DIGITAL TWIN AI and Machine Learning: Deep Learning II: Convolutional Neural Networks

Prof. Andrew D. Bagdanov
andrew.bagdanov AT unifi.it

Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Firenze

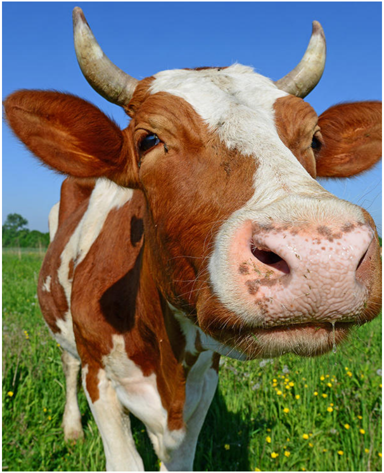
27 November 2020

Outline

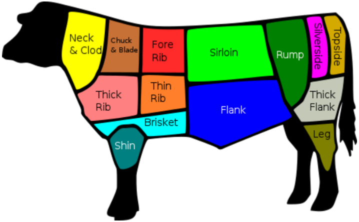
A Critique

What makes a cow, a *cow*?

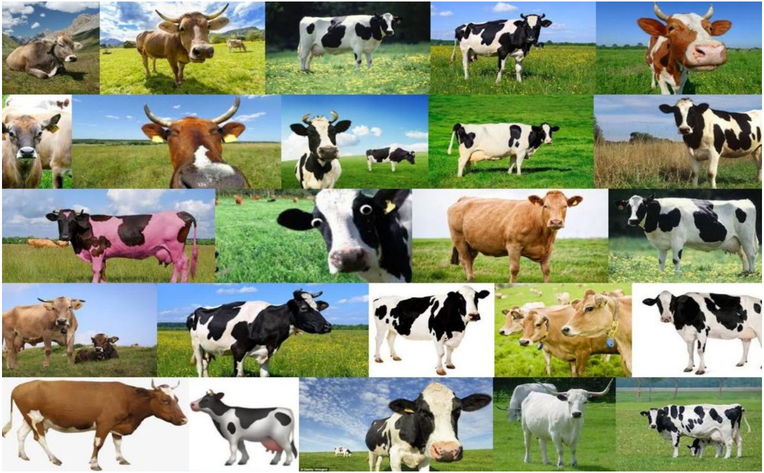
What makes a cow, a *cow*?



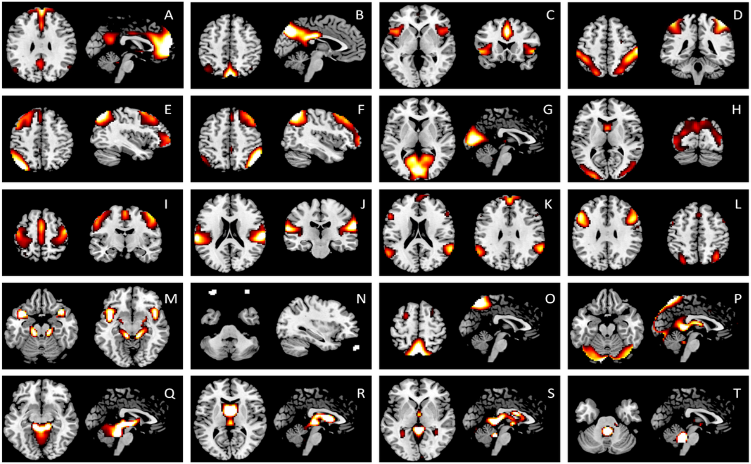
What makes a cow, a *cow*?



What makes a cow, a *cow*?

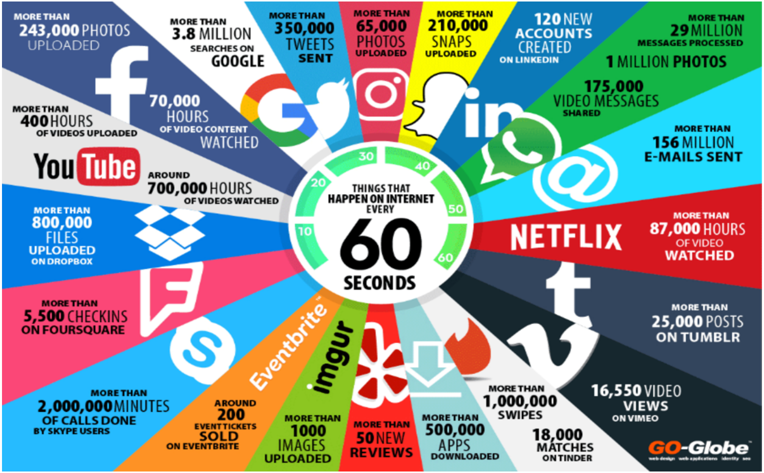


A picture is worth a thousand words. . .



[Cabral et al., 2013]

Content crash: why do we care?

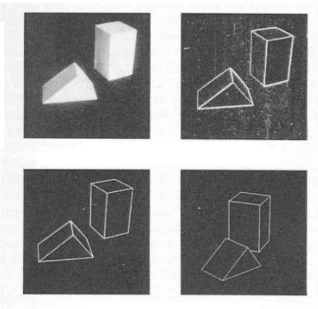


[GO-Gulf Web Design, 2017]

Visual recognition: explicit models

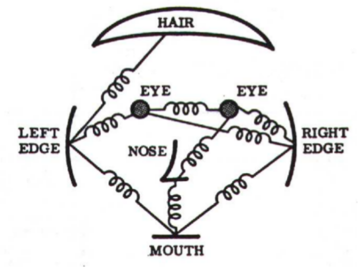
- Early works on visual recognition used (very) explicit models [Roberts, 1963].
- They are significant as first steps and appeal to our analytic beliefs about image understanding.

L. Roberts



Visual recognition: explicit models

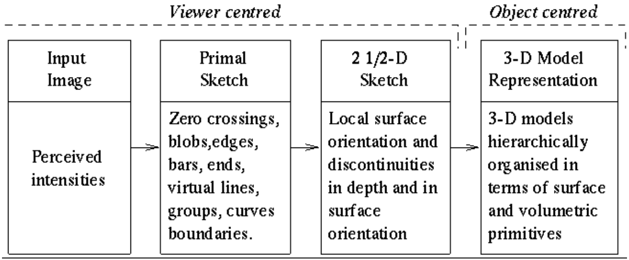
- This type of explicit model of recognition gave way to part-based representations [Fischler and Elschlager, 1973].
- An object was represented by a set of parts arranged in an elastic configuration.
- The trend: move away from models and move towards the image.



[Fischler & Elschlager 73]

Visual recognition: Marr's Vision

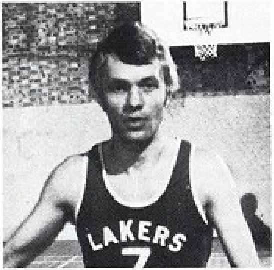
- In his book, Marr developed a modular framework for computer vision [Marr, 1982].
- This framework consists of three representations that are created, maintained, and interpreted by the process of vision:



Vision: A computational investigation into the human representation and processing of visual information

Visual recognition: Marr's Vision

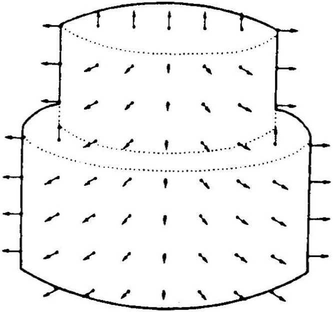
- The Primal Sketch is a description of the intensity changes in the image and their local geometry.
- It is based on the assumption that intensity variations are likely to correspond to physical realities like object boundaries.



Vision: A computational investigation into the human representation and processing of visual information

Visual recognition: Marr's Vision

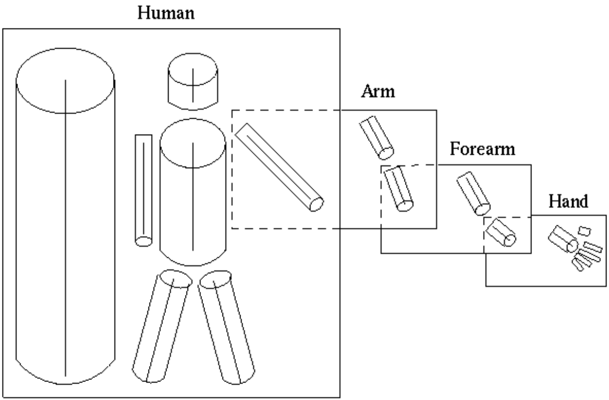
- The 2.5D Sketch is a viewer-centric representation of orientation and depth of visible surfaces drawing from the primal sketch.
- Note that no grouping is done yet: we are only associating weak geometry to image elements.
- Hence the metaphor 2.5D sketch.



Vision: A computational investigation into the human representation and processing of visual information

Visual recognition: Marr's Vision

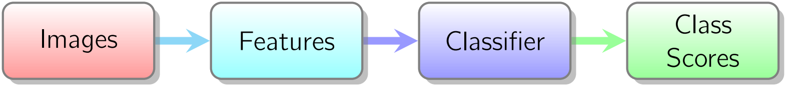
- The 3D Model is an object-centric representation of 3D objects in the image.
- The goal of this model is to enable object manipulation and recognition.



Vision: A computational investigation into the human representation and processing of visual information

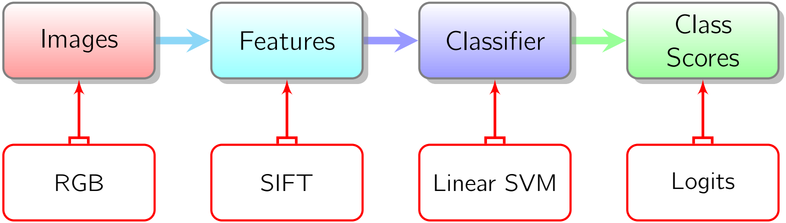
Visual recognition: implicit models

- Let's consider a more-or-less Standard setup of supervised learning for visual classification.
- We can imagine a simple pipeline like below.
- Each stage has it's own design space and critical choices to be made.
- This appeals to the computer scientist in us since we are effectively dividing, modularizing, and (hopefully) conquering.



Visual recognition: implicit models

- Let's consider a more-or-less Standard setup of supervised learning for visual classification.
- We can imagine a simple pipeline like below.
- Each stage has it's own design space and critical choices to be made.
- This appeals to the computer scientist in us since we are effectively dividing, modularizing, and (hopefully) conquering.



Visual recognition: why is this hard?

- A paper appeared in 2000 that summarized the state-of-the-art in visual recognition [Smeulders et al., 2000].
- It introduced sensory gap into the conversation on visual recognition:

The sensory gap is the gap between the object in the world and the information in a (computational) description derived from a recording of that scene.

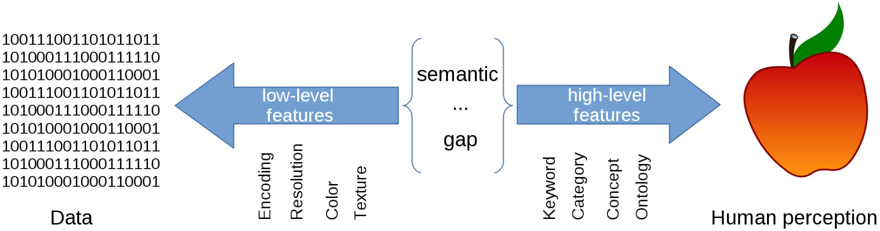
- Think about this for a moment: we are always working with an imperfect reconstruction of the real world.
- Images have limitations: they have finite resolution, they are subject to noise processes, they are acquired with a sensor which is another free object in the world.
- This sensory gap must be surpassed in order to render object recognition invariant to scene-incidental artifacts.

Content-based image retrieval at the end of the early years

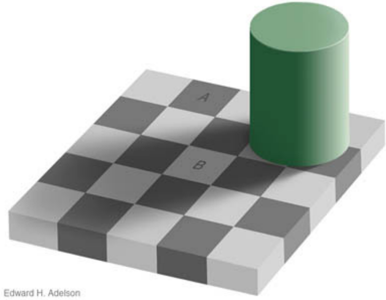
The semantic gap

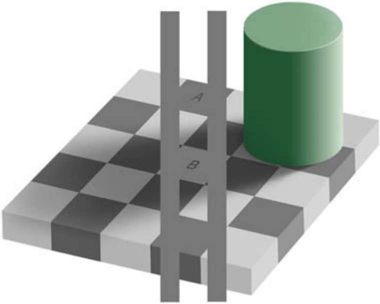
- The other key concept is the semantic gap:

The semantic gap is the lack of coincidence between the information that one can extract from the visual data and the interpretation that the same data have for a user in a given situation.



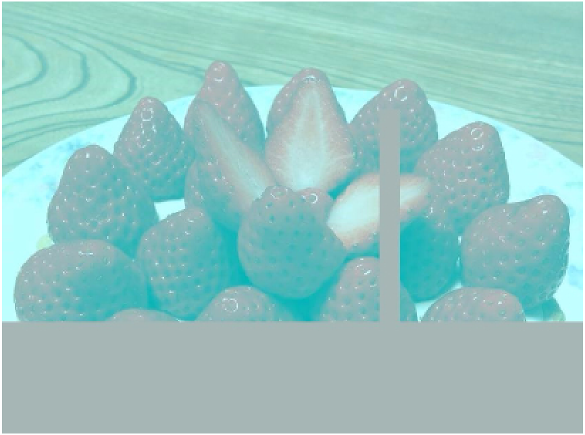
The semantic gap





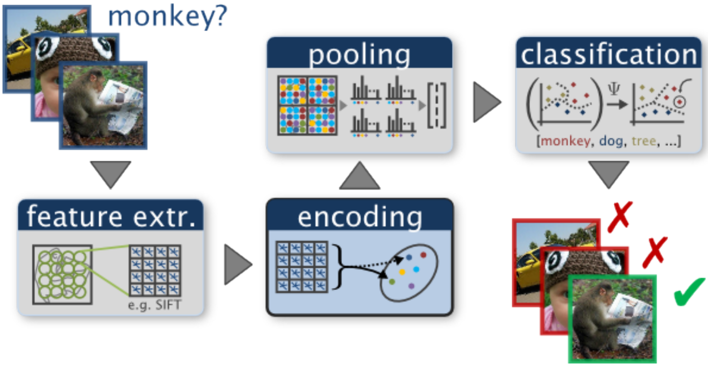
The semantic gap

The semantic gap

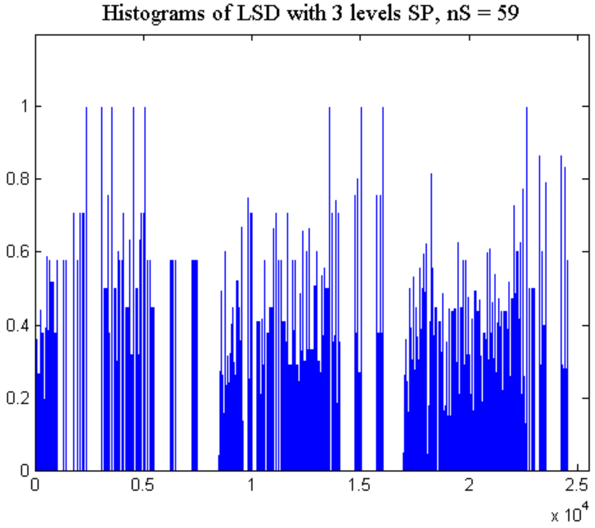


Historical context: Bags of Features

- This was the state-of-the-art in 2011:

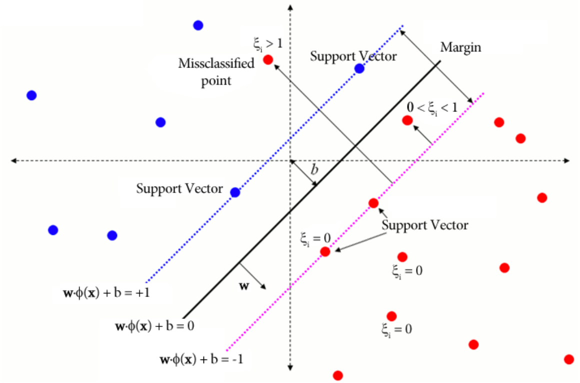


Historical context: This was a "cat"



Historical context: Now "learn"

- To each image representation was associated one (or more) labels.
- Then we feed these into a multi-class SVM.



- Which ran for a while. . . (for some predictable value of "a while").

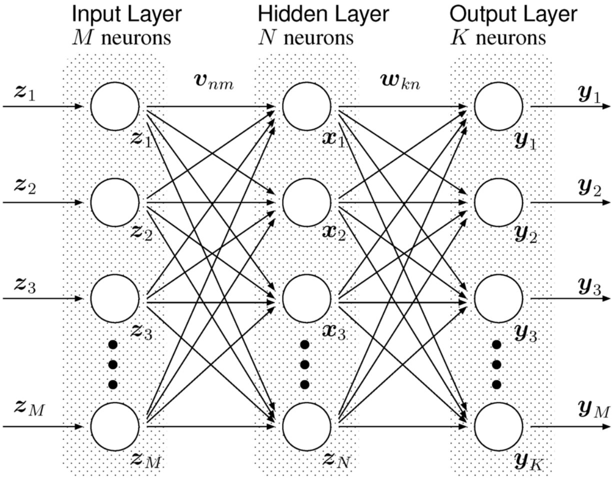
Historical context: A recipe

- Then returned the optimal decision boundaries between all classes (and all the others).
- The process is important:
  1. First: extract a handcrafted representation of fiducial points.
  2. Then: encode these into a global image representation.
  3. Then: fit an SVM (with or without kernel).
- Pro: the actual learning has few hyperparameters (usually just one).
- Con: many handcrafted elements with many (basically infinitely many) hyperparameters.
- Con: learning is separate from representation.

Foundations

Connectionism: The Multilayer Perceptron

- Let's look at a simple Neural Network architecture known as the Multilayer Perceptron (MLP):

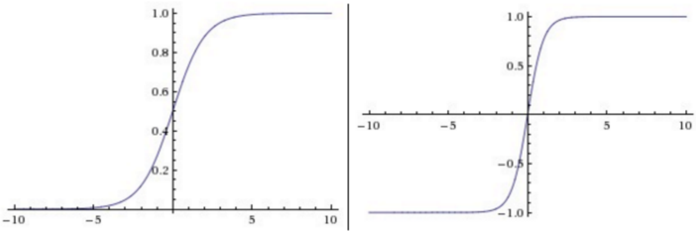


Connectionism: The Multilayer Perceptron

- The MLP equation (one hidden layer):

$$\hat{\mathbf{y}}(\mathbf{x}) = \sigma(\mathbf{w}_2^T \sigma(\mathbf{w}_1^T \mathbf{x} + b_1) + b_2)$$

- Except for the activation function σ, this is a linear system.
- Common activation functions (elementwise):
  - $\sigma(\mathbf{x}) = \tanh(\mathbf{x})$
  - $\sigma(\mathbf{x}) = (1 + e^{-x})^{-1}$
  - $\sigma(\mathbf{x}) = \frac{\exp(x)}{\sum_i e^{x_i}}$ (softmax, used for outputs).



Connectionism: The Multilayer Perceptron

- How do you train a model?
- Decide on a loss function (like the negative log-likelihood):

$$L(\mathbf{y}, \hat{\mathbf{y}}(\mathbf{x})) = -\frac{1}{C}\sum_i y_i \log(\hat{y}_i)$$

- And perform gradient descent w.r.t. all model parameters:

$$\begin{aligned}\boldsymbol{\theta}_{n+1} &= \boldsymbol{\theta}_n - \varepsilon \nabla_{\boldsymbol{\theta}} L(\mathbf{y}, \hat{\mathbf{y}}(\mathbf{x})) \\ \boldsymbol{\theta}_{n+1} &= \boldsymbol{\theta}_n - \varepsilon \sum_{i=1}^{N} \frac{1}{N} \nabla_{\boldsymbol{\theta}} L(\mathbf{y}, \hat{\mathbf{y}}(\mathbf{x}_i))\end{aligned}$$

- Where ε is the learning rate.
- The standard algorithm for this is known as backpropagation and it is very clever and efficient.

Connectionism: The Multilayer Perceptron

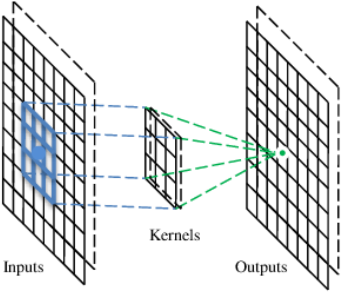
- Problems with this approach:
  - Model size: many, many parameters for even small-sized images. This leads to memory and efficiency problems.
  - Overfitting: many parameters (and limited training data) mean that it is easy to overfit the model to your training set.
  - Undergeneralization: overfitting means that a trained model is unlikely to generalize to new data.
  - Vanishing gradients: a known problem with backpropagation (due to application of the chain rule) leads to very small gradient values near the beginning of the network.
  - Saturating units: traditional activation functions can lead to saturated units (outputs near 1 or 0 (or -1)), which have near-zero derivatives.
- These problems (and others) led the community to largely ignore the potential of these models for decades.

Connectionism: from MLP to CNNs

- However, MLPs have a number extremely attractive features:
  - It is an end-to-end model: we can train everything in the model using a single optimization algorithm.
  - MLPs learn representations of input and classifier.
  - Why can't we just use this model for image recognition problems?
  - An MLP should be able to learn feature representations that are in turn good representations for classification.
  - Why is this problematic?

Connectionism: from MLP to CNNs

- The early layers of a CNN are convolutional (surprise surprise).
- This means that the weights are shared across locations of the image.
- The input of size $w \times h \times d$ is transformed into an output of size $w \times h \times d'$.
- The outputs are called feature maps and they are derived by convolving the image with a 3D tensor of size $u \times v \times d'$.
- So, the number of parameters is "merely" $u * v * d' + d'$.
- The output feature maps can be very large however.



Connectionism: from MLP to CNNs

- What's the link to MLPs?

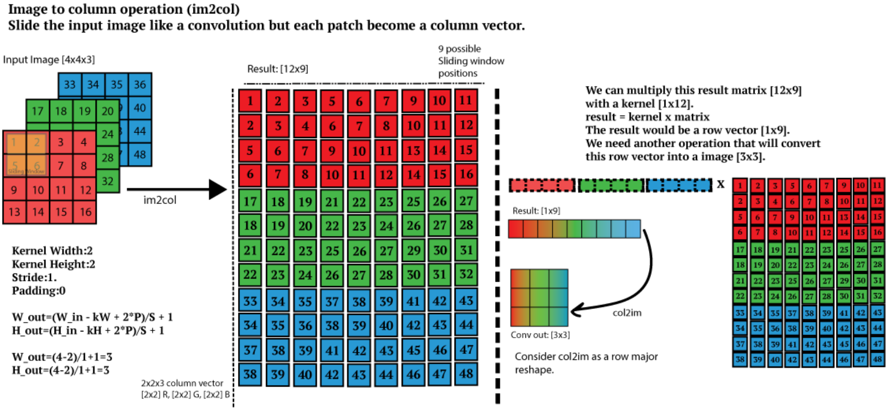


Figure from: https://github.com/leonardoaraujosantos

A Tour

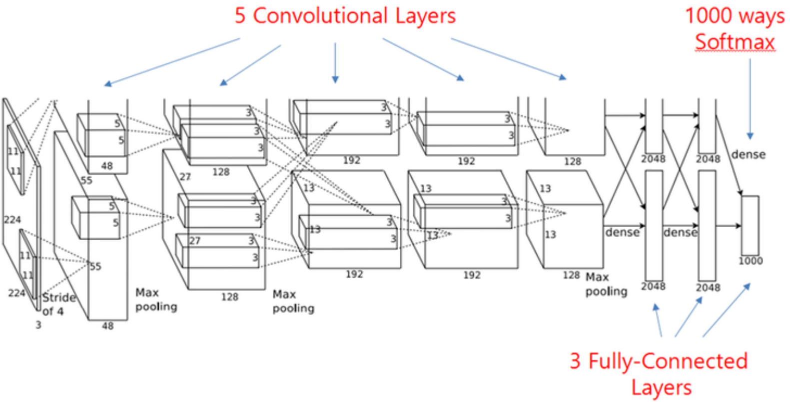
AlexNet: Introduction

- We will now take a look at the International Large Scale Visual Recognition Competition (ILSVRC) submission that changed everything [Krizhevsky et al., 2012].
- This architecture systematically addresses most of the problems with training large network architectures on large datasets.
- It is a Convolutional Neural Network (CNN) that is universally called AlexNet.
- It is also a Deep Network because it has many hidden layers.

ImageNet Classification with Deep Convolutional Neural Networks

AlexNet: The Architecture

- Let's look first at the overall architecture and then analyze in detail how each component addresses specific problems.
- It is also helpful to examine how data flows through the network.



ImageNet Classification with Deep Convolutional Neural Networks

AlexNet: Pooling Features

- Like in the Bag-of-Words model, we can pool local features.
- AlexNet uses 3×3 pooling regions with a stride of 2 pixels.
- This means that after some convolutional layers the feature map size is reduced by a factor of 2.
- They use max pooling: in each feature map, keep the maximum value in each overlapping 3×3 pooling region (in each feature map).
- This helps to contain the size of feature maps propagated through the network.
- And it also helps to build higher-level representations of the image.
- This is because, halving the image resolution is the same as doubling the size of subsequent convolutions.

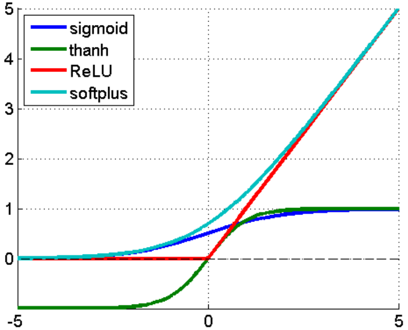
ImageNet Classification with Deep Convolutional Neural Networks

AlexNet: Unit Saturation

- Another innovation in AlexNet is the use of the Rectified Linear Unit (ReLU) activation function.

$$\sigma(\mathbf{x}) = max(0, \mathbf{x})$$

- This activation function does not saturate like sigmoids.
- The result is a 6x speedup in training.



ImageNet Classification with Deep Convolutional Neural Networks

AlexNet: Reducing Overfitting

- Even with convolutional weight sharing, AlexNet still has 60M parameters.
- To reduce overfitting, the authors use two extra (now standard) tricks:
  - Data augmentation: random translations and reflections of input images are generated, plus random variation in principal directions of RGB space.
  - Dropout: an advanced trick from the Neural Network community which randomly removes half of the inputs to select layers at training time.

ImageNet Classification with Deep Convolutional Neural Networks

AlexNet: More Tricks

- The AlexNet paper is an excellent resource because it explains all of the tricks necessary to get a deep network to learn:
  - Local response normalization: keep local variation in feature maps under control (section 3.3).
  - Momentum: limits the "skateboard" effect when following valleys in the loss surface, equivalent to L1 (or L2) regularization of weights (section 5).
  - Mini-batch Stochastic Gradient Descent (SGD): with 1.2M training samples, we cannot consider the entire dataset in one batch; instead, randomly sample mini-batches of 128 images (section 5).
  - Multiple GPUs: AlexNet was too big to fit in a single GPU (in 2012), so feature maps are split over two GPUs (section 3.2).
  - Model averaging: state-of-the-art results are obtained by training multiple CNNs and averaging outputs.

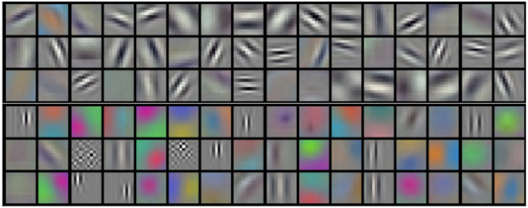
ImageNet Classification with Deep Convolutional Neural Networks

AlexNet: Results

- The proof is in the pudding:

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SIFT + FVs [7]	—	—	*26.2%*
1 CNN	40.7%	18.2%	—
5 CNNs	38.1%	16.4%	**16.4%**
1 CNN*	39.0%	16.6%	—
7 CNNs*	36.7%	15.4%	**15.3%**

- And in the representations the network learns:



ImageNet Classification with Deep Convolutional Neural Networks

AlexNet: Reflections

- AlexNet took the object recognition world by storm.
- Many of the elements of the model are not really new.
- However, this was the first work to convincingly demonstrate how state-of-the-art object recognition systems can be trained end-to-end on real problems.
- This was made possible by a number of confluent development:
  - The availability of enormous amounts of annotate data (ImageNet, with 1.2M training images).
  - Modern GPUs, which make convolutions super fast.
  - Decades of persistent theoretical development (ReLUs, fast backprop, dropout, etc).

Reflections: CNNs are really big

- One of the first observations one can make about CNNs is that they have a HUGE number of parameters.
- Even modestly-sized, state-of-the-art networks can have on the order of *150 million trainable parameters*.
- Fitting such models of course requires massive amounts of labeled training data.

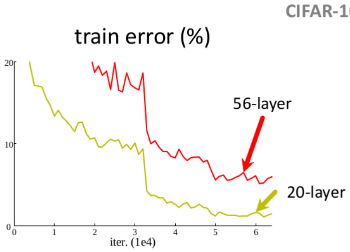
ResNets: Introduction

- From what we have seen so far, it seems like deeper networks generalize better.
- So, can we just keep stacking more and more layers onto the end of our CNNs?
- Aside from the computational complications (GPU memory is finite), this seems like it should "just work".
- We will now look at our last state-of-the-art network architecture (known as ResNet) which looks at this question in detail [He et al., 2016].

Deep residual learning for image recognition

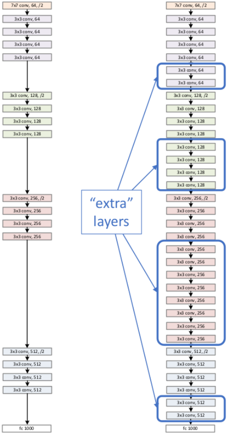
ResNets: Deeper isn't better?

- Pre-ResNet Thinking: *Deeper networks should always perform better – at least on the training data.*



Deep residual learning for image recognition

ResNets: Wait, shouldn't training error be lower?

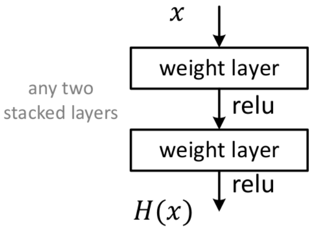


- Using an artificial construction, we see that the training error at least shouldn't increase with depth.
- Just copy pre-trained weights from plain network into a deeper network with new, randomly initialized weights.

Deep residual learning for image recognition

ResNets: Targets and Residuals

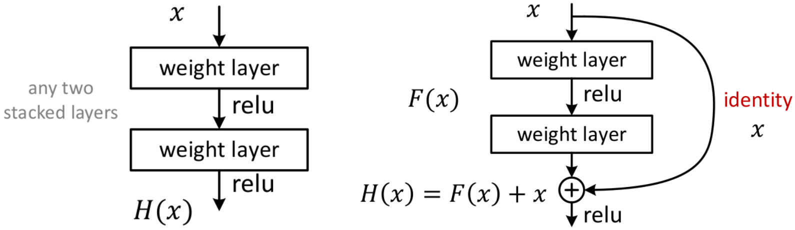
- Let's say that the network is learning towards some optimal feature representation $H(x)$.
- The compositional and feed-forward nature of the CNN isn't really helping.



Deep residual learning for image recognition

ResNets: Targets and Residuals

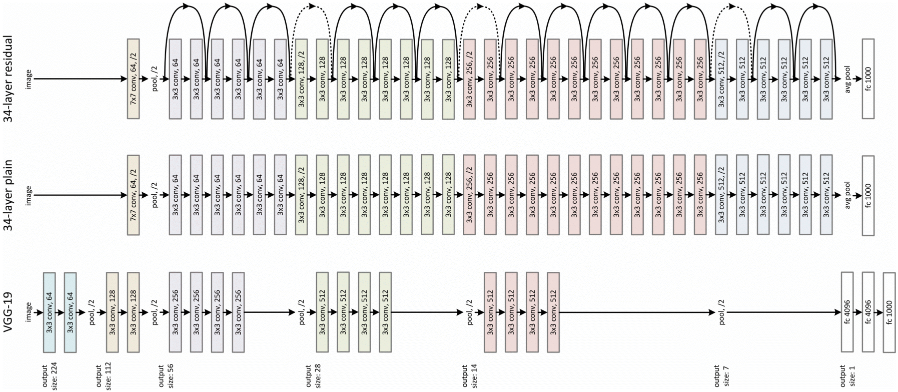
- Let's say that the network is learning towards some optimal feature representation $H(x)$.
- The compositional and feed-forward nature of the CNN isn't really helping.
- Instead, we can help the network out by not requiring it to pass through as much information.
- Pass x forward and add it to the output of the residual block – now we "only" need to learn $H(x) - x$.



Deep residual learning for image recognition

ResNets: Comparison

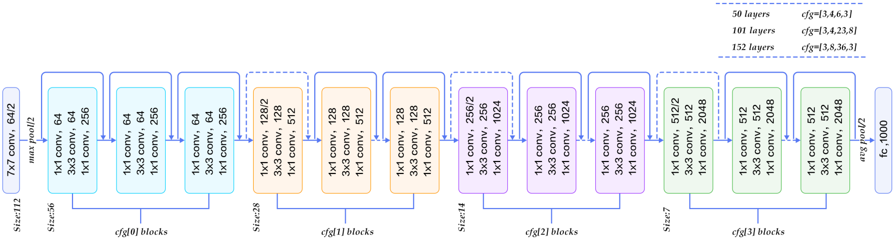
- Here is a comparison of VGG19 and ResNet-34:



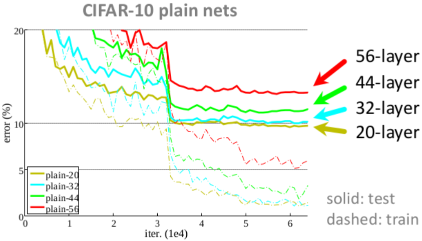
Deep residual learning for image recognition

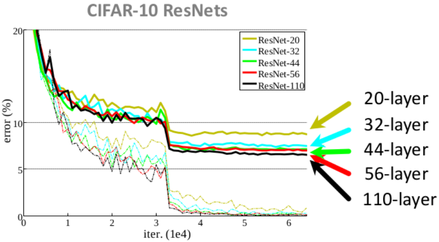
ResNets: Parametric Modularity

- And this is a common way of parametrically representing the various ResNet configurations:



ResNets: Results

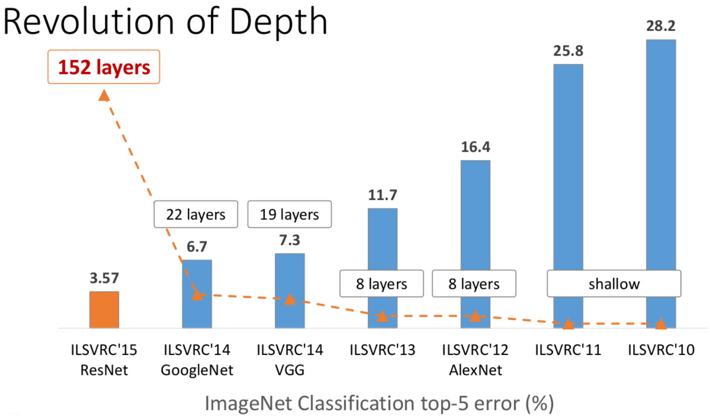




Deep residual learning for image recognition

ResNet: Results

- And the proof, as always, is in the pudding:



CNNs: How do you train CNNs?

- CNNs work great, but it's not always sunshine and lollipops trying to get them to work.
- The community has developed a number of tricks, techniques, and heuristics that are proven to help.
- Let's look at a few of them.

CNNs: Batch Normalization

- Attention to the data distribution (through the whole network) and normalization are critical [Ioffe and Szegedy, 2015]:

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1 \ldots m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

Input: Network N with trainable parameters Θ;
subset of activations $\{x^{(k)}\}_{k=1}^{K}$
Output: Batch-normalized network for inference, $N_{\mathrm{BN}}^{\mathrm{inf}}$

$N_{\mathrm{BN}}^{\mathrm{tr}} \leftarrow N$ // Training BN network
for $k = 1 \ldots K$ **do**
  Add transformation $y^{(k)} = \mathrm{BN}_{\gamma^{(k)},\beta^{(k)}}(x^{(k)})$ to $N_{\mathrm{BN}}^{\mathrm{tr}}$ (Alg. 1)
  Modify each layer in $N_{\mathrm{BN}}^{\mathrm{tr}}$ with input $x^{(k)}$ to take $y^{(k)}$ instead
end for
Train $N_{\mathrm{BN}}^{\mathrm{tr}}$ to optimize the parameters $\Theta \cup \{\gamma^{(k)}, \beta^{(k)}\}_{k=1}^{K}$
$N_{\mathrm{BN}}^{\mathrm{inf}} \leftarrow N_{\mathrm{BN}}^{\mathrm{tr}}$ // Inference BN network with frozen // parameters
for $k = 1 \ldots K$ **do**
  // For clarity, $x \equiv x^{(k)}, \gamma \equiv \gamma^{(k)}, \mu_{\mathcal{B}} \equiv \mu_{\mathcal{B}}^{(k)}$, etc.
  Process multiple training mini-batches $\mathcal{B}$, each of size m, and average over them:

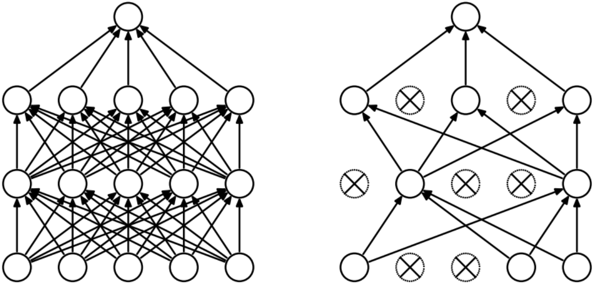
$$\mathrm{E}[x] \leftarrow \mathrm{E}_{\mathcal{B}}[\mu_{\mathcal{B}}]$$
$$\mathrm{Var}[x] \leftarrow \tfrac{m}{m-1}\mathrm{E}_{\mathcal{B}}[\sigma_{\mathcal{B}}^2]$$

  In $N_{\mathrm{BN}}^{\mathrm{inf}}$, replace the transform $y = \mathrm{BN}_{\gamma,\beta}(x)$ with $y = \frac{\gamma}{\sqrt{\mathrm{Var}[x]+\epsilon}} \cdot x + \left(\beta - \frac{\gamma\,\mathrm{E}[x]}{\sqrt{\mathrm{Var}[x]+\epsilon}}\right)$
end for

Algorithm 2: Training a Batch-Normalized Network

Batch normalization: Accelerating deep network training by reducing internal covariate shift

CNNs: Dropout



(a) Standard Neural Net (b) After applying dropout.

With unlimited computation, the best way to "regularize" a fixed-sized model is to average the predictions of all possible settings of the parameters, weighting each setting by its posterior probability given the training data.

– Srivastava et al. [2014]

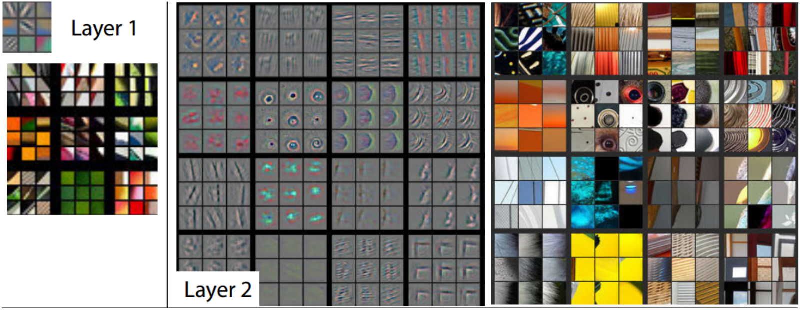
Dropout: A Simple Way to Prevent Neural Networks from Overfitting

CNNs: What's Going On?

- Remember earlier I mentioned that one of the biases against using neural networks was that lack of interpretability.
- As soon as the spectacular results of CNNs on object recognition started coming in, researchers began inventing new ways to interpret the innards of these huge networks.
- Nowadays there are many tools and tricks you can use to understand what the network has learned.

CNNs: What's Going On

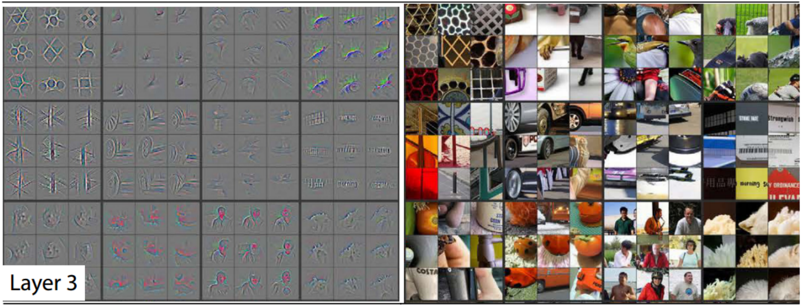
- An early work looked at just this problem and the paper has a ton of interesting analysis of how these networks work [Zeiler and Fergus, 2014].
- I am only going to talk about how visualizations of feature map activations demonstrate what's going on.



Visualizing and understanding convolutional networks

CNNs: What's Going On

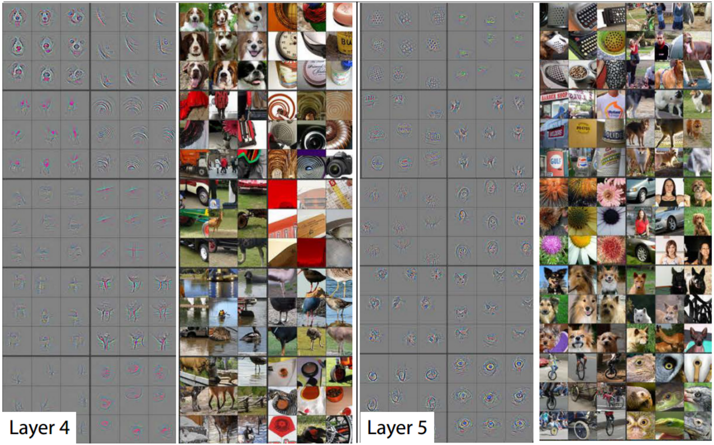
- As we go deeper into the network, feature activations correspond to higher-level semantics.



Visualizing and understanding convolutional networks

CNNs: What's Going On

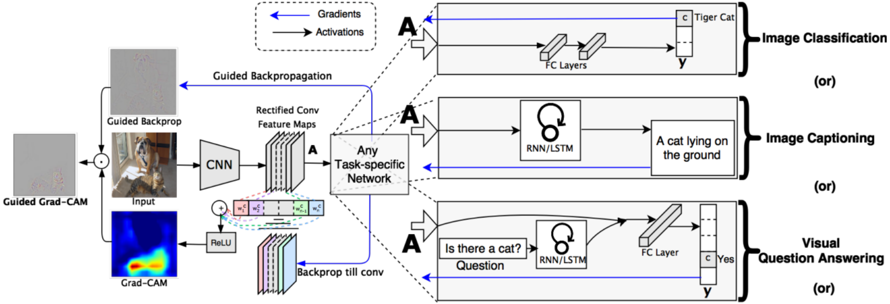
- Until the network is really indicating the presence of "eyes" and "cat faces", etc.



Visualizing and understanding convolutional networks

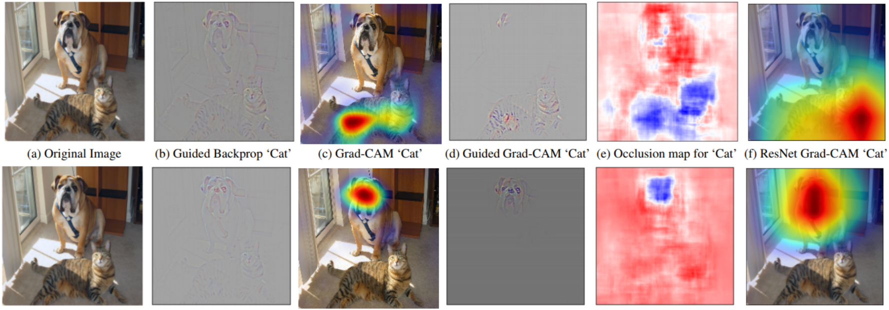
CNNs: What's Going On

- The Grad-CAM is an intuitive way to visualize, well, what makes a cow, a cow [Selvaraju et al., 2017]:



CNNs: What's Going On

- The Grad-CAM is an intuitive way to visualize, well, what makes a cow, a cow [Selvaraju et al., 2017]:



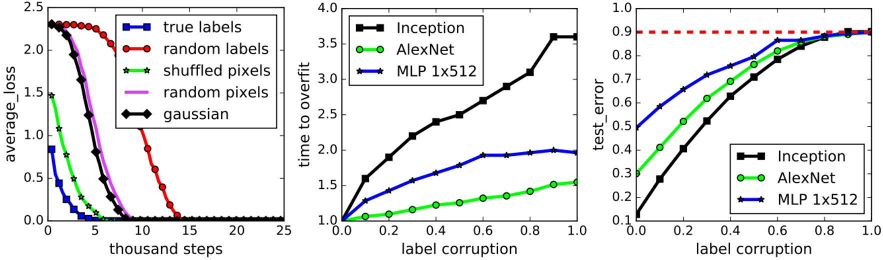
(a) Original Image (b) Guided Backprop 'Cat' (c) Grad-CAM 'Cat' (d) Guided Grad-CAM 'Cat' (e) Occlusion map for 'Cat' (f) ResNet Grad-CAM 'Cat'

(g) Original Image (h) Guided Backprop 'Dog' (i) Grad-CAM 'Dog' (j) Guided Grad-CAM 'Dog' (k) Occlusion map for 'Dog' (l) ResNet Grad-CAM 'Dog'

Discussion

Discussion: The burden of supervision

- What is 1.5M annotations really worth [Zhang et al., 2016]?
- If labels are equally probable, a randomly shuffled set of ImageNet labels contains about $1.5M * \log_2(1000) \approx 15$Mbits.

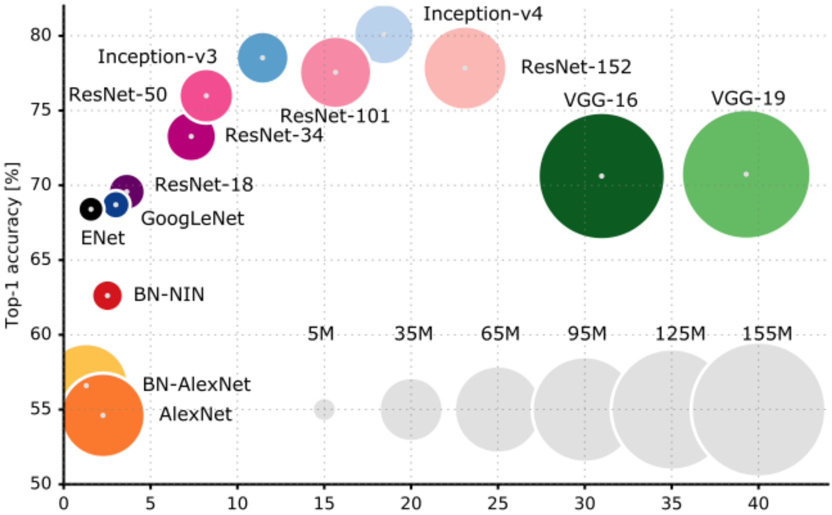


(a) learning curves (b) convergence slowdown (c) generalization error growth

Understanding deep learning requires rethinking generalization

Discussion: The state-of-the-art

- We have come a long way in a few years.



Discussion: CNNs and the way forward

Bibliography I

A. Canziani, A. Paszke, and E. Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016.

M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on computers*, (1):67–92, 1973.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

D. Marr. Vision: A computational investigation into the human representation and processing of visual information, 1982.

L. G. Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963.

Bibliography II

R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (12):1349–1380, 2000.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Laboratory

- The laboratory notebook for today:

http://bit.ly/DTwin-ML8