Category Recognition

Jia-Bin Huang
Virginia Tech
ECE 6554 Advanced Computer Vision
Administrative stuffs

• Presentation and discussion leads assigned
  • https://docs.google.com/spreadsheets/d/1P5pfyCio5flq3QCy4Mo1XS66l6d14jqDxE2Tny4efVs/edit#gid=0

• Questions?
Today’s class

• Finish instance recognition

• Category recognition

• Convolutional neural network
From Dusk till Dawn: Modeling in the Dark, CVPR 2016
Lift: Learned invariant feature transform, ECCV 2016
Instance recognition

• Motivation – visual search
• Visual words
  • quantization, index, bags of words
• Spatial verification
  • affine; RANSAC, Hough
• Other text retrieval tools
  • tf-idf, query expansion
• Example applications
Instance recognition: remaining issues

• How to summarize the content of an entire image? And gauge overall similarity?

• How large should the vocabulary be? How to perform quantization efficiently?

• Is having the same set of visual words enough to identify the object/scene? How to verify spatial agreement?

• How to score the retrieval results?
Spatial Verification

Both image pairs have many visual words in common.

Slide credit: Ondrej Chum
Spatial Verification

Only some of the matches are mutually consistent
Spatial Verification: two basic strategies

• RANSAC
  • Typically sort by BoW similarity as initial filter
  • Verify by checking support (inliers) for possible transformations
    • e.g., “success” if find a transformation with > N inlier correspondences

• Generalized Hough Transform
  • Let each matched feature cast a vote on location, scale, orientation of the model object
  • Verify parameters with enough votes
RANSAC verification
Recall: Fitting an affine transformation

\[
\begin{bmatrix}
  x'_i \\
y'_i
\end{bmatrix}
= \begin{bmatrix}
m_1 & m_2 \\
m_3 & m_4
\end{bmatrix}
\begin{bmatrix}
x_i \\
y_i
\end{bmatrix}
+ \begin{bmatrix}
t_1 \\
t_2
\end{bmatrix}
\]

Approximates viewpoint changes for roughly planar objects and roughly orthographic cameras.
RANSAC verification
Video Google System

1. Collect all words within query region
2. Inverted file index to find relevant frames
3. Compare word counts
4. Spatial verification

Sivic & Zisserman, ICCV 2003

- Demo online at:
  http://www.robots.ox.ac.uk/~vgg/research/vgoogle/index.html
Example Applications

Mobile tourist guide
- Self-localization
- Object/building recognition
- Photo/video augmentation

[Quack, Leibe, Van Gool, CIVR’08]
Application: Large-Scale Retrieval

Query

Results from 5k Flickr images (demo available for 100k set)

[Philbin CVPR’07]
Web Demo: Movie Poster Recognition

50’000 movie posters indexed

Query-by-image from mobile phone available in Switzerland

1. Take a picture with your mobile phone camera
2. Send it:
   - in Switzerland to 5555 (Orange Customers 079 394 5700).
   - in Germany to 81000
   - everywhere else to m@kooaba.ch
3. Search result is sent straight to your phone.

Spatial Verification: two basic strategies

• RANSAC
  • Typically sort by BoW similarity as initial filter
  • Verify by checking support (inliers) for possible transformations
    • e.g., “success” if find a transformation with > N inlier correspondences

• Generalized Hough Transform
  • Let each matched feature cast a vote on location, scale, orientation of the model object
  • Verify parameters with enough votes
Voting: Generalized Hough Transform

- If we use scale, rotation, and translation invariant local features, then each feature match gives an alignment hypothesis (for scale, translation, and orientation of model in image).

Model  

Novel image

Adapted from Lana Lazebnik
Voting: Generalized Hough Transform

- A hypothesis generated by a single match may be unreliable,
- So let each match **vote** for a hypothesis in Hough space
Gen Hough Transform details (Lowe’s system)

• **Training phase:** For each model feature, record 2D location, scale, and orientation of model (relative to normalized feature frame)

• **Test phase:** Let each match between a test SIFT feature and a model feature vote in a 4D Hough space
  - Use broad bin sizes of 30 degrees for orientation, a factor of 2 for scale, and 0.25 times image size for location
  - Vote for two closest bins in each dimension

• Find all bins with at least three votes and perform geometric verification
  - Estimate least squares *affine* transformation
  - Search for additional features that agree with the alignment

Example result

Background subtract for model boundaries

Objects recognized,

Recognition in spite of occlusion

[Lowe]
Recall: difficulties of voting

• Noise/clutter can lead to as many votes as true target

• Bin size for the accumulator array must be chosen carefully

• In practice, good idea to make broad bins and spread votes to nearby bins, since verification stage can prune bad vote peaks.
Gen Hough vs RANSAC

**GHT**
- Single correspondence -> vote for all consistent parameters
- Represents uncertainty in the model parameter space
- Linear complexity in number of correspondences and number of voting cells; beyond 4D vote space impractical
- Can handle high outlier ratio

**RANSAC**
- Minimal subset of correspondences to estimate model -> count inliers
- Represents uncertainty in image space
- Must search all data points to check for inliers each iteration
- Scales better to high-d parameter spaces
China is forecasting a trade surplus of $90bn (£51bn) to $100bn this year, a threefold increase on 2004's $32bn. The Commerce Ministry said the surplus would be created by a predicted 30% jump in exports to $750bn, compared with a 18% rise in imports to $660bn. The figures are likely to further annoy the US, which has long argued that China's exports are unfairly helped by a deliberately undervalued yuan. Beijing agrees the surplus is too high, but says the yuan is only one factor. Bank of China governor Zhou Xiaochuan said the country also needed to do more to boost domestic demand so more goods stayed within the country. China increased the value of the yuan against the dollar by 2.1% in July and permitted it to trade within a narrow band, but the US wants the yuan to be allowed to trade freely. However, Beijing has made it clear that it will take its time and tread carefully before allowing the yuan to rise further in value.
**tf-idf weighting**

- **Term frequency – inverse document frequency**
- Describe frame by frequency of each word within it, downweight words that appear often in the database
- (Standard weighting for text retrieval)

\[ t_i = \frac{n_{id}}{n_d} \log \left( \frac{N}{n_i} \right) \]

- Number of occurrences of word \( i \) in document \( d \)
- Number of words in document \( d \)
- Total number of documents in database
- Number of documents word \( i \) occurs in, in whole database
Query Expansion

Query image

Results

Spatial verification

New results

New query

Chum, Philbin, Sivic, Isard, Zisserman: Total Recall..., ICCV 2007

Slide credit: Ondrej Chum
Recognition via alignment

**Pros:**
- Effective when we are able to find reliable features within clutter
- Great results for matching specific instances

**Cons:**
- Scaling with number of models
- Spatial verification as post-processing – not seamless, expensive for large-scale problems
- Not suited for category recognition.
Making the Sky Searchable: Fast Geometric Hashing for Automated Astrometry

Sam Roweis, Dustin Lang & Keir Mierle
University of Toronto

David Hogg & Michael Blanton
New York University
Example

A shot of the Great Nebula, by Jerry Lodriguss (c.2006), from astropix.com
http://astrometry.net/gallery.html
Example

An amateur shot of M100, by Filippo Ciferri (c.2007) from [flickr.com](http://flickr.com) and [astrometry.net/gallery.html](http://astrometry.net/gallery.html)
A beautiful image of Bode's nebula (c.2007) by Peter Bresseler, from starlightfriend.de
http://astrometry.net/gallery.html
Things to remember

• Matching local invariant features
  • Useful not only to provide matches for multi-view geometry, but also to find objects and scenes.

• Bag of words representation: quantize feature space to make discrete set of visual words
  • Summarize image by distribution of words
  • Index individual words

• Inverted index: pre-compute index to enable faster search at query time

• Recognition of instances via alignment: matching local features followed by spatial verification
  • Robust fitting : RANSAC, GHT
Discussion – Think-pair-share

• Find a person you don’t know

• Discuss
  • strength,
  • weakness, and
  • potential extension

• Share with class
Image Categorization: Training phase

Training Images

Training

Training Labels

Image Features

Classifier Training

Trained Classifier
Image Categorization: Testing phase

Training Images

Training

Image Features

Classifier Training

Trained Classifier

Training Labels

Testing

Image Features

Trained Classifier

Prediction Outdoor

Test Image
Image categorization

- Cat vs Dog
Image categorization

• Object recognition

Caltech 101 Average Object Images
Image categorization

- Fine-grained recognition

- Generalist
- Insect catching
- Grain eating
- Coniferous-seed eating
- Nectar feeding
- Chiseling
- Dip netting
- Surface skimming
- Scything
- Probing
- Aerial fishing
- Pursuit fishing
- Scavenging
- Raptorial
- Filter feeding

Visipedia Project
Image categorization

• Place recognition

Places Database [Zhou et al. NIPS 2014]
Image categorization

- Visual font recognition

[Chen et al. CVPR 2014]
Image categorization

• Dating historical photos

[Palermo et al. ECCV 2012]
Image categorization

• Image style recognition

HDR | Macro | Baroque | Roccoco
---|---|---|---
Vintage | Noir | Northern Renaissance | Cubism
Minimal | Hazy | Impressionism | Post-Impressionism
Long Exposure | Romantic | Abs. Expressionism | Color Field Painting

Flickr Style: 80K images covering 20 styles.
Wikipaintings: 85K images for 25 art genres.

[Karayev et al. BMVC 2014]
Features are the Keys

SIFT [Loewe IJCV 04]

HOG [Dalal and Triggs CVPR 05]

SPM [Lazebnik et al. CVPR 06]

DPM [Felzenszwalb et al. PAMI 10]

Color Descriptor [Van De Sande et al. PAMI 10]
Learning a Hierarchy of Feature Extractors

- Each layer of hierarchy extracts features from output of previous layer
- All the way from pixels $\rightarrow$ classifier
- Layers have the (nearly) same structure
Biological neuron and Perceptrons

A biological neuron

An artificial neuron (Perceptron) - a linear classifier
Suggested a hierarchy of feature detectors in the visual cortex, with higher level features responding to patterns of activation in lower level cells, and propagating activation upwards to still higher level cells.
Hubel/Wiesel Architecture and Multi-layer Neural Network

Hubel and Weisel’s architecture

multi-layer Neural Network
- A non-linear classifier
Multi-layer Neural Network

- A non-linear classifier

**Training:** find network weights $\mathbf{w}$ to minimize the error between true training labels $y_i$ and estimated labels $f_{\mathbf{w}}(\mathbf{x}_i)$

$$E(\mathbf{w}) = \sum_{i=1}^{N} (y_i - f_{\mathbf{w}} (\mathbf{x}_i))^2$$

- Minimization can be done by gradient descent provided $f$ is differentiable
- This training method is called **back-propagation**
Convolutional Neural Networks

• Also known as CNN, ConvNet, DCN

• CNN = a multi-layer neural network with
  1. Local connectivity
  2. Weight sharing
CNN: Local Connectivity

- # input units (neurons): 7
- # hidden units: 3
- Number of parameters
  - Global connectivity: $3 \times 7 = 21$
  - Local connectivity: $3 \times 3 = 9$
CNN: Weight Sharing

Without weight sharing

- # input units (neurons): 7
- # hidden units: 3
- Number of parameters
  - Without weight sharing: $3 \times 3 = 9$
  - With weight sharing: $3 \times 1 = 3$

With weight sharing
CNN with multiple input channels

**Single** input channel

**Multiple** input channels

Filter weights
CNN with multiple output maps

Single output map

Multiple output maps

Filter weights

Filter weights
Putting them together

• Local connectivity
• Weight sharing
• Handling multiple input channels
• Handling multiple output maps

Image credit: A. Karpathy
Neocognitron [Fukushima, Biological Cybernetics 1980]

Deformation-Resistant Recognition

S-cells: (simple)
- extract local features

C-cells: (complex)
- allow for positional errors
LeNet [LeCun et al. 1998]

Gradient-based learning applied to document recognition [LeCun, Bottou, Bengio, Haffner 1998]

LeNet-1 from 1993
What is a Convolution?

• Weighted moving sum

slide credit: S. Lazebnik
Convolutional Neural Networks

- Convolution (Learned)
- Non-linearity
- Spatial pooling
- Normalization
- Feature maps

Input Image
Convolutional Neural Networks

Feature maps
Normalization
Spatial pooling
Non-linearity
Convolution (Learned)
Input Image

Input
Feature Map

slide credit: S. Lazebnik
Convolutional Neural Networks

Feature maps
Normalization
Spatial pooling
Non-linearity
Convolution (Learned)
Input Image

Rectified Linear Unit (ReLU)

slide credit: S. Lazebnik
Convolutional Neural Networks

Max-pooling: a non-linear down-sampling

Provide *translation invariance*
Convolutional Neural Networks

- Feature maps
- Normalization
- Spatial pooling
- Non-linearity
- Convolution (Learned)
- Input Image

Feature Maps

Feature Maps After Contrast Normalization

slide credit: S. Lazebnik
Convolutional Neural Networks

1. Input Image
2. Convolution (Learned)
3. Non-linearity
4. Spatial pooling
5. Normalization
6. Feature maps

slide credit: S. Lazebnik
Convolutional filters are trained in a supervised manner by back-propagating classification error.
Gradient-Based Learning Applied to Document Recognition, LeCun, Bottou, Bengio and Haffner, Proc. of the IEEE, 1998

Imagenet Classification with Deep Convolutional Neural Networks, Krizhevsky, Sutskever, and Hinton, NIPS 2012

Slide Credit: L. Zitnick
Gradient-Based Learning Applied to Document Recognition, LeCun, Bottou, Bengio and Haffner, Proc. of the IEEE, 1998

* Rectified activations and dropout

Slide Credit: L. Zitnick
SIFT Descriptor

- **Image Pixels**
- **Apply gradient filters**
- **Spatial pool (Sum)**
- **Normalize to unit length**
- **Feature Vector**

Lowe [IJCV 2004]
SIFT Descriptor

Image Pixels → Apply oriented filters

Spatial pool (Sum) → Normalize to unit length

Feature Vector

Lowe [IJCV 2004]

slide credit: R. Fergus
Spatial Pyramid Matching

- SIFT Features → Filter with Visual Words
- Max
- Multi-scale spatial pool (Sum) → Classifier

Lazebnik, Schmid, Ponce [CVPR 2006]

slide credit: R. Fergus
Deformable Part Models are Convolutional Neural Networks [Girshick et al. CVPR 15]
AlexNet

• Similar framework to LeCun’98 but:
  • Bigger model (7 hidden layers, 650,000 units, 60,000,000 params)
  • More data ($10^6$ vs. $10^3$ images)
  • GPU implementation (50x speedup over CPU)
    • Trained on two GPUs for a week

A. Krizhevsky, I. Sutskever, and G. Hinton,
ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012
Using CNN for Image Classification

AlexNet

Fully connected layer Fc7
d = 4096

Averaging

Softmax Layer

“Jia-Bin”

Fixed input size:
224x224x3

d = 4096
Progress on ImageNet

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2012</td>
<td>AlexNet</td>
<td>16.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>ZF</td>
<td></td>
<td>11.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>VGG</td>
<td>7.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2014</td>
<td>GoogLeNet</td>
<td></td>
<td></td>
<td>6.7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015</td>
<td>ResNet</td>
<td></td>
<td></td>
<td></td>
<td>3.57</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2016</td>
<td>GoogLeNet-v4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3.08</td>
<td></td>
</tr>
</tbody>
</table>

ImageNet Image Classification Top5 Error
VGG-Net

• The deeper, the better

• Key design choices:
  • 3x3 conv. Kernels
    - very small
  • conv. stride 1
    - no loss of information

• Other details:
  • Rectification (ReLU) non-linearity
  • 5 max-pool layers (x2 reduction)
  • no normalization
  • 3 fully-connected (FC) layers
VGG-Net

• Why 3x3 layers?
  • Stacked conv. layers have a large receptive field
  • two 3x3 layers – 5x5 receptive field
  • three 3x3 layers – 7x7 receptive field

• More non-linearity
  • Less parameters to learn
  • ~140M per net
ResNet

- Can we just increase the layer?
- How can we train very deep network?
  - Residual learning

<table>
<thead>
<tr>
<th>method</th>
<th>top-5 err. (test)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG [41] (ILSVRC’14)</td>
<td>7.32</td>
</tr>
<tr>
<td>GoogLeNet [44] (ILSVRC’14)</td>
<td>6.66</td>
</tr>
<tr>
<td>VGG [41] (v5)</td>
<td>6.8</td>
</tr>
<tr>
<td>PReLU-net [13]</td>
<td>4.94</td>
</tr>
<tr>
<td>BN-Inception [16]</td>
<td>4.82</td>
</tr>
<tr>
<td>ResNet (ILSVRC’15)</td>
<td><strong>3.57</strong></td>
</tr>
</tbody>
</table>
DenseNet

• Shorter connections (like ResNet) help
• Why not just connect them all?
Training Convolutional Neural Networks

- Backpropagation + stochastic gradient descent with momentum
  - [Neural Networks: Tricks of the Trade](#)
- Dropout
- Data augmentation
- Batch normalization
- Initialization
  - Transfer learning
Training CNN with gradient descent

• A CNN as composition of functions
  \[ f_w(x) = f_L(\ldots (f_2(f_1(x; w_1); w_2) \ldots; w_L) \]

• Parameters
  \[ w = (w_1, w_2, \ldots, w_L) \]

• Empirical loss function
  \[ L(w) = \frac{1}{n} \sum_i l(z_i, f_w(x_i)) \]

• Gradient descent
  \[ w^{t+1} = w^t - \eta_t \frac{\partial f}{\partial w}(w^t) \]
An Illustrative example

\[ f(x, y) = xy, \quad \frac{\partial f}{\partial x} = y, \frac{\partial f}{\partial y} = x \]

Example: \( x = 4, y = -3 \Rightarrow f(x, y) = -12 \)

**Partial derivatives**

\[ \frac{\partial f}{\partial x} = -3, \quad \frac{\partial f}{\partial y} = 4 \]

**Gradient**

\[ \nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right] \]
\[ f(x, y, z) = (x + y)z = qz \]

\[
\begin{align*}
q &= x + y \\
\frac{\partial q}{\partial x} &= 1, \\
\frac{\partial q}{\partial y} &= 1
\end{align*}
\]

\[
\begin{align*}
f &= qz \\
\frac{\partial f}{\partial q} &= z, \\
\frac{\partial f}{\partial z} &= q
\end{align*}
\]

Goal: compute the gradient

\[
\nabla f = \left[ \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right]
\]
\[ f(x, y, z) = (x + y)z = qz \]

\[
q = x + y \\
\frac{\partial q}{\partial x} = 1, \quad \frac{\partial q}{\partial y} = 1
\]

\[
f = qz \\
\frac{\partial f}{\partial q} = z, \quad \frac{\partial f}{\partial z} = q
\]

**Chain rule:**
\[
\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x}
\]

---

# set some inputs
\( x = -2; \, y = 5; \, z = -4 \)

# perform the forward pass
\( q = x + y \) \# \( q \) becomes 3
\( f = q \times z \) \# \( f \) becomes -12

# perform the backward pass (backpropagation) in reverse order:
# first backprop through \( f = q \times z \)
\( dfdz = q \) \# \( df/dz = q \), so gradient on \( z \) becomes 3
\( dfdq = z \) \# \( df/dq = z \), so gradient on \( q \) becomes -4

# now backprop through \( q = x + y \)
\( dfdx = 1.0 \times dfdq \) \# \( dq/dx = 1 \). And the multiplication here is the chain rule!
\( dfdy = 1.0 \times dfdq \) \# \( dq/dy = 1 \)

---

Example credit: Andrej Karpathy
Backpropagation (recursive chain rule)

\[
\frac{\partial f}{\partial w_i} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial w_i} \frac{\partial f}{\partial q}
\]

Gate gradient
Local gradient

Can be computed during forward pass
The gate receives this during backprop
Dropout

Intuition: successful conspiracies
- 50 people planning a conspiracy

- Strategy A: plan a big conspiracy involving 50 people
  - Likely to fail. 50 people need to play their parts correctly.

- Strategy B: plan 10 conspiracies each involving 5 people
  - Likely to succeed!

Dropout: A simple way to prevent neural networks from overfitting [Srivastava JMLR 2014]
Dropout

**Main Idea**: approximately combining exponentially many different neural network architectures efficiently

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-1 (val)</th>
<th>Top-5 (val)</th>
<th>Top-5 (test)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM on Fisher Vectors of Dense SIFT and Color Statistics</td>
<td>-</td>
<td>-</td>
<td>27.3</td>
</tr>
<tr>
<td>Avg of classifiers over FVs of SIFT, LBP, GIST and CSIFT</td>
<td>-</td>
<td>-</td>
<td>26.2</td>
</tr>
<tr>
<td>Conv Net + dropout (Krizhevsky et al., 2012)</td>
<td>40.7</td>
<td>18.2</td>
<td>-</td>
</tr>
<tr>
<td>Avg of 5 Conv Nets + dropout (Krizhevsky et al., 2012)</td>
<td>38.1</td>
<td>16.4</td>
<td>16.4</td>
</tr>
</tbody>
</table>

Table 6: Results on the ILSVRC-2012 validation/test set.

Dropout: A simple way to prevent neural networks from overfitting [Srivastava JMLR 2014]
Data Augmentation (Jittering)

- Create *virtual* training samples:
  - Horizontal flip
  - Random crop
  - Color casting
  - Geometric distortion

Deep Image [Wu et al. 2015]
Parametric Rectified Linear Unit

\[ f(y) = \begin{cases} 
  y & \text{if } y > 0 \\
  0 & \text{otherwise} 
\end{cases} \]

\[ f(y) = ay \]

<table>
<thead>
<tr>
<th>in competition</th>
<th>team</th>
<th>top-5 (test)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VGG [25]</td>
<td>7.32</td>
</tr>
<tr>
<td></td>
<td>GoogLeNet [29]</td>
<td>6.66</td>
</tr>
<tr>
<td>post-competition</td>
<td>VGG [25] (arXiv v5)</td>
<td>6.8</td>
</tr>
<tr>
<td></td>
<td>Baidu [32]</td>
<td>5.98</td>
</tr>
<tr>
<td></td>
<td>MSRA, PReLU-nets</td>
<td><strong>4.94</strong></td>
</tr>
</tbody>
</table>

Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification [He et al. 2015]
Batch Normalization

Input: Values of $x$ over a mini-batch: $\mathcal{B} = \{x_1...m\}$; Parameters to be learned: $\gamma, \beta$

Output: $\{y_i = \text{BN}_{\gamma,\beta}(x_i)\}$

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

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

$$
\tilde{x}_i \leftarrow \frac{x_i - \mu_\mathcal{B}}{\sqrt{\sigma^2_\mathcal{B} + \epsilon}} \quad \text{// normalize}
$$

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

Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [Ioffe and Szegedy 2015]
Things to remember

• Visual categorization help transfer knowledge

• Convolutional neural networks
  • A cascade of conv + ReLU + pool
  • Representation learning
  • Advanced architectures
  • Tricks for training CNN