



Deep Learning - MAI

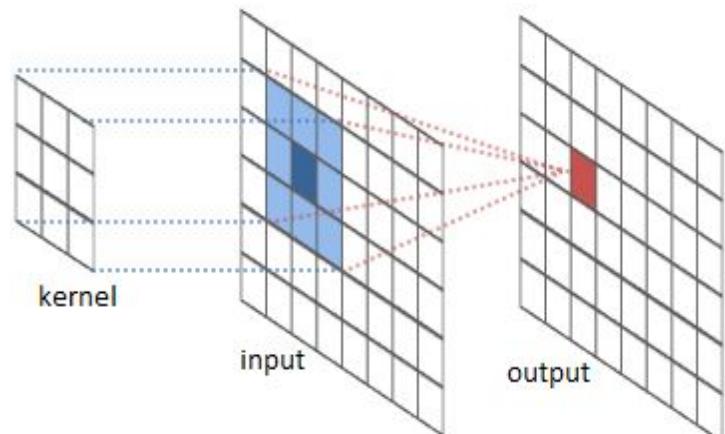
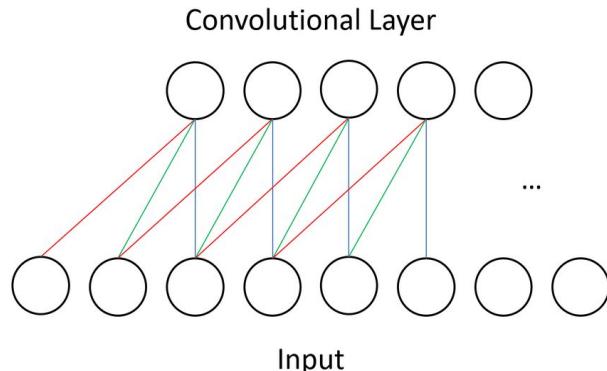
Convolutional neural networks

THEORY

Dario Garcia Gasulla
dario.garcia@bsc.es

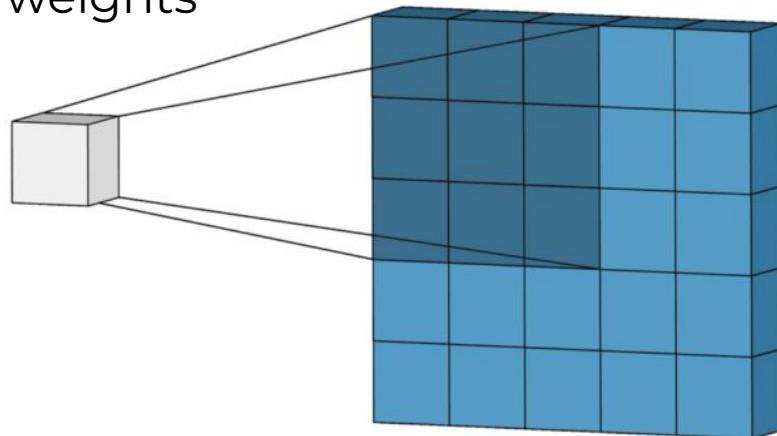
Spatial Connectivity

- ❖ Some data has spatial correlations that can be exploited
 - 1D, 2D, 3D, ...
- ❖ Near-by data points are more relevant than far-away.
- ❖ Sparsify connectivity to reduce complexity and ease the learning



Weight Sharing

- ❖ Sparse connectivity is nice, but we want to apply filters everywhere.
- ❖ Each filter will get convolved all over the image: 2D activations matrix
- ❖ In static we have sets of neurons sharing weights
- ❖ In this context, what is a neuron?



Convolution in Action

Kernel size 3x3
(neuron input = 9)

1 0 1
0 1 0
1 0 1

Detect 'X'

1 <small>$\times 1$</small>	1 <small>$\times 0$</small>	1 <small>$\times 1$</small>	0	0
0 <small>$\times 0$</small>	1 <small>$\times 1$</small>	1 <small>$\times 0$</small>	1	0
0 <small>$\times 1$</small>	0 <small>$\times 0$</small>	1 <small>$\times 1$</small>	1	1
0	0	1	1	0
0	1	1	0	0

Image

Filter convolution
process

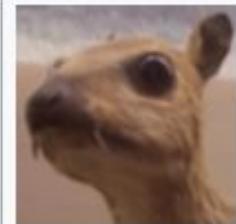
4		

Convolved
Feature

Activations (pre-func.)

Image Transformations

- ❖ Convolving filters transform the image
- ❖ Let the model learn the kernels it needs

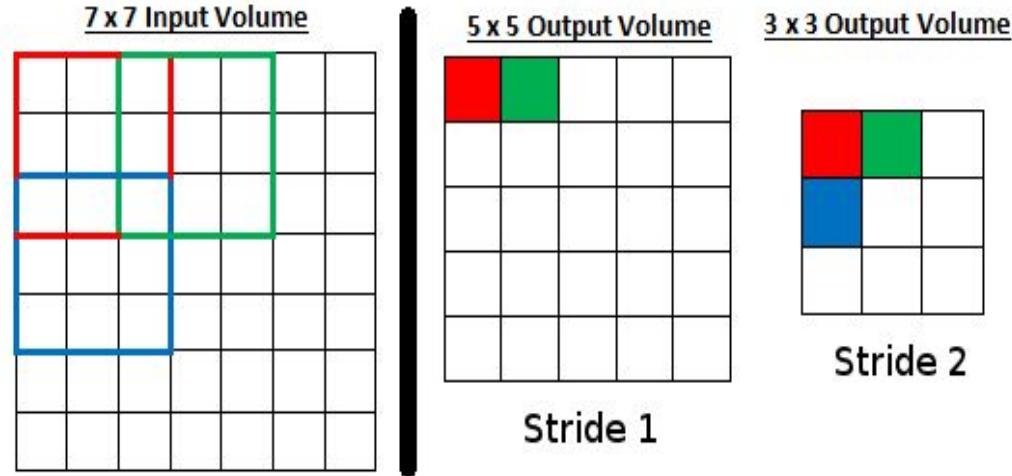
Edge detection	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Gaussian blur 3 × 3	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Convolution Details

Kernel size: Size of the receptive field of convolutional neurons

Stride: Steps size of convolution

Padding: Allows focus on border

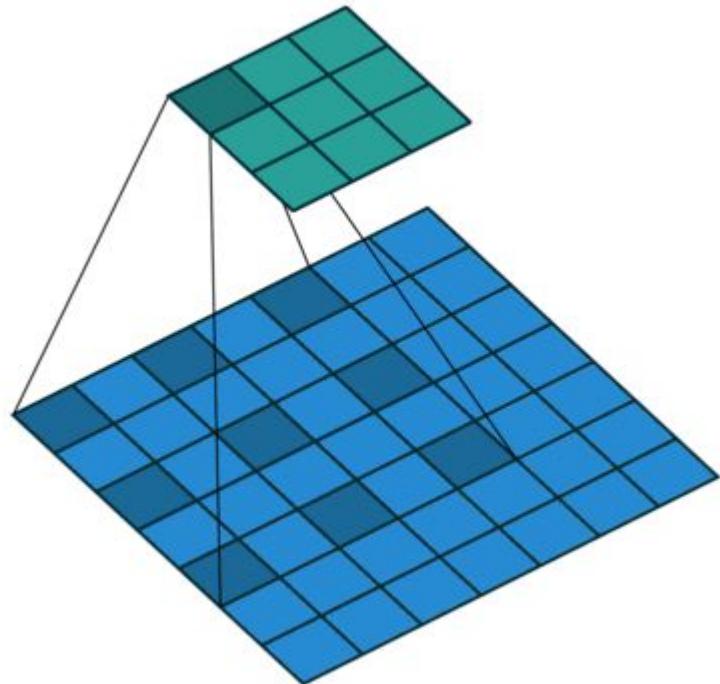


$$\text{OutputSize} = \frac{\text{InputSize} - \text{KernelSize} + 2 * \text{Padding}}{\text{Stride}} + 1$$

Dilated/Atrous Convolutions

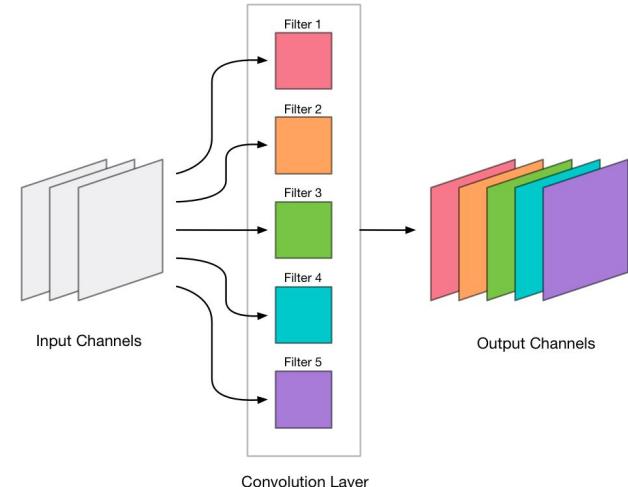
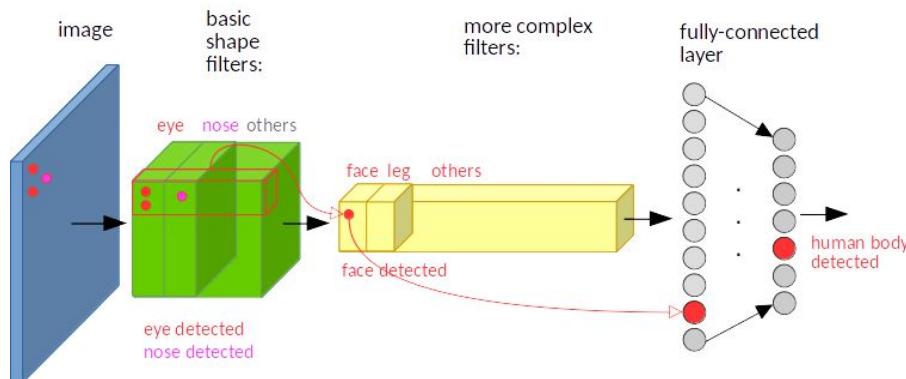
Sparsify the kernel

- ❖ Increases perceptive field without added complexity
- ❖ Loses details, gains context
- ❖ Another hyperparam :(
- ❖ Used for
 - Down/Upsampling (segmentation)
 - High Resolution inputs

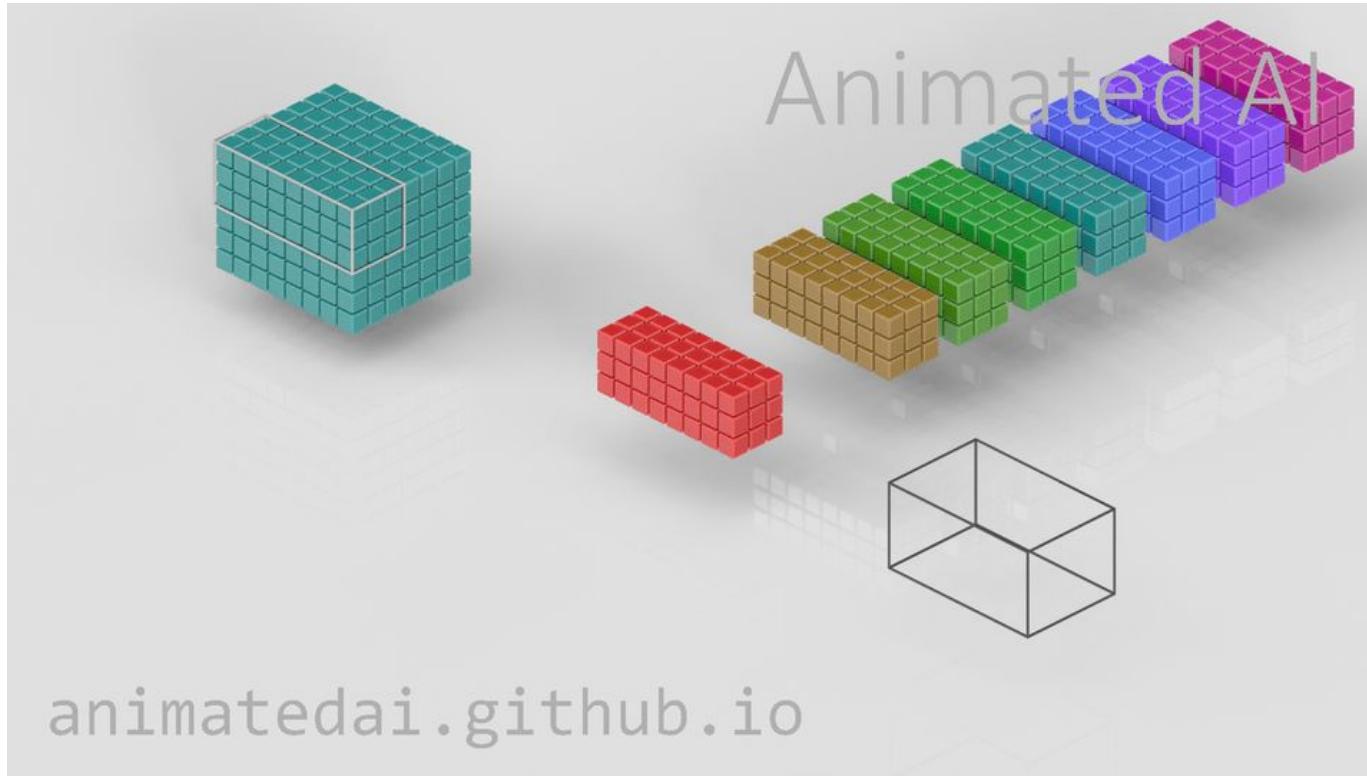


Output Volumes

- ❖ Typically, conv filters are full depth ($N \times N \times \text{input_depth}$)
- ❖ Each conv filter (often 3D) convolved generates a 2D plane of data
- ❖ Depth provides all the views on a part of the input
- ❖ Output volume: New representation of input with different dimensions



Output Volumes



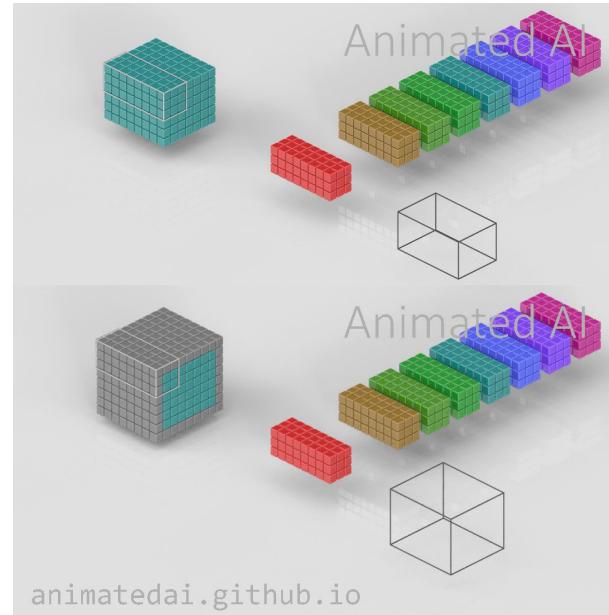
Padding policies

❖ Size

- Valid (no padding): Internal only. May skip data. Reduces dims.
- Same: Keep dimensionality with stride 1

❖ Filling

- Zeros
- Reflect
- Circular
- ...

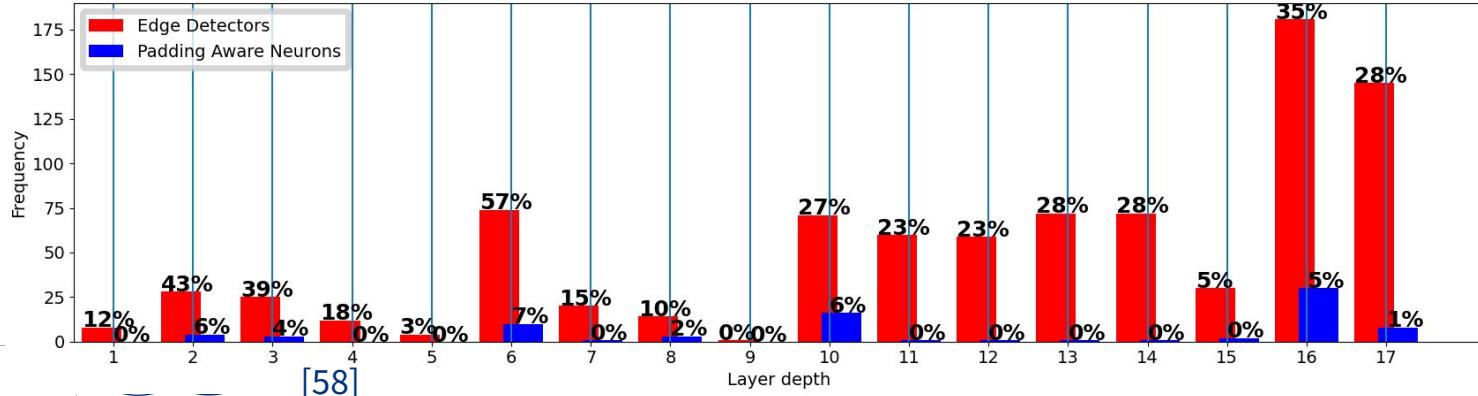
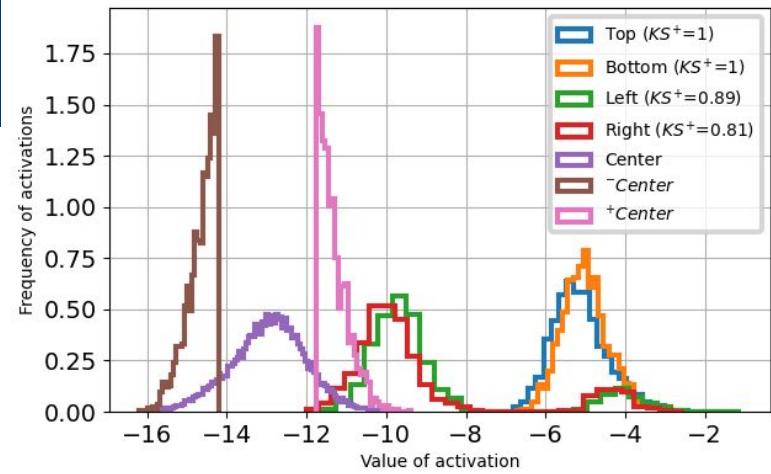
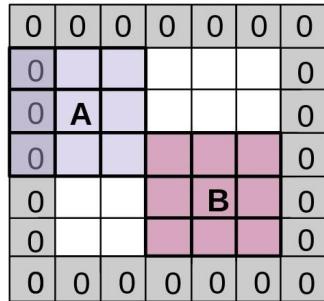


PANs

Too much bias

$$\begin{bmatrix} -2 & 1 & 1 \\ -2 & 1 & 1 \\ -2 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} -3 & 2 & -1 \\ -3 & 2 & -1 \\ -3 & 2 & -1 \end{bmatrix}$$

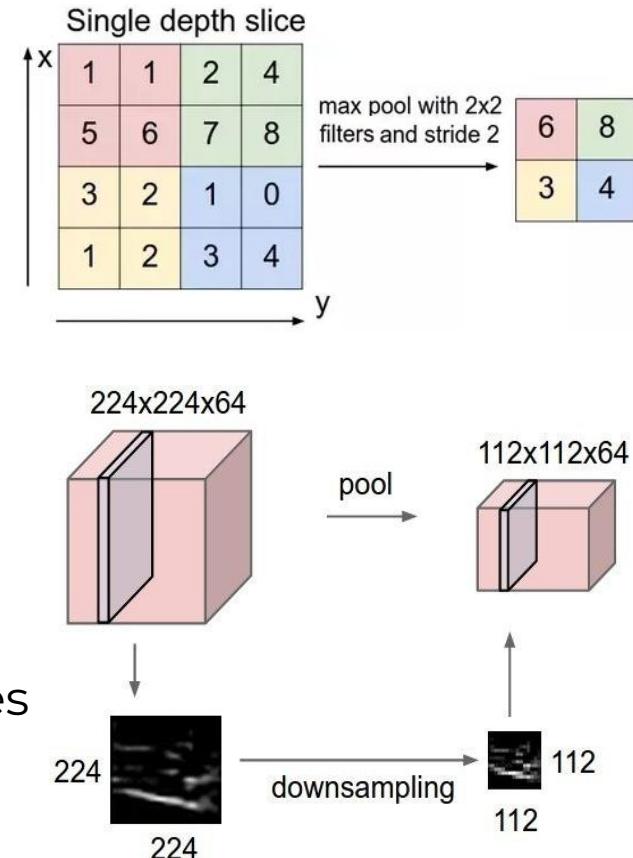


To Pool Or Not To Pool

- ❖ Operation: **Max** or Avg
- ❖ Dimensionality reduction (along x and y only)
- ❖ Rarely applied full depth
- ❖ Parameter free layer
- ❖ Hyperparams: Size & Stride
- ❖ Loss in spatial precision / Robust to invariance

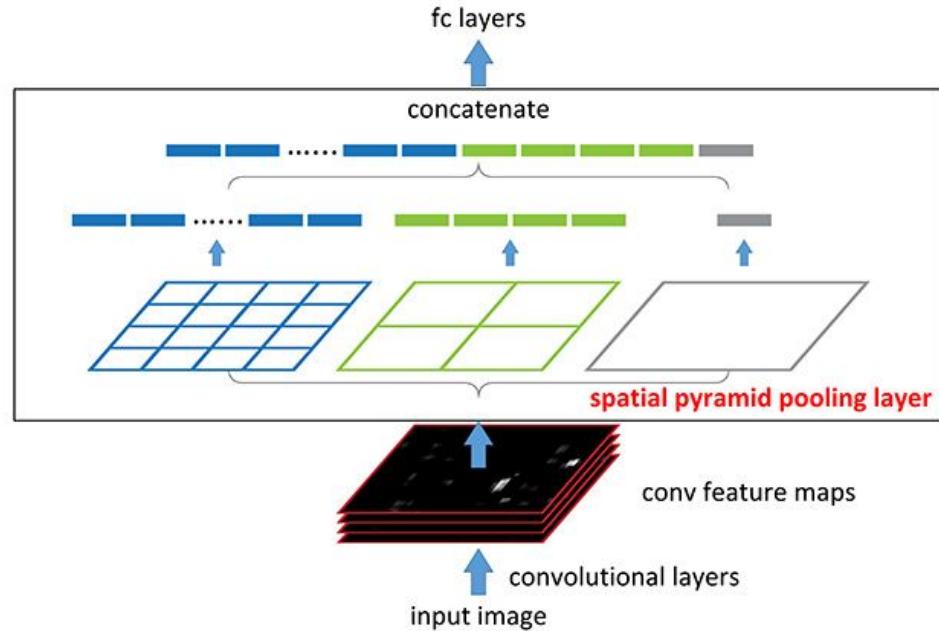
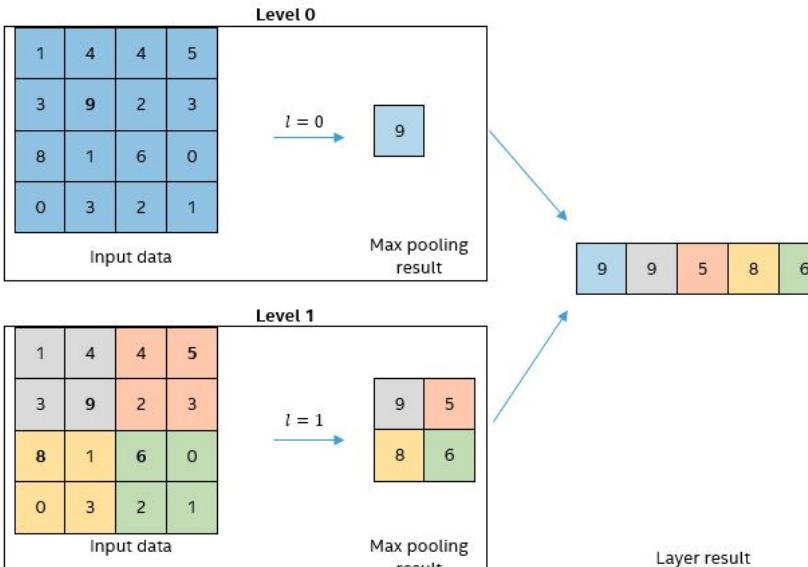
Other means to reduce complexity

- ❖ Depth-wise separable convs, bigger conv. strides



Spatial Pyramid Pooling (SPP)

- ❖ Multi-scale Pool (by powers of 2)
- ❖ Often used between conv and fc



More alternatives: Atrous spatial PP, Global average pooling, Pyramid pooling module, Adaptive PP

Practical Tips XI

Convolutional

- Small/big filters (3x3, 5x5, 7x7)
 - Cheap/Expensive
 - Local/General
 - Bigger/Smaller outputs (stride)
- Kernel Size = input size: fc
- Kernel size = 1x1: Alter depth)

Pooling

- 2x2, stride 1 is the least invasive

Hyperparameters incomplete list #4

- Kernel size (conv & pool)
- Stride (conv & pool)
- Padding (conv & pool)
- Num. filters
- Dilatation rate





CNNs

Emerging regularizers

Dario Garcia Gasulla
dario.garcia@bsc.es

Data Augmentation for CNNs

Apply what is safe for each case

- ❖ Problem specific
- ❖ Limited impact
- ❖ Computation
- ❖ Train/Val/Test

Geometry based



Color based



Noise / occlusion



Weather



Advanced image regularization/augmentation

Increase train variance forcing attention on full input (adds noise)

- ❖ MixUp (merge two samples), AdaMixup (manifold intrusion)
- ❖ CutOut (remove a patch)
- ❖ CutMix (merge samples w/ patch)
- ❖ Auto/DeepAugment (learn $\langle op., mag. \rangle$ from the data. Danger!)



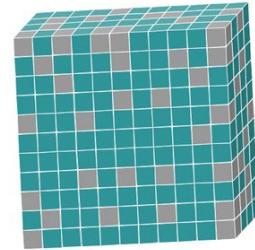
Beware. **More data** is always better than more augmentation.

[40,41,42,43,44,50]

Spatial Dropout

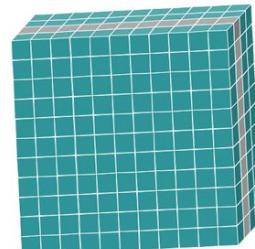
Standard Dropout is suboptimal for spatially related data

- ❖ Consecutive inputs can be strongly redundant



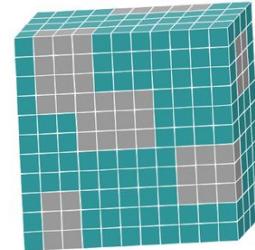
Spatial Dropout

- ❖ Drop entire feature maps, aka channels



Cutout

- ❖ Drop connected components along width, height and/or depth



Noisy Student (not only for CNNs)

A semi-supervised training paradigm

1. Train model A (teacher) with the labeled data
 2. Use A to generate pseudo-labels for an unlabeled data set
 3. Train model B (student) with both labeled and pseudo-labeled data
-
- ❖ Iterate, re-labeling the unlabeled data each time
 - ❖ Highly regularized (noise!) student to guarantee improvement
 - ❖ Each student has more capacity than the previous



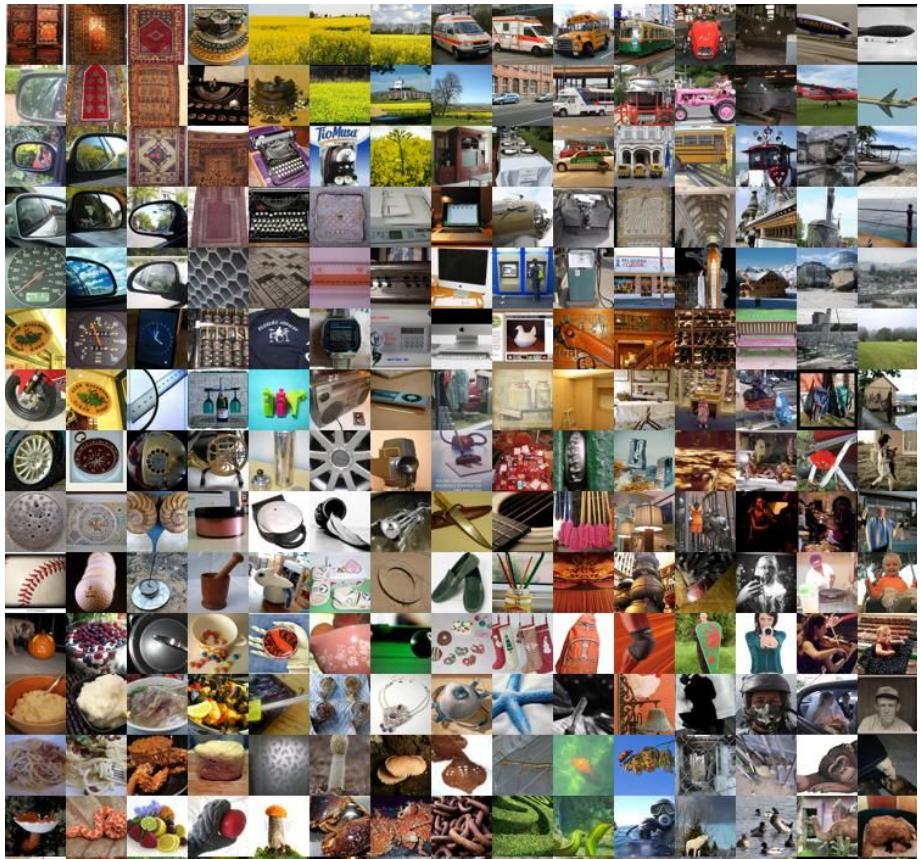
CNNs

Architectures

Dario Garcia Gasulla
dario.garcia@bsc.es

ILSVRC'12 (aka “Imagenet”)

- ❖ Classification: 1K classes
- ❖ Train: 1.2M, Val: 50K

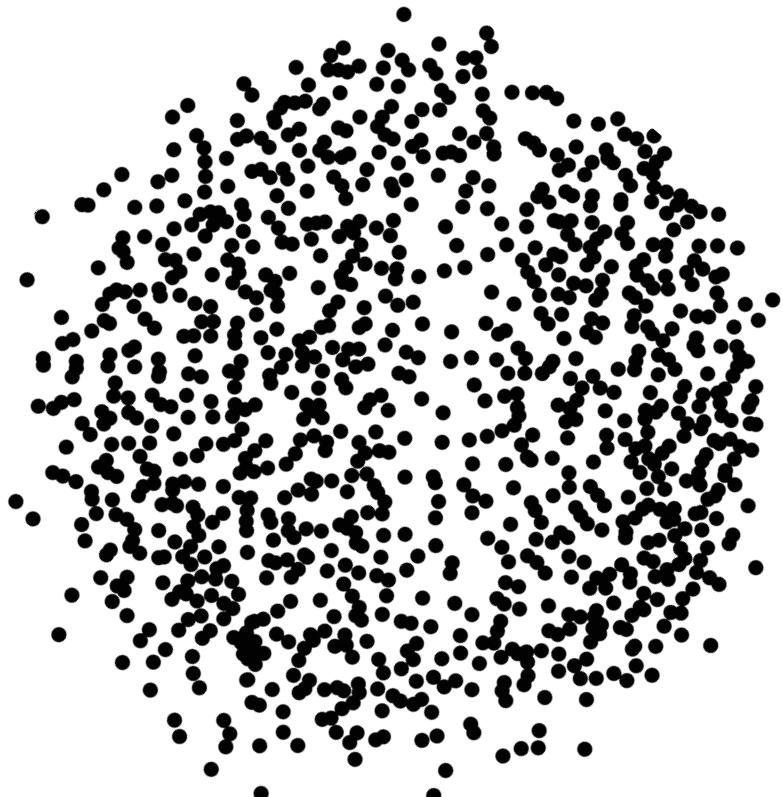


ImageNet limitations

- ❖ Noisy
 - Multiclass
 - Wrong (~6%)
- ❖ Overkilled
 - 90% pruning -> 3% perf. loss
- ❖ Overused
 - -10% performance on new test set

ILSVRC'12 (aka “Imagenet”)

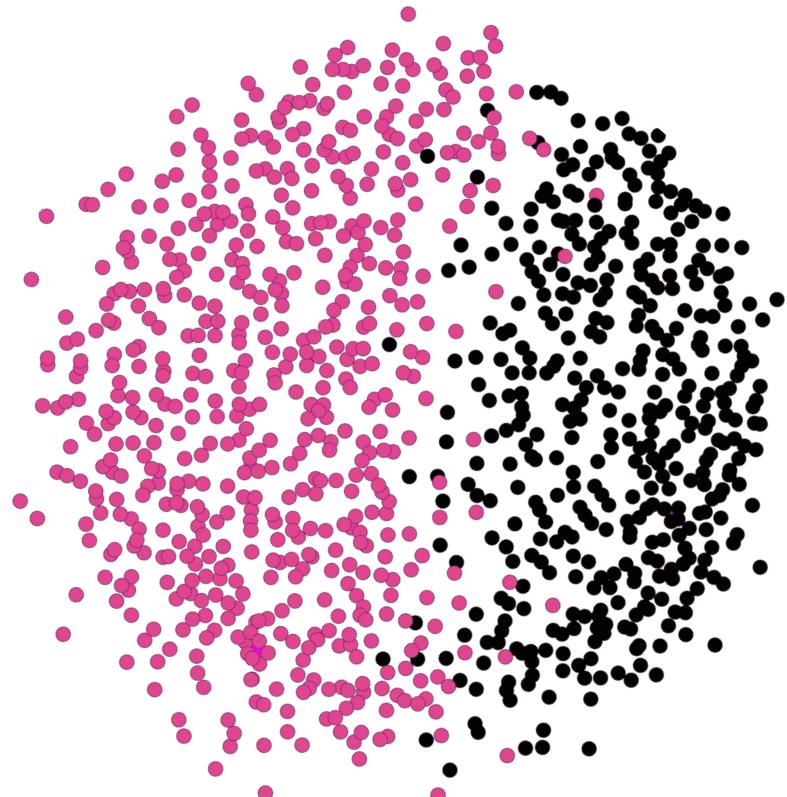
- ❖ Classification: 1K classes
 - Distances among internal representations



ILSVRC'12 (aka “Imagenet”)

- ❖ Classification: 1K classes
 - Distances among internal representations

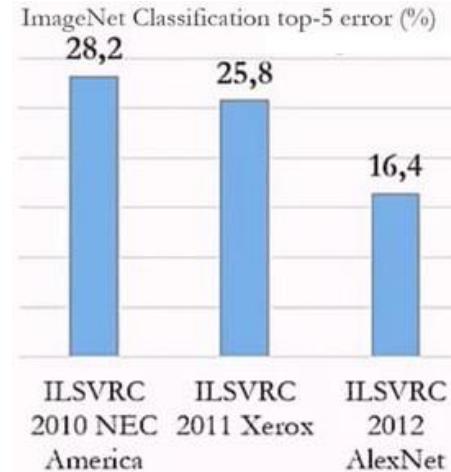
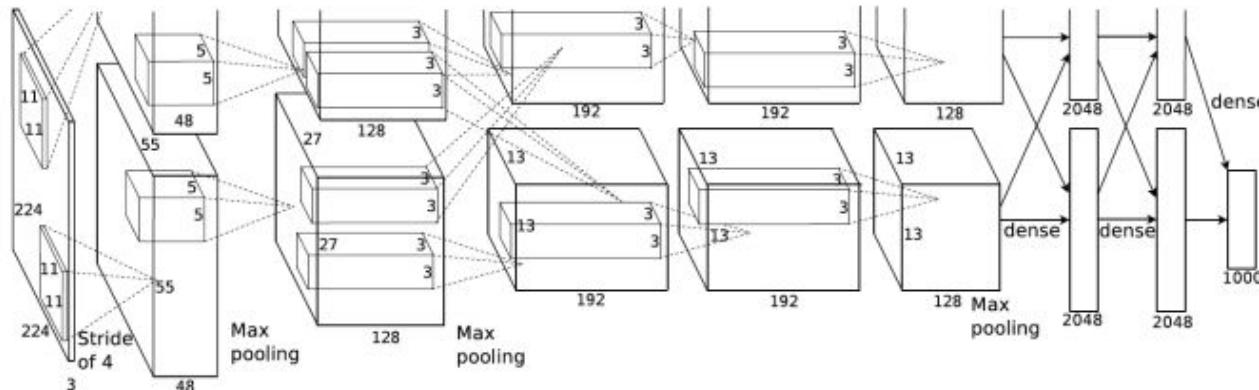
Man-made things Living things



CNNs Big Bang

AlexNet (2012)

- ❖ Breakthrough in ILSVRC
- ❖ 5 convs+ pools, ReLU, 2 dense, and dropout
- ❖ 62M parameters

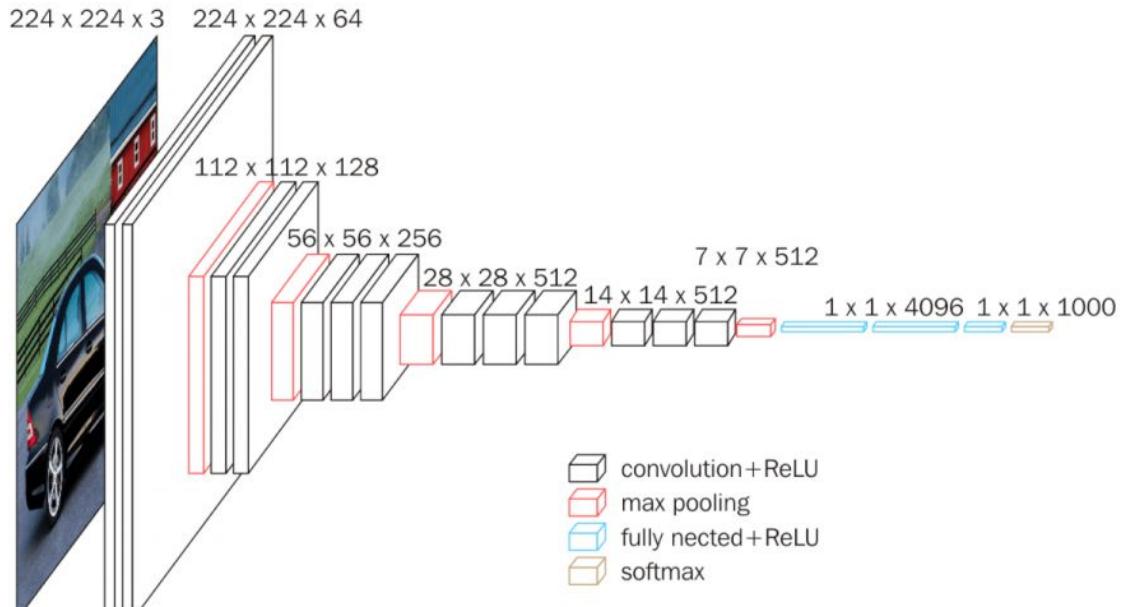
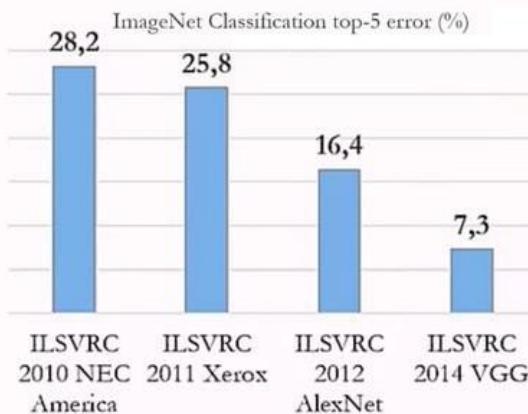


On the shoulders of giants

Optimizing cp*f

VGG 11/13/16/19 (2014)

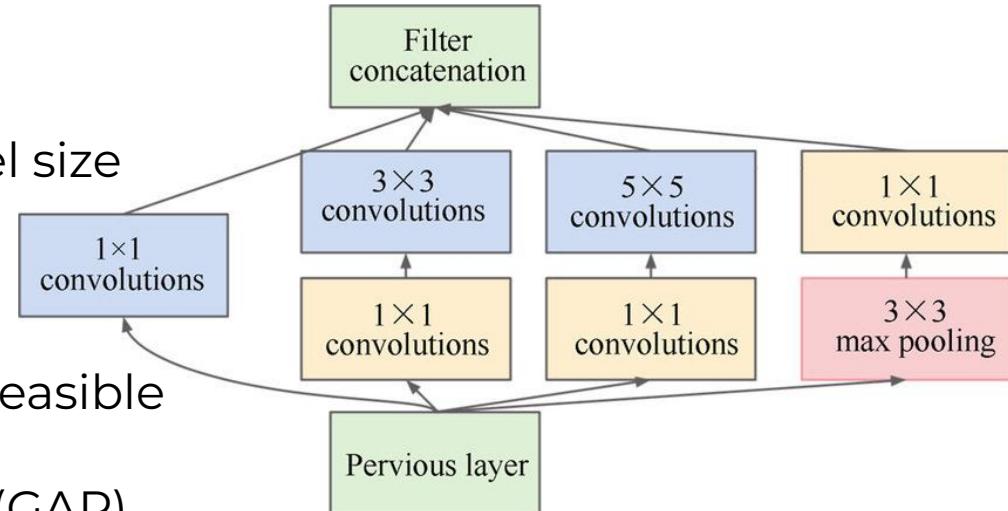
- ❖ Prototype of (conv-pool)*+dense* architecture
- ❖ 133-144M parameters
- ❖ 3x3 convs only



The Inception Family

GoogLeNet (2014)

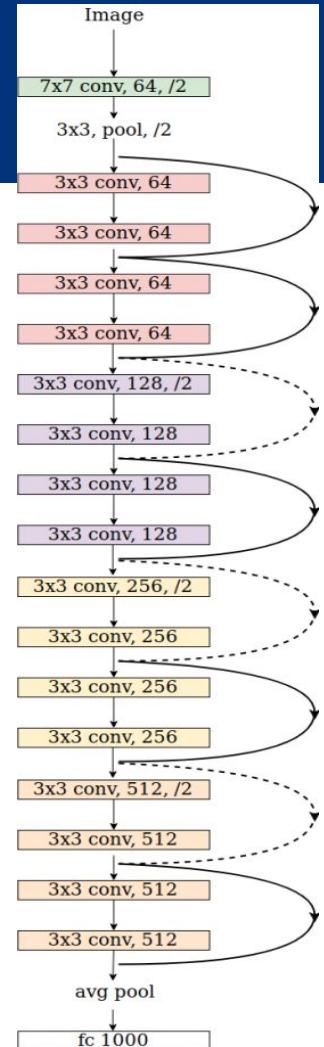
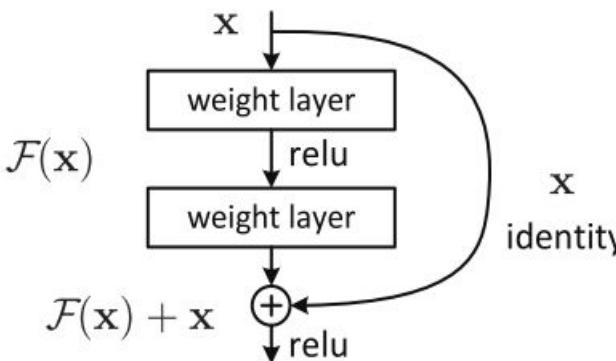
- ❖ The Inception block
- ❖ Let the model decide the kernel size
- ❖ Better scale adaptation
- ❖ Bottleneck 1x1 conv to make it feasible
- ❖ No FC: Global Average Pooling (GAP)



The Skipped Connection

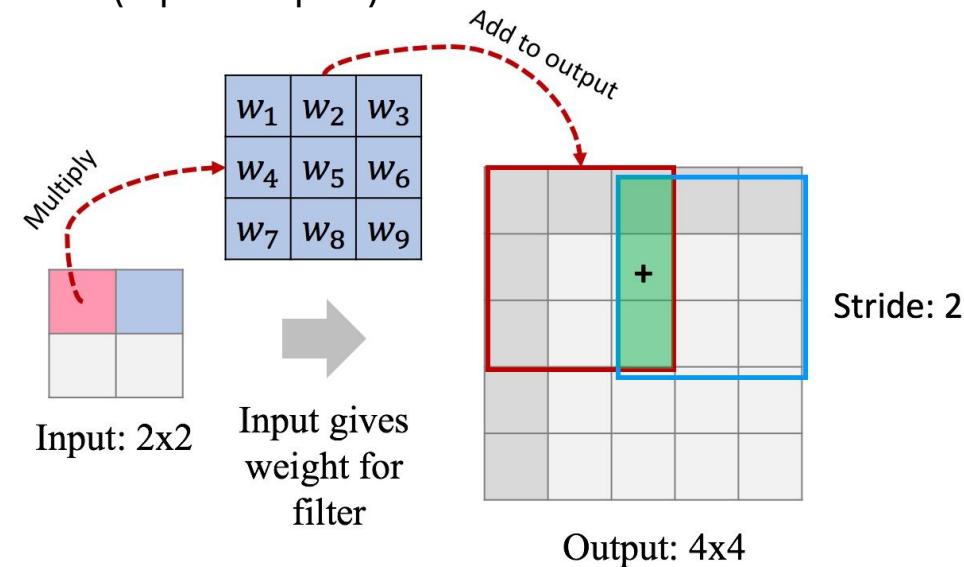
ResNet (2015)

- ❖ Residual blocks / Skip connections
- ❖ Deeper should never be worse
 - Learning the identity is hard
 - Learning to cancel out is easy
- ❖ Shallow ensemble of nets
- ❖ Train up to 1K layers (do not!)
- ❖ ILSVRC'12 human level



Transposed Convolution Deconvolution

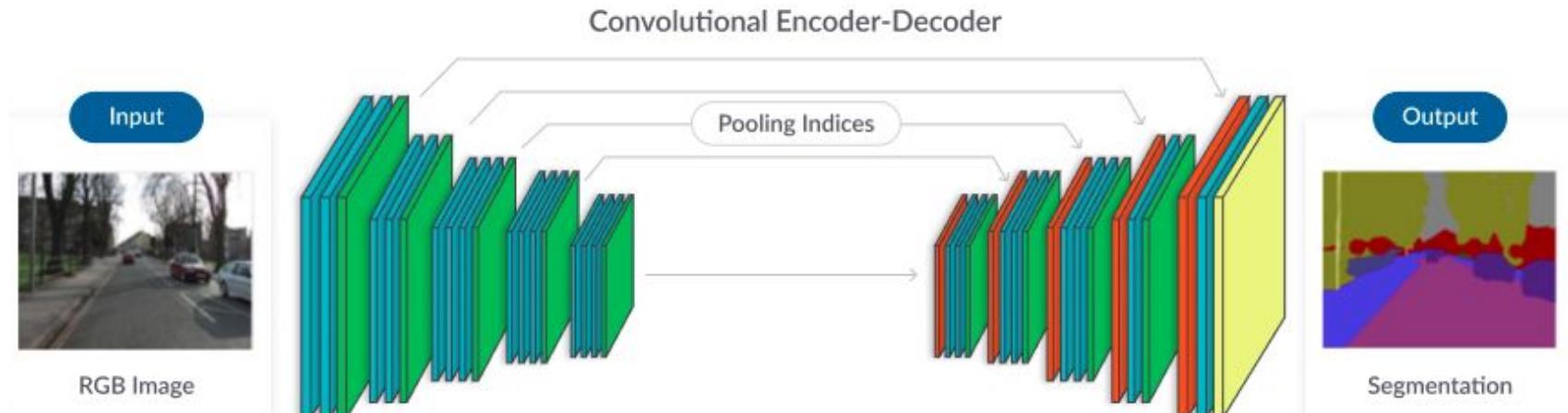
- ❖ Reverse effect of regular convolution (upsample)
- ❖ Learnt interpolation
- ❖ Applications
 - Segmentation
 - GANs
 - Super-Resolution
 - Conv. Autoencoders



Input	Kernel	Output
$\begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 4 & 6 \\ 4 & 12 & 9 \end{bmatrix}$
	$=$	$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} + \begin{bmatrix} 0 & 2 \\ 4 & 6 \end{bmatrix} + \begin{bmatrix} 0 & 3 \\ 6 & 9 \end{bmatrix}$

Encoder-Decoder aka Bottleneck

- ❖ Pixel-wise classification task (image reconstruction loss)
- ❖ Bottlenecking makes it cheaper



● Conv + Batch Normalisation + ReLU

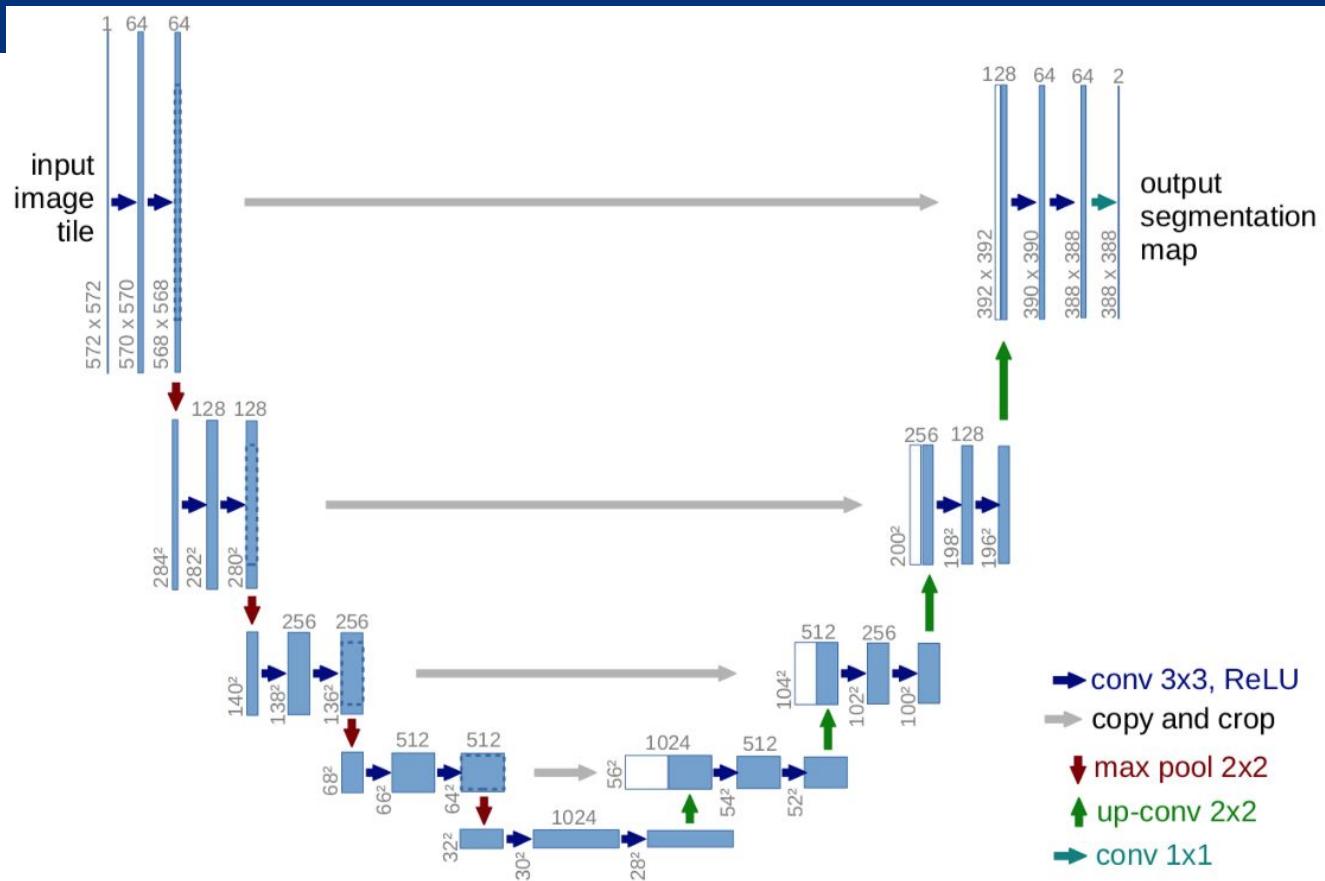
● Pooling

● Upsampling

● Softmax

A standard

❖ U-Net



Depth-wise Separable Convolutions

- ❖ Depth-wise convolutions (spatial)

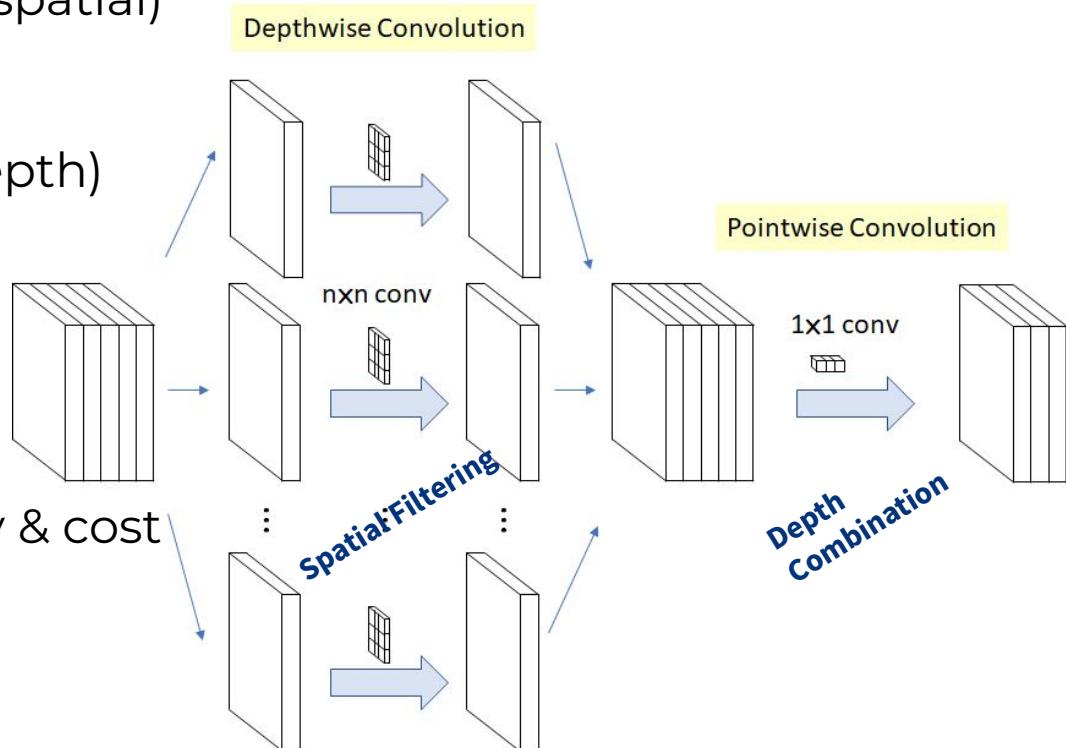
- Filters: $N \times N \times 1$

- ❖ Point-wise convolution (depth)

- Filters: $1 \times 1 \times \text{input_depth}$

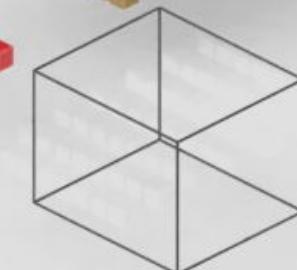
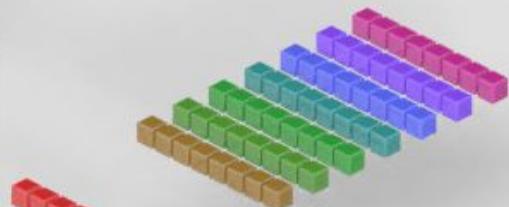
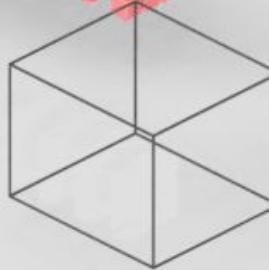
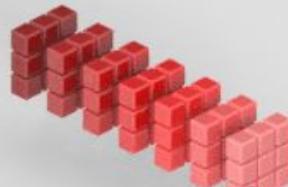
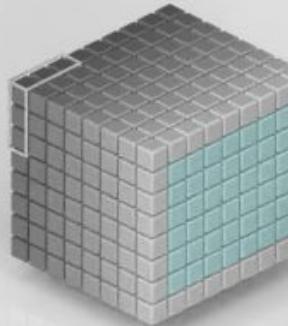
- ❖ Params: $N \times N + N$

- Decrease in complexity & cost



Depth-wise Separable Convolutions

Animated AI



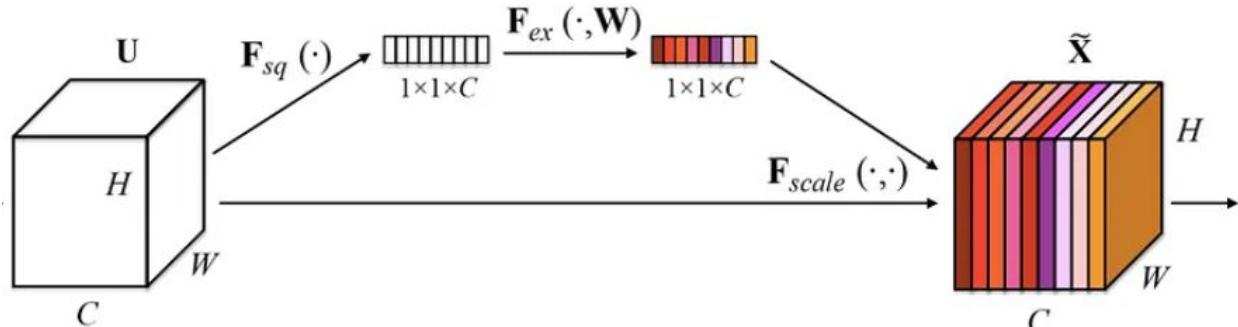
Notice the two
types of 2D conv
filters

animatedai.github.io

Squeeze & Excite

- ❖ Increase/Decrease channel depth
- ❖ Non-spatial
- ❖ Parameter efficient

1. GAP
2. FC-ReLU
3. FC-Sigmoid
4. Channel-wise weight product



Inverted Residuals

1. Point-wise conv

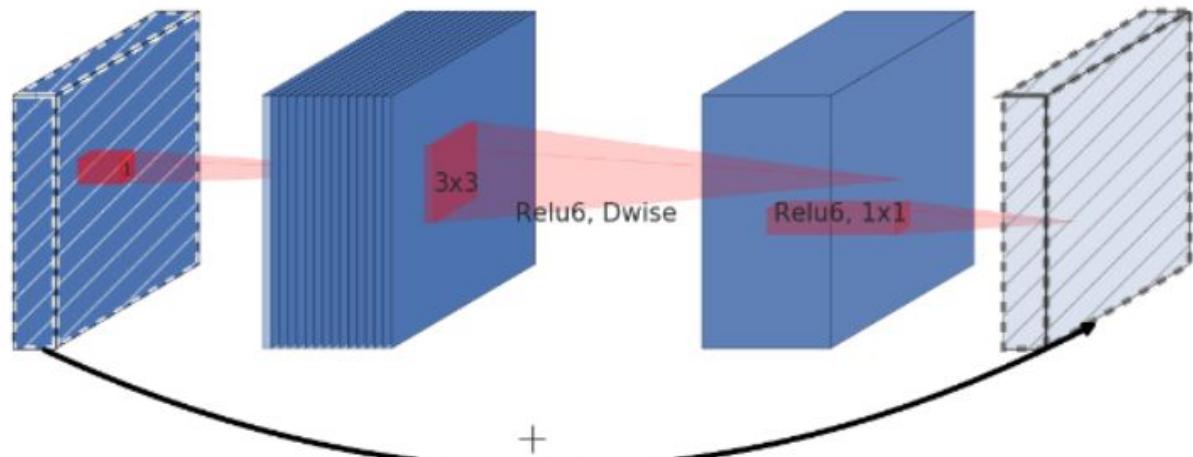
- Channel compute
- +Depth

2. Depth-wise conv

- Spatial compute

3. Point-wise conv

- Channel compute
- -Depth



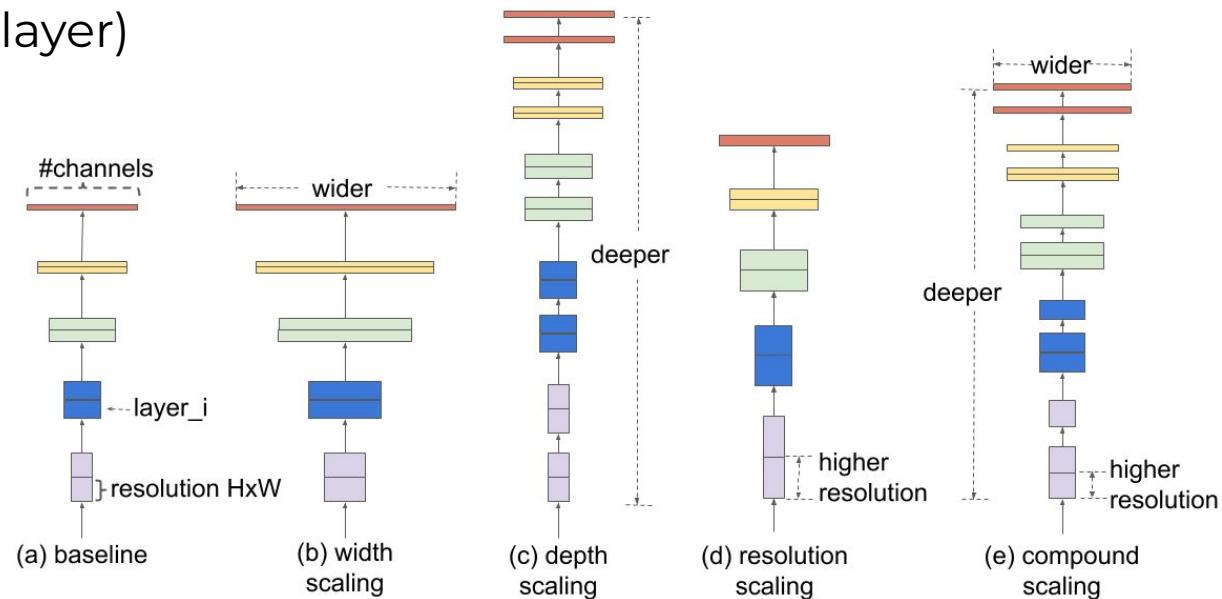
- ❖ Residual link
- ❖ Linear bottleneck (//)
- ❖ Efficient

EfficientNet

Should I go deeper, wider or bigger?

- ❖ Find a balance between them (all related)

- Width (neurons per layer)
- Depth (layers)
- Resolution (input)
- B0 to B7
- Inverted Res. Blocks



ConvNext, transforming CNNs

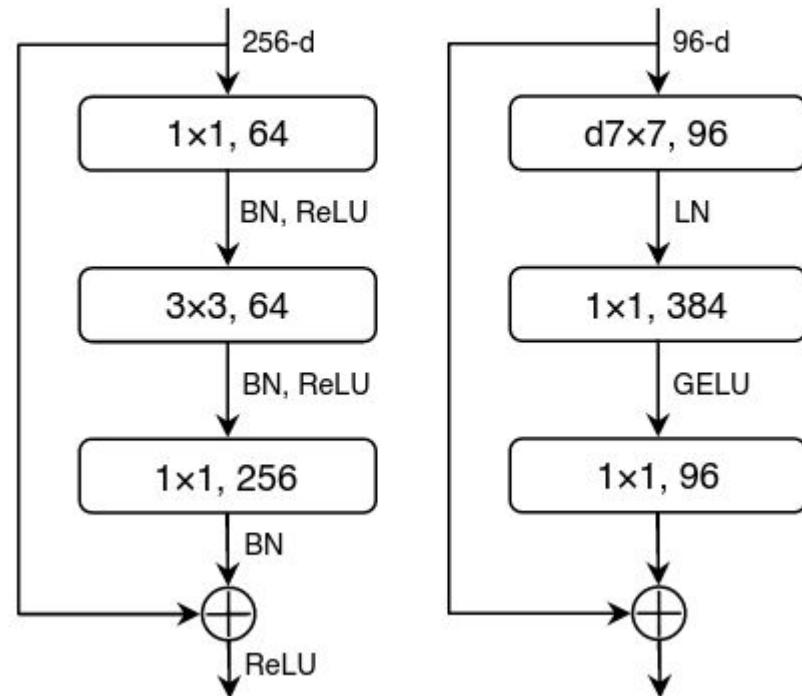
ViT learnt from CNN (Swin Transformer)

- ❖ AdamW (L2 regularization after step computation. Safe.)
- ❖ Regularize: Data augmentation (MixUp, Cutmix, ...), Label smoothing, ...
- ❖ Compute distribution (pool separated blocks): (3,4,6,3) -> (3,3,9,3)
- ❖ Patchify: First layer 4x4 stride 4 conv
- ❖ Depth-wise conv (spatial or channel mix). Inverted bottleneck.
- ❖ Larger kernels: 7x7
- ❖ GeLU, LN, BN

ConvNext, transforming CNNs

1. Patchify
2. Depth-wise conv
3. Inverted bottleneck
4. Larger kernels: 7×7
5. GeLU
6. Less activation functions
7. LN instead of BN
8. Less normalization layers

ResNet Block **ConvNeXt Block**



Practical Tips XII

CNN design policies

- Few filters at the beginning
- Hierarchy
- Max. complexity 2/3ds in

Things to monitor, layer wise

- Volume sizes
- Num. parameters



Visualizing CNNs

Biases everywhere

The Basics

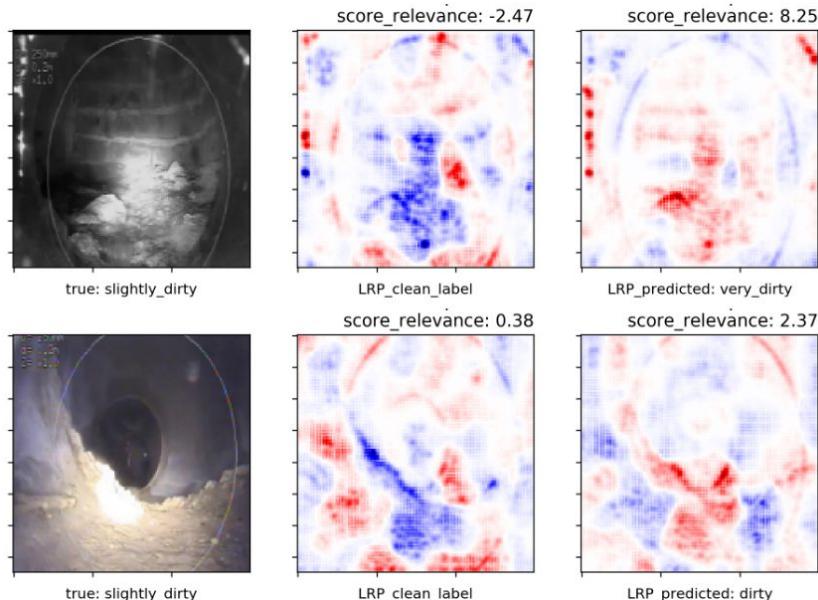
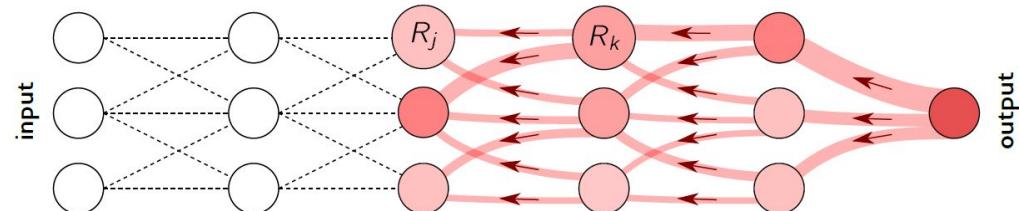
- ❖ NN are representation learning techniques
- ❖ CNNs build hierarchically complex features
 - From Gabor filters to dog faces
 - Induced by convolution
 - Tend to focus on the “non obvious for humans”
 - Backgrounds, textures
- ❖ The closer to the loss, more classifier (task) and less representation (data)

Ways of Looking at CNNs

- ❖ *Feature Attribution:* **Where** is the network looking?
 - Grounded. Instance based.
 - Explainability in practice.
- ❖ *Feature Visualization:* **What** is the network seeing?
 - Uncontextualized. Maximization based.
 - Diagnosys & Insight
- ❖ *Exemplification:* **How** does the network react?
 - Max. activations
 - Samples from a distribution

Attribution (Where)

- ❖ Finding the importance of pixels
- ❖ Layerwise Relevance Propagation (LRP)
 - Backpropagate an output. Find the relevance of each neuron
 - Weighted by CNN parameters



Feature Visualization (What)

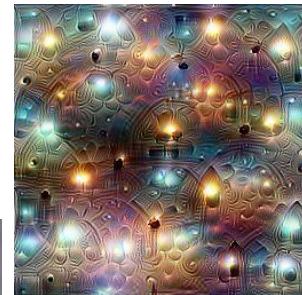
- ❖ Optimizing the input to maximize the output

- A neuron



Low level

- A channel



High level

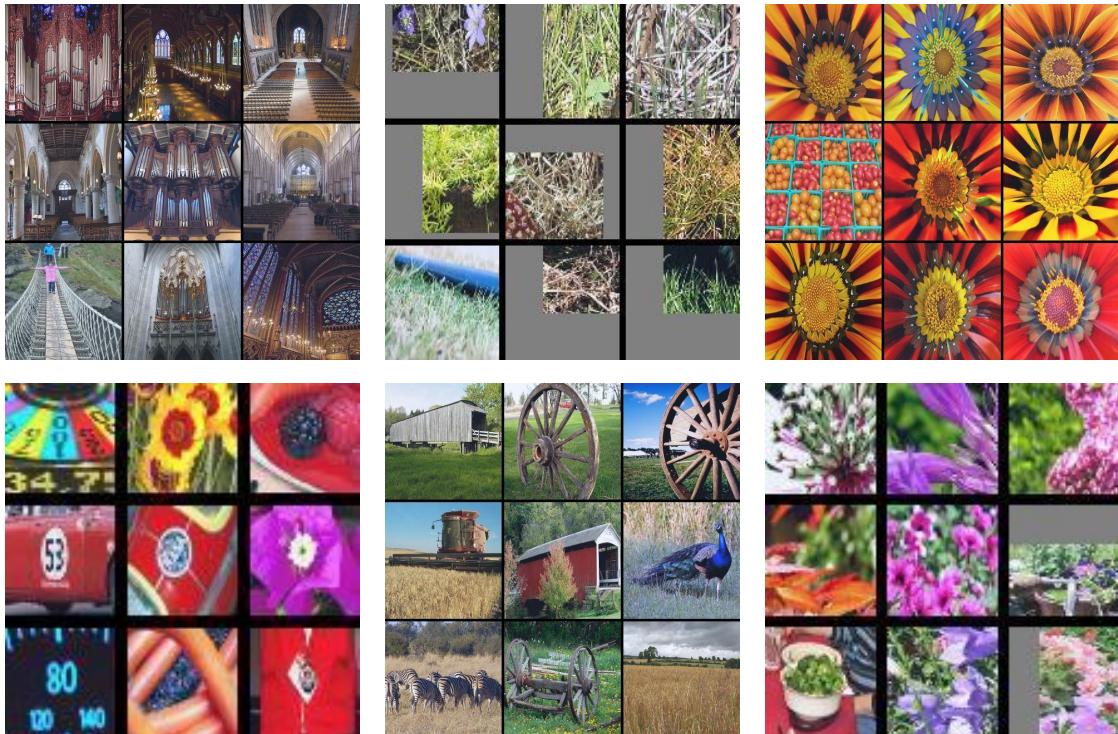
- A layer (DeepDream)



Exemplification (How)

- ❖ Finding images within a dataset maximizing outputs

- Subjective
- Partial
- Stochastic



Bias in DL

“All models are wrong, some are useful” - George Box

-

“All DL models are biased, some are usefully biased”

Bias in DL

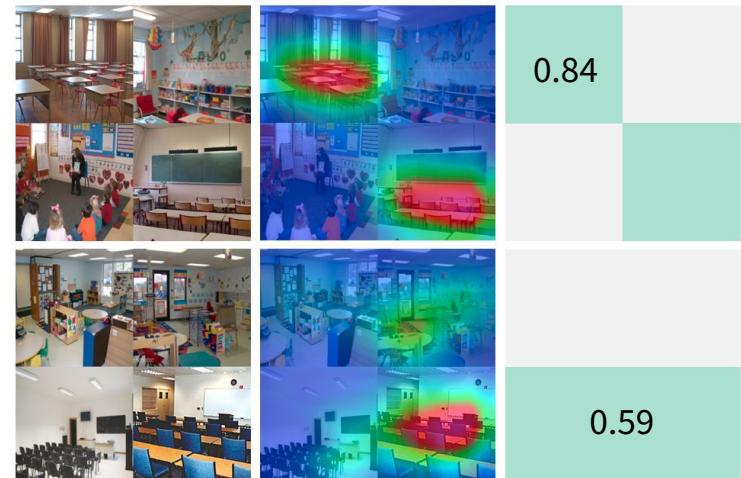
- ❖ Bias is what makes ML work. Is a form of generalization.
 - *Identification:* What bias?
 - Bonus track: Human bias (Pareidolia)
 - *Appreciation:* Desirable bias?
 - *Mitigation:* Altering dataset or model?

Bias Detection through XAI Attribution

Focus & Mosaics: An eye-tracking game

Why is this mosaic of class “cat”?

- ❖ Identification: Many examples needed
- ❖ Evaluation: Expert decision
- ❖ Mitigation:
 - Shared bias:
 - Add target samples without bias
 - Add non-target samples with bias
 - Missing bias: Add target samples with bias



Target class: Classroom
Outer class: Kindergarten



Playing with CNNs

Automatic Image Colorization

- ❖ Another pixel-wise classification application

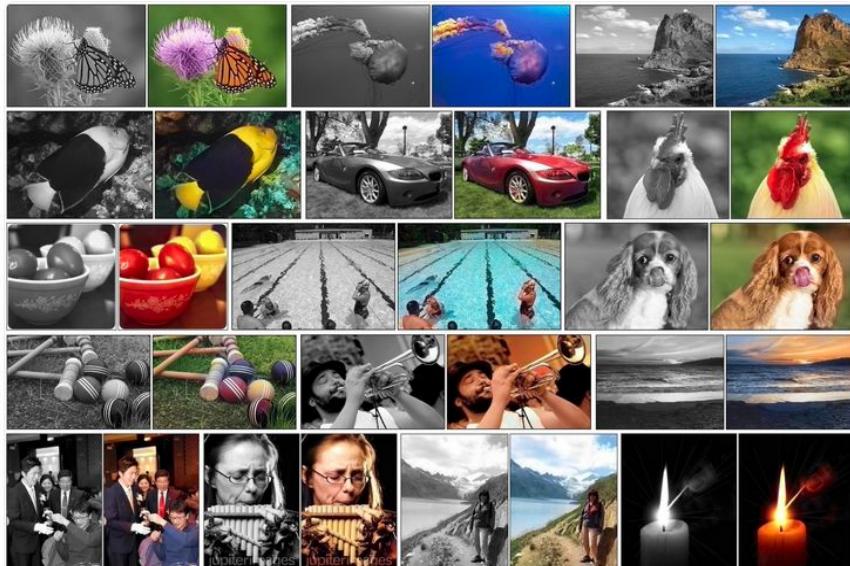


(a) Colorado National Park, 1941

(b) Textile Mill, June 1937

(c) Berry Field, June 1909

(d) Hamilton, 1936



Faster Segmentation

- ❖ ~~Pixel wise classification~~ Object detection (bounding box)
 - Can be done with a “regular” CNN
- ❖ R-CNN: Propose crops (SVM). Extract features (CNN). Classify crops (SVM)
- ❖ Fast R-CNN: Extract features. Propose crops. Classify/Bounding Box (CNN)
- ❖ Faster R-CNN: Propose crops through a specific sub-net (RPN)
- ❖ YOLO v? (no regions, faster, less accurate)
 - Divide into grid. Predict class and bounding box for each cell.

Better Segmentation

- ❖ Mask R-CNN

- Faster R-CNN for object detection
- FCN for instance segmentation (pixel classification)

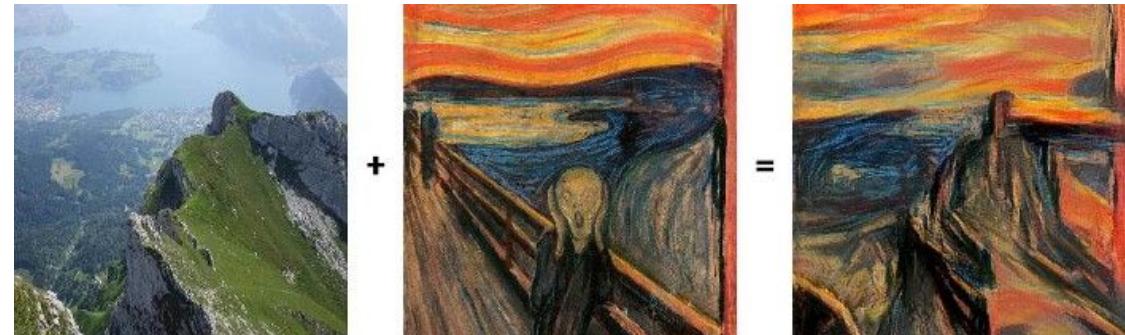
- ❖ Xception

- Depth-wise separable Convs (inverted order & w/o non-linearity)
- Skip connections
- Atrous SPP



Style Transfer

- ❖ What do the correlation of activations intra-layer tell us?
 - What if we force it on another image?
- ❖ Gram matrix represents the *style*
 - Channel-wise ($c \times c$)
 - Several mid layers
- ❖ Activations represents the *content*
 - One mid layer



- ❖ Optimize the **input** to minimize 2 losses
- ❖ Use a pre-trained net frozen
- ❖ Improved and extended

References

- [1] http://vordenker.de/ggphilosophy/mcculloch_a-logical-calculus.pdf
- [2] <http://www-public.tem-tsp.eu/~gibson/Teaching/Teaching-ReadingMaterial/Rosenblatt58.pdf>
- [3] <http://www.dtic.mil/dtic/tr/fulltext/u2/236965.pdf>
- [4] [https://en.wikipedia.org/wiki/Perceptrons_\(book\)](https://en.wikipedia.org/wiki/Perceptrons_(book))
- [5] <http://www.andreykurenkov.com/writing/a-brief-history-of-neural-nets-and-deep-learning/>
- [6] [https://en.wikipedia.org/wiki/Perceptrons_\(book\)](https://en.wikipedia.org/wiki/Perceptrons_(book))
- [7] Werbos et al. "Beyond regression:" new tools for prediction and analysis in the behavioral sciences." Ph. D. dissertation, Harvard University (1974).
- [8] Rummelhart et al. "Learning Internal Representations by Error Propagation". MIT Press (1986).

References

- [9] <https://towardsdatascience.com/effect-of-gradient-descent-optimizers-on-neural-network-training-d44678d27060>
- [10] <https://arxiv.org/abs/1711.05101>
- [11] <https://bbabenko.github.io/weight-decay/>
- [12] <https://towardsdatascience.com/weight-decay-l2-regularization-90a9e17713cd>
- [13] Veit, Andreas, Michael J. Wilber, and Serge Belongie. "Residual networks behave like ensembles of relatively shallow networks." Advances in neural information processing systems. 2016.
- [14] <https://thegradient.pub/semantic-segmentation/>
- [15] <https://arxiv.org/pdf/1603.08511.pdf>
- [16] <https://pdfs.semanticscholar.org/5c6a/0a8d993edf86846ac7c6be335fba244a59f8.pdf>

References

- [17] <https://arxiv.org/pdf/1606.00915.pdf>
- [18] <https://arxiv.org/pdf/1610.02357.pdf>
- [19]
https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf
- [20] <https://arxiv.org/abs/1603.08155>
- [21] <https://arxiv.org/abs/1603.03417>
- [22] <https://ai.googleblog.com/2016/10/supercharging-style-transfer.html>
- [23] <https://arxiv.org/pdf/1903.07291.pdf>
- [24] <http://nvidia-research-mingyuliu.com/gauGAN>
- [25] Geirhos, Robert, et al. "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness." arXiv preprint arXiv:1811.12231 (2018).

References

- [26] Beery, Sara, Grant Van Horn, and Pietro Perona. "Recognition in terra incognita." Proceedings of the European Conference on Computer Vision (ECCV). 2018.
- [27] <https://distill.pub/2017/feature-visualization/>
- [28] <https://distill.pub/2018/building-blocks/>
- [29] Montavon, Grégoire, et al. "Layer-wise relevance propagation: an overview." Explainable AI: interpreting, explaining and visualizing deep learning. Springer, Cham, 2019. 193-209.
- [30]
- <https://medium.com/machine-intelligence-report/how-do-neural-networks-work-57d1ab5337ce>
- [31] Hebb, D.O. (1949), The organization of behavior, New York: Wiley

References

- [32] Dauphin, Yann N., et al. "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization." Advances in neural information processing systems. 2014.
- [33] Ruder, Sebastian. "An overview of gradient descent optimization algorithms." arXiv preprint arXiv:1609.04747 (2016).
- [34] Viazovetskyi, Yuri, Vladimir Ivashkin, and Evgeny Kashin. 'StyleGAN2 Distillation for Feed-Forward Image Manipulation'. 7 March 2020. <http://arxiv.org/abs/2003.03581>.
- [35] https://medium.com/@jonathan_hui/gan-stylegan-stylegan2-479bdf256299
- [36] <https://www.justinpinkney.com/making-toonify/>
- [37] <http://chengao.vision/FGVC/files/FGVC.pdf>
- [38] Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

References

- [39] Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." arXiv preprint arXiv:1704.04861 (2017).
- [40] <https://arxiv.org/abs/1710.09412v2>
- [41] <https://arxiv.org/abs/1708.04552>
- [42] <https://arxiv.org/pdf/1905.04899.pdf>
- [43] <https://arxiv.org/abs/1809.02499>
- [44]
https://openaccess.thecvf.com/content_CVPR_2019/papers/Cubuk_AutoAugment_Learning_Augmentation_Strategies_From_Data_CVPR_2019_paper.pdf
- [45]
<https://towardsdatascience.com/12-main-dropout-methods-mathematical-and-visual-explanation-58cdc2112293>

References

- [46] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, Efficient object localization using convolutional networks
- [47] T. DeVries and G. W. Taylor, Improved regularization of convolutional neural networks with cutout
- [48] Arias-Duart, Anna, Ferran Parés, and Dario Garcia-Gasulla. "Focus! Rating XAI Methods and Finding Biases with Mosaics" arXiv preprint arXiv:2109.15035 (2021).
- [49] Recht, Benjamin, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. "Do imagenet classifiers generalize to imagenet?." In International Conference on Machine Learning, pp. 5389-5400. PMLR, 2019.
- [50] <https://blog.insightdatascience.com/automl-for-data-augmentation-e87cf692c366>
- [51] Chollet, François. "Xception: Deep learning with depthwise separable convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

References

- [52] Liu, Ze, et al. "Swin transformer: Hierarchical vision transformer using shifted windows." Proceedings of IEEE/CVF international conference on computer vision. 2021.
- [53] Liu, Zhuang, et al. "A convnet for the 2020s." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.
- [54] <https://towardsdatascience.com/why-adamw-matters-736223f31b5d>
- [55] Garcia-Gasulla, Dario, et al. "A visual embedding for the unsupervised extraction of abstract semantics." Cognitive Systems Research 42 (2017): 73-81.
- [56] Trevor Gale, Erich Elsen, and Sara Hooker. The state of sparsity in deep neural networks. CoRR, abs/1902.09574, 2019. URL <http://arxiv.org/abs/1902.09574>
- [57] Northcutt, Curtis G., Anish Athalye, and Jonas Mueller. "Pervasive label errors in test sets destabilize machine learning benchmarks." arXiv preprint arXiv:2103.14749 (2021).

References

- [58] Garcia-Gasulla, D., Gimenez Abalos, V., and Agustin Martin-Torres, P. "Padding Aware Neurons." In 4th Visual Inductive Priors for Data-Efficient Deep Learning Workshop. 2023.
- [59] <https://arxiv.org/abs/1709.01507>

Dario Garcia-Gasulla (BSC)
dario.garcia@bsc.es

