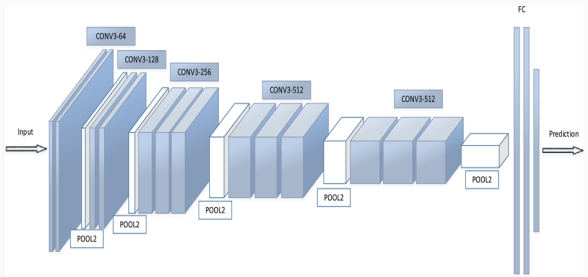


" This is a car !"

backpropagation to modify the pixels

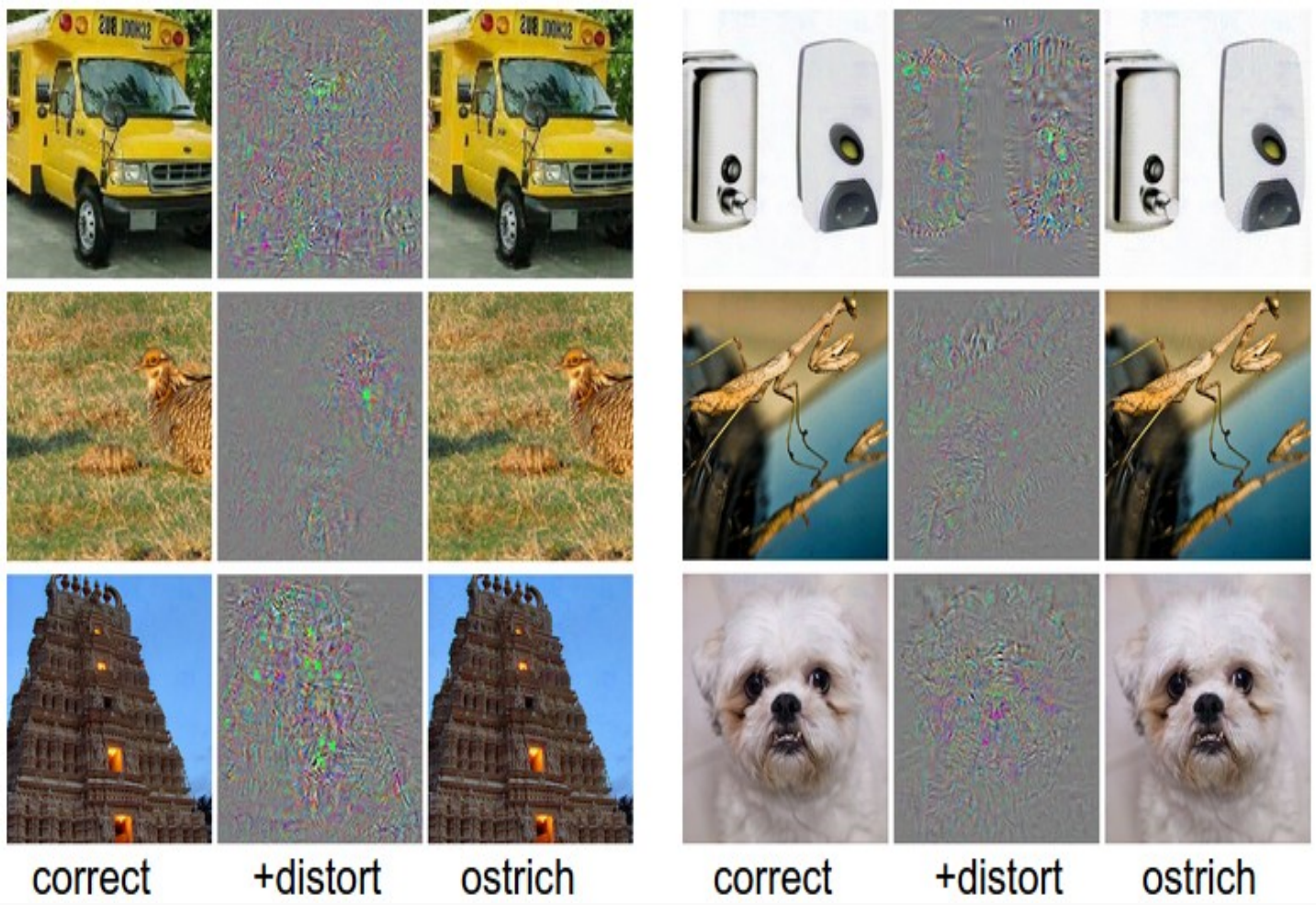


changing the prediction

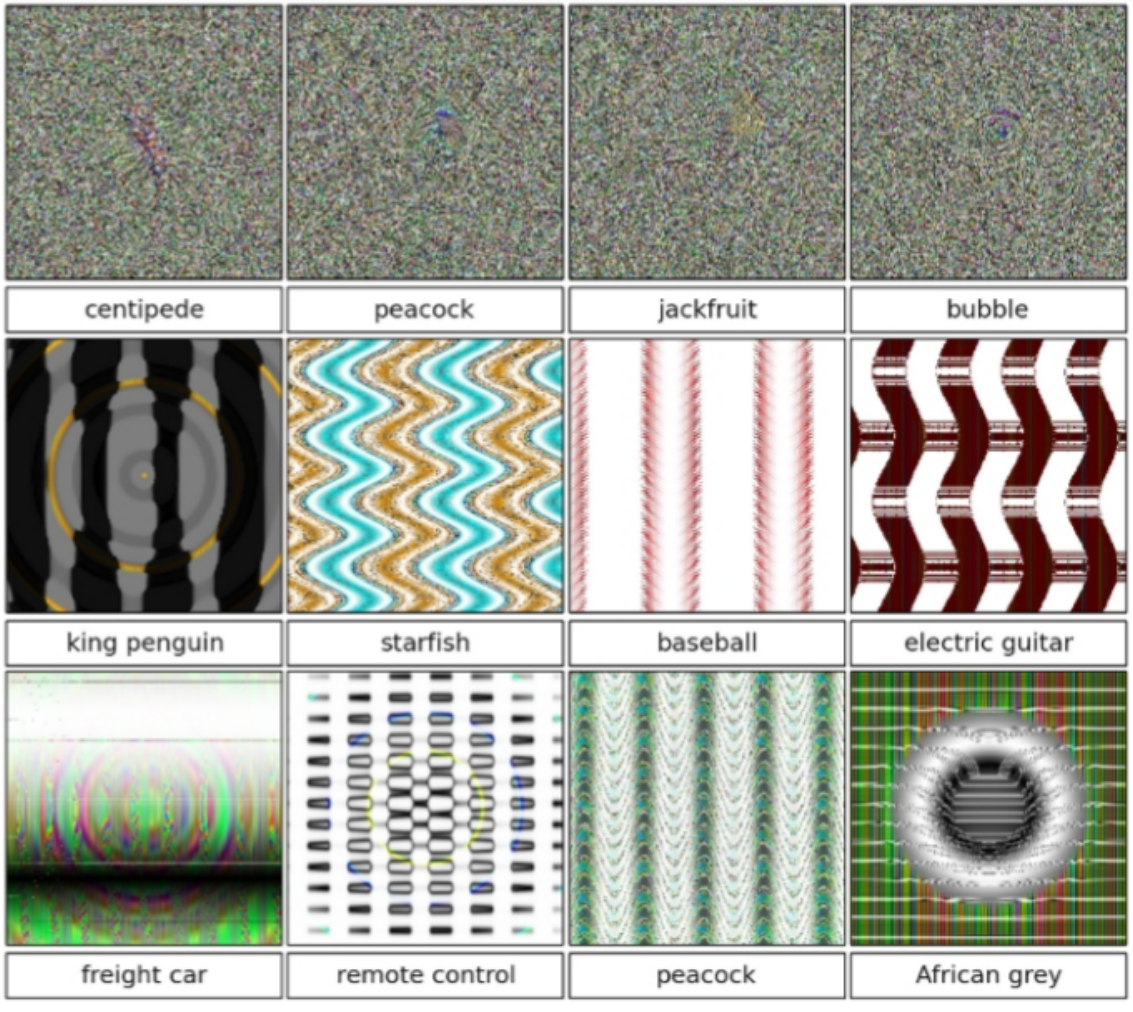


"This is a plane !"

Orienting mis-predictions



Pushing the "bouchon"



Confidence $\geqslant 96\%$

Definition: Adversarial Example

Definition: $\hat{x}$ is called adversarial iff:

- given image x
- low distortion $|| x - \hat{x} ||< \epsilon, \quad (\epsilon > 0$, few pixels)
- given network's probabilities $f_\theta(x)$
- Different predictions! $argmax f_\theta(x) \neq argmax f_\theta(\hat{x})$

Taking picture from adversarial images

Adversarial examples in the physical world from Alexey Kurakin, Ian Goodfellow, Samy Bengio, 2016
`https://youtu.be/zQ_uMenoBCk`

Adversarial objects

from `https://www.csail.mit.edu/news/fooling-neural-networks-w3d-printed-objects`

Adversarial sounds

from `https://nicholas.carlini.com/code/audio_adversarial_examples/`