Introduction to Deep Learning and its applications in Computer Vision

Xiaogang Wang
Department of Electronic Engineering, The Chinese University of Hong Kong
Outline

• Introduction to deep learning
• Deep learning for object recognition
• Deep learning for object segmentation
• Deep learning for object detection
• Open questions and future works
Part I: Introduction to Deep Learning

- Historical review of deep learning
- Introduction to classical deep models
- Why does deep learning work?
Machine Learning

\[ x \xrightarrow{} F(x) \xrightarrow{} y \]

Class label (Classification)

Vector (Estimation)

\{dog, cat, horse, flower, \ldots\}

Object recognition

Super resolution

High-resolution image

Low-resolution image
Neural network
Back propagation

Nature

1986

• Solve general learning problems
• Tied with biological system
Neural network
Backpropagation

1986

\[ g(x) = f(\sum_{i=1}^{d} x_i w_i + w_0) = f(w^T x) \]
Neural network
Back propagation

1986

• Solve general learning problems
• Tied with biological system

But it is given up...

• Hard to train
• Insufficient computational resources
• Small training sets
• Does not work well
Neural network
Back propagation

Nature

1986 2006

• SVM
• Boosting
• Decision tree
• KNN
• ...

• Flat structures
• Loose tie with biological systems
• Specific methods for specific tasks
  – Hand crafted features (GMM-HMM, SIFT, LBP, HOG)

Kruger et al. TPAMI’13
- Unsupervised & Layer-wised pre-training
- Better designs for modeling and training (normalization, nonlinearity, dropout)
- New development of computer architectures
  - GPU
  - Multi-core computer systems
- Large scale databases

Big Data!
Machine Learning with Big Data

• Machine learning with small data: overfitting, reducing model complexity (capacity)
• Machine learning with big data: underfitting, increasing model complexity, optimization, computation resource
How to increase model capacity?

Curse of dimensionality

↓

Blessing of dimensionality

↓

Learning hierarchical feature transforms
(Learning features with deep structures)

Deep Networks Advance State of Art in Speech

Deep Learning leads to breakthrough in speech recognition at MSR.
Neural network
Back propagation

Deep belief net
Science

Speech

IMAGENET

1986 2006 2011 2012

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>Error rate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>U. Toronto</td>
<td>0.15315</td>
<td>Deep learning</td>
</tr>
<tr>
<td>2</td>
<td>U. Tokyo</td>
<td>0.26172</td>
<td>Hand-crafted features and learning models.</td>
</tr>
<tr>
<td>3</td>
<td>U. Oxford</td>
<td>0.26979</td>
<td>Bottleneck.</td>
</tr>
<tr>
<td>4</td>
<td>Xerox/INRIA</td>
<td>0.27058</td>
<td></td>
</tr>
</tbody>
</table>

Object recognition over 1,000,000 images and 1,000 categories (2 GPU)

Examples from ImageNet

1000 object classes that we recognize

poster created by Fengjun Lv using VIPBase

• ImageNet 2013 – image classification challenge

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>Error rate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>NYU</td>
<td>0.11197</td>
<td>Deep learning</td>
</tr>
<tr>
<td>2</td>
<td>NUS</td>
<td>0.12535</td>
<td>Deep learning</td>
</tr>
<tr>
<td>3</td>
<td>Oxford</td>
<td>0.13555</td>
<td>Deep learning</td>
</tr>
</tbody>
</table>

MSRA, IBM, Adobe, NEC, Clarifai, Berkley, U. Tokyo, UCLA, UIUC, Toronto … Top 20 groups all used deep learning

• ImageNet 2013 – object detection challenge

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>Mean Average Precision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>UvA-Euvison</td>
<td>0.22581</td>
<td>Hand-crafted features</td>
</tr>
<tr>
<td>2</td>
<td>NEC-MU</td>
<td>0.20895</td>
<td>Hand-crafted features</td>
</tr>
<tr>
<td>3</td>
<td>NYU</td>
<td>0.19400</td>
<td>Deep learning</td>
</tr>
</tbody>
</table>
• **ImageNet 2014 – Image classification challenge**

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>Error rate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Google</td>
<td>0.06656</td>
<td>Deep learning</td>
</tr>
<tr>
<td>2</td>
<td>Oxford</td>
<td>0.07325</td>
<td>Deep learning</td>
</tr>
<tr>
<td>3</td>
<td>MSRA</td>
<td>0.08062</td>
<td>Deep learning</td>
</tr>
</tbody>
</table>

• **ImageNet 2014 – object detection challenge**

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>Mean Average Precision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Google</td>
<td>0.43933</td>
<td>Deep learning</td>
</tr>
<tr>
<td>2</td>
<td>CUHK</td>
<td>0.40656</td>
<td>Deep learning</td>
</tr>
<tr>
<td>3</td>
<td>DeepInsight</td>
<td>0.40452</td>
<td>Deep learning</td>
</tr>
<tr>
<td>4</td>
<td>UvA-Euvision</td>
<td>0.35421</td>
<td>Deep learning</td>
</tr>
<tr>
<td>5</td>
<td>Berkley Vision</td>
<td>0.34521</td>
<td>Deep learning</td>
</tr>
</tbody>
</table>
• ImageNet 2014 – object detection challenge

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Model average</td>
<td>0.439</td>
<td><strong>0.439</strong></td>
<td>0.405</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Single model</td>
<td>0.380</td>
<td><strong>0.427</strong></td>
<td>0.402</td>
<td>0.354</td>
<td>0.345</td>
<td>0.314</td>
</tr>
</tbody>
</table>

• Google and Baidu announced their deep learning based visual search engines (2013)
  – Google
    • “on our test set we saw double the average precision when compared to other approaches we had tried. We acquired the rights to the technology and went full speed ahead adapting it to run at large scale on Google’s computers. We took cutting edge research straight out of an academic research lab and launched it, in just a little over six months.”
  – Baidu
• Deep learning achieves 99.47% face verification accuracy on Labeled Faces in the Wild (LFW), higher than human performance


Labeled Faces in the Wild (2007)

Best results without deep learning

Random guess (50%)
Eigenface (60%)

TL Joint Bayesian (96.33%), 2013
Human cropped (97.53%)

Human funneled (99.20%)
**Our deep learning result (99.47%)**
<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy ± Standard Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute classifiers\textsuperscript{11}</td>
<td>0.8525 ± 0.0060</td>
</tr>
<tr>
<td>Simile classifiers\textsuperscript{11}</td>
<td>0.8414 ± 0.0041</td>
</tr>
<tr>
<td>Attribute and Simile classifiers\textsuperscript{11}</td>
<td>0.8554 ± 0.0035</td>
</tr>
<tr>
<td>Multiple LE + comp\textsuperscript{14}</td>
<td>0.84*5 ± 0.0040</td>
</tr>
<tr>
<td>Associate-Predict\textsuperscript{18}</td>
<td>0.9057 ± 0.0056</td>
</tr>
<tr>
<td>Tom-vs-Pete\textsuperscript{23}</td>
<td>0.9310 ± 0.0135</td>
</tr>
<tr>
<td>Tom-vs-Pete + Attribute\textsuperscript{23}</td>
<td>0.9330 ± 0.0128</td>
</tr>
<tr>
<td>combined Joint Bayesian\textsuperscript{26}</td>
<td>0.9242 ± 0.0108</td>
</tr>
<tr>
<td>high-dim LBP\textsuperscript{27}</td>
<td>0.9517 ± 0.0113</td>
</tr>
<tr>
<td>DFD\textsuperscript{33}</td>
<td>0.8402 ± 0.0044</td>
</tr>
<tr>
<td>TL Joint Bayesian\textsuperscript{34}</td>
<td>0.9633 ± 0.0108</td>
</tr>
<tr>
<td>face.com r2011b\textsuperscript{19}</td>
<td>0.9130 ± 0.0030</td>
</tr>
<tr>
<td>Face++\textsuperscript{40}</td>
<td>0.9727 ± 0.0065</td>
</tr>
<tr>
<td>Deep-face-ensemble\textsuperscript{41}</td>
<td>0.9735 ± 0.0025</td>
</tr>
<tr>
<td>ConvNet-RBM\textsuperscript{42}</td>
<td>0.9252 ± 0.0038</td>
</tr>
<tr>
<td>POOF-gradhist\textsuperscript{44}</td>
<td>0.9313 ± 0.0040</td>
</tr>
<tr>
<td>POOF-HOG\textsuperscript{44}</td>
<td>0.9280 ± 0.0047</td>
</tr>
<tr>
<td>FR+FCN\textsuperscript{45}</td>
<td>0.9645 ± 0.0025</td>
</tr>
<tr>
<td>DeepID\textsuperscript{46}</td>
<td>0.9745 ± 0.0026</td>
</tr>
<tr>
<td>GaussianFace\textsuperscript{47}</td>
<td>0.9852 ± 0.0066</td>
</tr>
<tr>
<td>DeepID\textsuperscript{248}</td>
<td>0.9915 ± 0.0013</td>
</tr>
</tbody>
</table>

Table 6: Mean classification accuracy $\hat{\mu}$ and standard error of the mean $S_{E}$. 
<table>
<thead>
<tr>
<th><strong>10 Breakthrough Technologies 2013</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deep Learning</strong></td>
</tr>
<tr>
<td>With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart.</td>
</tr>
<tr>
<td><strong>Temporary Social Media</strong></td>
</tr>
<tr>
<td>Messages that quickly self-destruct could enhance the privacy of online communications and make people freer to be spontaneous.</td>
</tr>
<tr>
<td><strong>Prenatal DNA Sequencing</strong></td>
</tr>
<tr>
<td>Reading the DNA of fetuses will be the next frontier of the genomic revolution. But do you really want to know about the genetic problems or musical aptitude of your unborn child?</td>
</tr>
<tr>
<td><strong>Additive Manufacturing</strong></td>
</tr>
<tr>
<td>Spraying about 3D printing? GE, the world’s largest manufacturer, is on the verge of using the technology to make jet parts.</td>
</tr>
<tr>
<td><strong>Baxter: The Blue-Collar Robot</strong></td>
</tr>
<tr>
<td>Rodney Brooks’s newest creation is easy to interact with, but the complex innovations behind the robot show just how hard it is to get along with people.</td>
</tr>
<tr>
<td><strong>Memory Implants</strong></td>
</tr>
<tr>
<td>A maverick neuroscientist believes he has deciphered the code by which the brain forms long-term memories. Next: testing a prosthetic implant for people suffering from long-term memory loss.</td>
</tr>
<tr>
<td><strong>Smart Watches</strong></td>
</tr>
<tr>
<td>The designers of the Pebble watch realized that a mobile phone is more useful if you don’t have to take it out of your pocket.</td>
</tr>
<tr>
<td><strong>Ultra-Efficient Solar Power</strong></td>
</tr>
<tr>
<td>Doubling the efficiency of a solar cell would completely change the economics of renewable energy. Nanotechnology just might make it possible.</td>
</tr>
<tr>
<td><strong>Big Data from Cheap Phones</strong></td>
</tr>
<tr>
<td>Collecting and analyzing information from simple cell phones can provide surprising insights into how people move about and behave – and even help us understand the spread of diseases.</td>
</tr>
<tr>
<td><strong>Supergrids</strong></td>
</tr>
<tr>
<td>A new high-power circuit breaker could finally make highly efficient DC power grids practical.</td>
</tr>
</tbody>
</table>
Design Cycle

Domain knowledge

Start:
- Collect data
- Preprocessing
- Feature design
- Choose and design model
- Train classifier
- Evaluation

End

Interest of people working on computer vision, speech recognition, medical image processing, ...

Interest of people working on machine learning

Interest of people working on machine learning and computer vision, speech recognition, medical image processing, ...

Preprocessing and feature design may lose useful information and not be optimized, since they are not parts of an end-to-end learning system.

Preprocessing could be the result of another pattern recognition system.
Person re-identification pipeline

- Pedestrian detection
- Pose estimation
- Body parts segmentation
- Photometric & geometric transform
- Feature extraction
- Classification

Face recognition pipeline

- Face alignment
- Geometric rectification
- Photometric rectification
- Feature extraction
- Classification
Design Cycle with Deep Learning

- Learning plays a bigger role in the design circle
- Feature learning becomes part of the end-to-end learning system
- Preprocessing becomes optional means that several pattern recognition steps can be merged into one end-to-end learning system
- Feature learning makes the key difference
- We underestimated the importance of data collection and evaluation
What makes deep learning successful in computer vision?

Li Fei-Fei

One million images with labels

Evaluation task

Predict 1,000 image categories

Deep learning

Geoffrey Hinton

CNN is not new

Design network structure

New training strategies

Feature learned from ImageNet can be well generalized to other tasks and datasets!
Learning features and classifiers separately

- Not all the datasets and prediction tasks are suitable for learning features with deep models
Deep learning can be treated as a language to described the world with great flexibility.
Introduction to Deep Learning

- Historical review of deep learning
- Introduction to classical deep models
- Why does deep learning work?
Introduction on Classical Deep Models

- **Convolutional Neural Networks (CNN)**

- **Deep Belief Net (DBN)**

- **Auto-encoder**
Classical Deep Models

• Convolutional Neural Networks (CNN)
  – First proposed by Fukushima in 1980
  – Improved by LeCun, Bottou, Bengio and Haffner in 1998
Backpropagation

\[ W \leftarrow W - \eta \nabla J(W) \]

\( W \) is the parameter of the network; \( J \) is the objective function

Classical Deep Models

- **Deep belief net**
  - Hinton’06

  \[ P(x,h_1,h_2) = p(x|h_1) p(h_1,h_2) \]

  \[ P(x,h_1) = \frac{e^{-E(x,h_1)}}{\sum_{x,h_1} e^{-E(x,h_1)}} \]

  \[ E(x,h_1) = b'x + c'h_1 + h_1'Wx \]
Classical Deep Models

• Auto-encoder
  – Hinton and Salakhutdinov 2006

Encoding: \( h_1 = \sigma(W_1x + b_1) \)
\( h_2 = \sigma(W_2h_1 + b_2) \)

Decoding: \( \tilde{h}_1 = \sigma(W'_{2}h_2 + b_3) \)
\( \tilde{x} = \sigma(W'_{1}h_1 + b_4) \)
Introduction to Deep Learning

• Historical review of deep learning
• Introduction to classical deep models
• Why does deep learning work?
Feature Learning vs Feature Engineering
Feature Engineering

• The performance of a pattern recognition system heavily depends on feature representations
• Manually designed features dominate the applications of image and video understanding in the past
  – Reply on human domain knowledge much more than data
  – Feature design is separate from training the classifier
  – If handcrafted features have multiple parameters, it is hard to manually tune them
  – Developing effective features for new applications is slow
Handcrafted Features for Face Recognition

- Geometric features
- Pixel vector
- Gabor filters: 2 parameters
- Local binary patterns: 3 parameters

Timeline:
- 1980s
- 1992
- 1997
- 2006
Feature Learning

• Learning transformations of the data that make it easier to extract useful information when building classifiers or predictors
  – Jointly learning feature transformations and classifiers makes their integration optimal
  – Learn the values of a huge number of parameters in feature representations
  – Faster to get feature representations for new applications
  – Make better use of big data
Deep Learning Means Feature Learning

- Deep learning is about learning hierarchical feature representations

\[ y = F(W^k \cdot F(W^{k-1} \cdot F(\ldots F(W^0 \cdot x))) \]

- Good feature representations should be able to disentangle multiple factors coupled in the data

Identity: face recognition
Pose: pose estimation
Expression: expression recognition
Age: age estimation

Data → Trainable Feature Transform → Trainable Feature Transform → Trainable Feature Transform → Trainable Feature Transform → ... → Classifier
Deep Learning Means Feature Learning

- How to effectively learn features with deep models
  - With challenging tasks
  - Predict high-dimensional vectors

Training stage A

Dataset A

(feature transform)

Classifier A

Distinguish 1000 categories

Training stage B

Dataset B

(feature transform)

Classifier B

Distinguish 201 categories

Training stage C

Dataset C

(feature transform)

SVM

Distinguish one object class from all the negatives
Example 1: deep learning generic image features

- Hinton group’s groundbreaking work on ImageNet
  - They did not have much experience on general image classification on ImageNet
  - It took one week to train the network with 60 Million parameters
  - The learned feature representations are effective on other datasets (e.g. Pascal VOC) and other tasks (object detection, segmentation, tracking, and image retrieval)
96 learned low-level filters
Image classification result
Top hidden layer can be used as feature for retrieval
Example 2: deep learning face identity features by recovering canonical-view face images

Reconstruction examples from LFW

• Deep model can disentangle hidden factors through feature extraction over multiple layers
• No 3D model; no prior information on pose and lighting condition
• Model multiple complex transforms
• Reconstructing the whole face is a much stronger supervision than predicting 0/1 class label and helps to avoid overfitting

Arbitrary view

Canonical view
## Comparison on Multi-PIE

<table>
<thead>
<tr>
<th>Method</th>
<th>-45°</th>
<th>-30°</th>
<th>-15°</th>
<th>+15°</th>
<th>+30°</th>
<th>+45°</th>
<th>Avg</th>
<th>Pose</th>
</tr>
</thead>
<tbody>
<tr>
<td>LGBP [26]</td>
<td>37.7</td>
<td>62.5</td>
<td>77</td>
<td>83</td>
<td>59.2</td>
<td>36.1</td>
<td>59.3</td>
<td>✓</td>
</tr>
<tr>
<td>VAAM [17]</td>
<td>74.1</td>
<td>91</td>
<td>95.7</td>
<td>95.7</td>
<td>89.5</td>
<td>74.8</td>
<td>86.9</td>
<td>✓</td>
</tr>
<tr>
<td>FA-EGFC[3]</td>
<td>84.7</td>
<td>95</td>
<td>99.3</td>
<td>99</td>
<td>92.9</td>
<td>85.2</td>
<td>92.7</td>
<td>x</td>
</tr>
<tr>
<td>SA-EGFC[3]</td>
<td>93</td>
<td>98.7</td>
<td>99.7</td>
<td>99.7</td>
<td>93.6</td>
<td>97.2</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>LE[4] + LDA</td>
<td>86.9</td>
<td>95.5</td>
<td>99.9</td>
<td>99.7</td>
<td>95.5</td>
<td>81.8</td>
<td>93.2</td>
<td>x</td>
</tr>
<tr>
<td>CRBM[9] + LDA</td>
<td>80.3</td>
<td>90.5</td>
<td>94.9</td>
<td>96.4</td>
<td>88.3</td>
<td>89.8</td>
<td>87.6</td>
<td>x</td>
</tr>
<tr>
<td>Ours</td>
<td>95.6</td>
<td>98.5</td>
<td>100.0</td>
<td>99.3</td>
<td>98.5</td>
<td>97.8</td>
<td>98.3</td>
<td>x</td>
</tr>
</tbody>
</table>

---


Deep learning 3D model from 2D images, mimicking human brain activities

Training stage A:
- Face images in arbitrary views
  - Face identity features
    - Regressor 1
    - Regressor 2
    - ... 
    - Reconstruct view 1
    - Reconstruct view 2
    - ... 

Deep learning

Training stage B:
- Two face images in arbitrary views
  - Feature transform
    - Linear Discriminant analysis
    - The two images belonging to the same person or not

Face reconstruction

Face verification
Example 3: deep learning face identity features from predicting 10,000 classes

- At training stage, each input image is classified into 10,000 identities with 160 hidden identity features in the top layer.
- The hidden identity features can be well generalized to other tasks (e.g. verification) and identities outside the training set.
- As adding the number of classes to be predicted, the generalization power of the learned features also improves.

Dataset A
Training stage A

Dataset B
Training stage B

**feature transform**

Classifier A

Distinguish 10,000 people

Face identification

Fixed

**feature transform**

Linear classifier B

The two images belonging to the same person or not

Face verification

**The two images belonging to the same person or not**
Deep Structures vs Shallow Structures
(Why deep?)
Shallow Structures

- A three-layer neural network (with one hidden layer) can approximate any classification function.
- Most machine learning tools (such as SVM, boosting, and KNN) can be approximated as neural networks with one or two hidden layers.
- Shallow models divide the feature space into regions and match templates in local regions. $O(N)$ parameters are needed to represent $N$ regions.

$$g(x) = b + \sum_i \alpha_i K(x, x_i)$$
Deep Machines are More Efficient for Representing Certain Classes of Functions

- Theoretical results show that an architecture with insufficient depth can require many more computational elements, potentially exponentially more (with respect to input size), than architectures whose depth is matched to the task (Hastad 1986, Hastad and Goldmann 1991)
- It also means many more parameters to learn
• Take the d-bit parity function as an example

\[(X_1, \ldots, X_d) \in \{0, 1\}^d \mapsto \begin{cases} 1, & \text{if } \sum_{i=1}^{d} X_i \text{ is even} \\ -1, & \text{otherwise} \end{cases}\]

• d-bit logical parity circuits of depth 2 have exponential size (Andrew Yao, 1985)

• There are functions computable with a polynomial-size logic gates circuits of depth k that require exponential size when restricted to depth k -1 (Hastad, 1986)
• Architectures with multiple levels naturally provide sharing and re-use of components
Humans Understand the World through Multiple Levels of Abstractions

- We do not interpret a scene image with pixels
  - Objects (sky, cars, roads, buildings, pedestrians) -> parts (wheels, doors, heads) -> texture -> edges -> pixels
  - Attributes: blue sky, red car
- It is natural for humans to decompose a complex problem into sub-problems through multiple levels of representations
Humans Understand the World through Multiple Levels of Abstractions

• Humans learn abstract concepts on top of less abstract ones
• Humans can imagine new pictures by re-configuring these abstractions at multiple levels. Thus our brain has good generalization can recognize things never seen before.
  – Our brain can estimate shape, lighting and pose from a face image and generate new images under various lightings and poses. That’s why we have good face recognition capability.
Local and Global Representations

Global representation

Blue eyes? (1/0)

Big nose? (1/0)

Yellow skin? (1/0)

Local representation

Oriental face

Occidental face
Human Brains Process Visual Signals through Multiple Layers

• A visual cortical area consists of six layers (Kruger et al. 2013)
Joint Learning vs Separate Learning

Data collection → Preprocessing step 1 → Preprocessing step 2 → Feature extraction → Classification

Data collection → Feature transform → Feature transform → Feature transform → Classification

End-to-end learning

Deep learning is a framework/language but not a black-box model

Its power comes from joint optimization and increasing the capacity of the learner
• Domain knowledge could be helpful for designing new deep models and training strategies
• How to formulate a vision problem with deep learning?
  – Make use of experience and insights obtained in CV research
  – Sequential design/learning vs **joint learning**
  – Effectively train a deep model (layerwise pre-training + fine tuning)
What if we treat an existing deep model as a black box in pedestrian detection?

ConvNet–U–MS

Results on Caltech Test

Results on ETHZ


Our Joint Deep Learning Model

Modeling Part Detectors

- Design the filters in the second convolutional layer with variable sizes.
Deformation Layer

- Summed map
  - Global max pooling
  - Part score

- Part detection map
  - $M_p$

- Deformation maps
  - $D_{1,p}$
  - $D_{2,p}$
  - $D_{3,p}$
  - $D_{4,p}$
Visibility Reasoning with Deep Belief Net

Correlates with part detection score
Experimental Results

- Caltech – Test dataset (largest, most widely used)
Experimental Results

- Caltech – Test dataset (largest, most widely used)
Experimental Results

- Caltech – Test dataset (largest, most widely used)

Histograms of oriented gradients for human detection
N Dalal, B Triggs - ... and Pattern Recognition, 2005. CVPR 2005 ... 2005 - ieeexplore.ieee.org
... We study the issue of feature sets for human detection, showing that locally normalized Histogram of Oriented Gradient (HOG) descriptors provide excellent performance relative to other existing feature sets including wavelets [17, 22]. ...
Experimental Results

- Caltech – Test dataset (largest, most widely used)

![Graph showing experimental results]

Object detection with discriminatively trained part-based models
PF Felzenszwalb, RB Girshick... - Pattern Analysis and ..., 2010 - ieeexplore.ieee.org

Abstract
We describe an object detection system based on mixtures of multiscale deformable part models. Our system is able to represent highly variable object classes and achieves state-of-the-art results in the PASCAL object detection challenges. While...
Experimental Results

- Caltech – Test dataset (largest, most widely used)


Large learning capacity makes high dimensional data transforms possible, and makes better use of contextual information
• How to make use of the large learning capacity of deep models?
  – High dimensional data transform
  – Hierarchical nonlinear representations
Face Parsing

Motivations

- Recast face segmentation as a cross-modality data transformation problem
- Cross modality autoencoder
- Data of two different modalities share the same representations in the deep model
- Deep models can be used to learn shape priors for segmentation
Training Segmentators

- **(b)** training segmentator: one-layer denoising autoencoder
- **(c)** training segmentator: deep autoencoder
- **(d)** testing segmentator

\[ \mathcal{L}_{\mathcal{H}}([\bar{I}_b \Lambda], [\tilde{I}_b \tilde{\Lambda}]) \]
Big data

Challenging supervision task with rich predictions

Rich information

How to make use of it?

Capacity

Hierarchical feature learning

Reduce capacity

Joint optimization

Capture contextual information

Domain knowledge

Make learning more efficient

Take large input

Go deeper

Go wider
Summary

- Automatically learns hierarchical feature representations from data and disentangles hidden factors of input data through multi-level nonlinear mappings.
- For some tasks, the expressive power of deep models increases exponentially as their architectures go deep.
- Jointly optimize all the components in a vision and create synergy through close interactions among them.
- Benefitting the large learning capacity of deep models, we also recast some classical computer vision challenges as high-dimensional data transform problems and solve them from new perspectives.
- It is more effective to train deep models with challenging tasks and rich predictions.
References

Outline

• Introduction to deep learning
• **Deep learning for object recognition**
• Deep learning for object segmentation
• Deep learning for object detection
• Open questions and future works
Part II: Deep Learning Object Recognition

- Deep learning for object recognition on ImageNet
- Deep learning for face recognition
  - Learn identity features from joint verification-identification signals
  - Learn 3D face models from 2D images
CNN for Object Recognition on ImageNet

- Krizhevsky, Sutskever, and Hinton, NIPS 2012
- Trained on one million images of 1000 categories collected from the web with two GPUs; 2GB RAM on each GPU; 5GB of system memory
- Training lasts for one week

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>Error rate</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>U. Toronto</td>
<td>0.15315</td>
<td>Deep learning</td>
</tr>
<tr>
<td>2</td>
<td>U. Tokyo</td>
<td>0.26172</td>
<td>Hand-crafted features and learning models. Bottleneck.</td>
</tr>
<tr>
<td>3</td>
<td>U. Oxford</td>
<td>0.26979</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Xerox/INRIA</td>
<td>0.27058</td>
<td></td>
</tr>
</tbody>
</table>
Model Architecture

- Max-pooling layers follow 1\textsuperscript{st}, 2\textsuperscript{nd}, and 5\textsuperscript{th} convolutional layers
- The number of neurons in each layer is given by 253440, 186624, 64896, 43264, 4096, 4096, 1000
- 650000 neurons, 60 million parameters, 630 million connections
Normalization

• Normalize the input by subtracting the mean image on the training set

Input image (256 x 256) ─ Mean image

Krizhevsky 2012
Activation Function

- Rectified linear unit leads to sparse responses of neurons, such that weights can be effectively updated with BP.
Data Augmentation

- The neural net has 60M parameters and it overfits
- Image regions are randomly cropped with shift; their horizontal reflections are also included

Krizhevsky 2012
Dropout

• Randomly set some input features and the outputs of hidden units as zero during the training process
• Feature co-adaptation: a feature is only helpful when other specific features are present
  – Because of the existence of noise and data corruption, some features or the responses of hidden nodes can be misdetected
• Dropout prevents feature co-adaptation and can significantly improve the generalization of the trained network
• Can be considered as another approach to regularization
• It can be viewed as averaging over many neural networks
• Slower convergence
Classification Result

Krizhevsky 2012
### Detection Result

<table>
<thead>
<tr>
<th>Object</th>
<th>Detected Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>bookshop</td>
<td>balance beam, cinema, marimba, parallel bars, computer keyboard</td>
</tr>
<tr>
<td>coyote</td>
<td>grey fox, kit fox, red fox, coyote, dhole</td>
</tr>
<tr>
<td>cradle</td>
<td>cradle, bassinet, diaper, crib, bath towel</td>
</tr>
<tr>
<td>wood rabbit</td>
<td>hare, wood rabbit, grey fox, coyote, wallaby</td>
</tr>
<tr>
<td>bottlecap</td>
<td>bottlecap, magnetic compass, puck, stopwatch, disk brake</td>
</tr>
<tr>
<td>harvester</td>
<td>harvester, thresher, plow, tractor, tow truck</td>
</tr>
<tr>
<td>garter snake</td>
<td>diamondback, leatherback turtle, sandbar, echidna, armadillo</td>
</tr>
<tr>
<td>Walker hound</td>
<td>beagle, Walker hound, English foxhound, muzzle, Italian greyhound</td>
</tr>
</tbody>
</table>
Image Retrieval

Krizhevsky 2012
Adaptation to Smaller Datasets

- Directly use the feature representations learned from ImageNet and replace handcrafted features with them in image classification, scene recognition, fine-grained object recognition, attribute recognition, image retrieval (Razavian et al. 2014, Gong et al. 2014)
- Use ImageNet to pre-train the model (good initialization), and use target dataset to fine-tune it (Girshick et al. CVPR 2014)
- Fix the bottom layers and only fine tune the top layers
GoogLeNet

- More than 20 layers
- Add supervision at multiple layers
- The error rate is reduced from 15.3% to 6.6%
Deep Learning Object Recognition

• Deep learning for object recognition on ImageNet

• Deep learning for face recognition
  – Learn identity features from joint verification-identification signals
  – Learn 3D face models from 2D images
### Deep Learning Results on LFW

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
<th># points</th>
<th># training images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Huang et al. CVPR’12</td>
<td>87%</td>
<td>3</td>
<td>Unsupervised</td>
</tr>
<tr>
<td>Sun et al. ICCV’13</td>
<td>92.52%</td>
<td>5</td>
<td>87,628</td>
</tr>
<tr>
<td>DeepFace (CVPR’14)</td>
<td>97.35%</td>
<td>6 + 67</td>
<td>7,000,000</td>
</tr>
<tr>
<td>Sun et al. (CVPR’14)</td>
<td>97.45%</td>
<td>5</td>
<td>202,599</td>
</tr>
<tr>
<td>Sun et al. (NIPS’14)</td>
<td>99.15%</td>
<td>18</td>
<td>202,599</td>
</tr>
<tr>
<td><strong>New: DeepID2+ (CVPR’15)</strong></td>
<td><strong>99.47%</strong></td>
<td><strong>18</strong></td>
<td><strong>450,000</strong></td>
</tr>
</tbody>
</table>

- The first deep learning work on face recognition was done by Huang et al. in 2012. With unsupervised learning, the accuracy was 87%.
- Our work at ICCV’13 achieved result (92.52%) comparable with state-of-the-art.
- Our work at CVPR’14 reached **97.45%** close to “human cropped” performance (**97.53%**).
- DeepFace developed by Facebook also at CVPR’14 used 73-point 3D face alignment and 7 million training data (35 times larger than us).
- Our most recent work reached **99.15%** close to “human funneled” performance (**99.20%**).

Closed- and open-set face identification on LFW

<table>
<thead>
<tr>
<th>Method</th>
<th>Rank-1 (%)</th>
<th>DIR @ 1% FAR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>COST-S1 [1]</td>
<td>56.7</td>
<td>25</td>
</tr>
<tr>
<td>COST-S1+s2 [1]</td>
<td>66.5</td>
<td>35</td>
</tr>
<tr>
<td>DeepFace [2]</td>
<td>64.9</td>
<td>44.5</td>
</tr>
<tr>
<td>DeepFace+ [3]</td>
<td>82.5</td>
<td>61.9</td>
</tr>
<tr>
<td>DeepID2 [4]</td>
<td>91.1</td>
<td>61.6</td>
</tr>
<tr>
<td>DeepID2+ [5]</td>
<td>95.0</td>
<td>80.7</td>
</tr>
</tbody>
</table>

Eternal Topic on Face Recognition

Intra-personal variation

Inter-personal variation

How to separate the two types of variations?
Are they the same person or not?

Nicole Kidman

Nicole Kidman
Are they the same person or not?

Coo d’Este
Melina Kanakaredes
Are they the same person or not?

Elijah Wood

Stefano Gabbana
Are they the same person or not?
Are they the same person or not?

Jacqueline Obradors  Julie Taymor
• Out of 6000 image pairs on the LFW test set, 51 pairs are misclassified with the deep model.

• We randomly mixed them and presented them to 10 Chinese subjects for evaluation. Their averaged verification accuracy is 56%, close to random guess (50%).
Linear Discriminate Analysis

\[ W^* = \arg \max_W \frac{|W'S_bW|}{|W'S_wW|} \]

\[ S_b = \sum n_k (\bar{x}_k - \bar{x})(\bar{x}_k - \bar{x})^t \propto \sum (\bar{x}_k - \bar{x}_k')(\bar{x}_k - \bar{x}_k')^t \]

\[ S_w = \sum \sum (x_i - \bar{x}_k)(x_i - \bar{x}_k)^t \propto \sum (x_i - x_j)(x_i - x_j)^t \]
\[ W^* = \arg \max_W |W^T S_h W | \quad s.t. \quad |W^T S_y W | = 1 \]

LDA seeks for linear feature mapping which maximizes the distance between class centers under the constraint what the intrapersonal variation is constant

\[ y_i = f(x_i) = W^T x_i \]

\[ f^* = \arg \max_{f'} \sum_{k,k'} |f(\bar{x}_k) - f(\bar{x}_{k'})|^2 \]

\[ s.t. \sum_{(i,j) \in \Omega} |f(x_i) - f(x_j)|^2 = 1 \]
Deep Learning for Face Recognition

• Extract identity preserving features through hierarchical nonlinear mappings
• Model complex intra- and inter-personal variations with large learning capacity
Learn Identity Features from Different Supervisory Tasks

- Face identification: classify an image into one of \( N \) identity classes
  - multi-class classification problem
- Face verification: verify whether a pair of images belong to the same identity or not
  - binary classification problem
Minimize the intra-personal variation under the constraint that the distance between classes is constant (i.e. contracting the volume of the image space without reducing the distance between classes)

\[
y = f(x); \quad g = \text{softmax}()
\]

\[
f^* = \arg\min_f \sum_{(i,j) \in \Omega_I} ||f(x_i) - f(x_j)||^2
\]

s.t. \( |g(f(x_i)) - g(f(x_j))| = 1, \; \text{label}(x_i) \neq \text{label}(x_j) \)
Learn Identity Features with Verification Signal

- Extract relational features with learned filter pairs
  \[ y^j = f \left( b^j + k_1^j \ast x^1 + k_2^j \ast x^2 \right) \]
- These relational features are further processed through multiple layers to extract global features
- The fully connected layer can be used as features to combine with multiple ConvNets

Results on LFW

- Unrestricted protocol without outside training data

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConvNet-RBM previous [43]</td>
<td>91.75 ± 0.48</td>
</tr>
<tr>
<td>VMRS [3]</td>
<td>92.05 ± 0.45</td>
</tr>
<tr>
<td>CMD+SLBP [23]</td>
<td>92.58 ± 1.36</td>
</tr>
<tr>
<td>VisionLabs ver. 1.0 [1]</td>
<td>92.90 ± 0.31</td>
</tr>
<tr>
<td>Fisher vector faces [41]</td>
<td>93.03 ± 1.05</td>
</tr>
<tr>
<td>High-dim LBP [13]</td>
<td>93.18 ± 1.07</td>
</tr>
<tr>
<td>Aurora [19]</td>
<td>93.24 ± 0.44</td>
</tr>
<tr>
<td>ConvNet-RBM</td>
<td><strong>93.83 ± 0.52</strong></td>
</tr>
</tbody>
</table>
Results on LFW

- Unrestricted protocol using outside training data

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joint Bayesian [12]</td>
<td>92.42 ± 1.08</td>
</tr>
<tr>
<td>ConvNet-RBM previous [43]</td>
<td>92.52 ± 0.38</td>
</tr>
<tr>
<td>Tom-vs-Pete (with attributes) [4]</td>
<td>93.30 ± 1.28</td>
</tr>
<tr>
<td>High-dim LBP [13]</td>
<td>95.17 ± 1.13</td>
</tr>
<tr>
<td>TL Joint Bayesian [10]</td>
<td>96.33 ± 1.08</td>
</tr>
<tr>
<td>ConvNet-RBM</td>
<td>97.08 ± 0.28</td>
</tr>
</tbody>
</table>
DeepID: Learn Identity Features with Identification Signal

• During training, each image is classified into 10,000 identities with 160 identity features in the top layer
• These features keep rich inter-personal variations
• Features from the last two convolutional layers are effective
• The hidden identity features can be well generalized to other tasks (e.g. verification) and identities outside the training set
• High-dimensional prediction is more challenging, but also adds stronger supervision to the network.
• As adding the number of classes to be predicted, the generalization power of the learned features also improves.
Extract Features from Multiple ConvNets
Learn Identity Features with Identification Signal

- After combining hidden identity features from multiple CovNets and further reducing dimensionality with PCA, each face image has 150-dimensional features as signature.
- These features can be further processed by other classifiers in face verification. Interestingly, we find Joint Bayesian is more effective than cascading another neural network to classify these features.
DeepID2: Joint Identification-Verification Signals

- Every two feature vectors extracted from the same identity should be close to each other

\[
\text{Verif}(f_i, f_j, y_{ij}, \theta_{ve}) = \begin{cases} 
\frac{1}{2} \|f_i - f_j\|_2^2 & \text{if } y_{ij} = 1 \\
\frac{1}{2} \max(0, m - \|f_i - f_j\|_2)^2 & \text{if } y_{ij} = -1
\end{cases}
\]

- \(f_i\) and \(f_j\) are feature vectors extracted from two face images in comparison
- \(y_{ij} = 1\) means they are from the same identity; \(y_{ij} = -1\) means different identities
- \(m\) is a margin to be learned

Balancing Identification and Verification Signals with Parameter $\lambda$

$\lambda = 0$: only identification signal
$\lambda = +\infty$: only verification signal
Rich Identity Information Improves Feature Learning

- Face verification accuracies with the number of training identities
Summary of DeepID2

• 25 face regions at different scales and locations around landmarks are selected to build 25 neural networks
• All the 160 X 25 hidden identity features are further compressed into a 180-dimensional feature vector with PCA as a signature for each image
• With a single Titan GPU, the feature extraction process takes 35ms per image
DeepID2+

- Larger network structures
- Larger training data
- Adding supervisory signals at every layer

Compare DeepID2 and DeepID2+ on LFW

Comparison of face verification accuracies on LFW with ConvNets trained on 25 face regions given in DeepID2

Best single model is improved from 96.72% to 98.70%
Final Result on LFW

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy (%)</td>
<td>95.17</td>
<td>96.33</td>
<td>97.35</td>
<td>97.45</td>
<td>99.15</td>
<td>99.47</td>
</tr>
</tbody>
</table>


Closed- and open-set face identification on LFW

<table>
<thead>
<tr>
<th>Method</th>
<th>Rank-1 (%)</th>
<th>DIR @ 1% FAR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>COST-S1 [1]</td>
<td>56.7</td>
<td>25</td>
</tr>
<tr>
<td>COST-S1+s2 [1]</td>
<td>66.5</td>
<td>35</td>
</tr>
<tr>
<td>DeepFace [2]</td>
<td>64.9</td>
<td>44.5</td>
</tr>
<tr>
<td>DeepFace+ [3]</td>
<td>82.5</td>
<td>61.9</td>
</tr>
<tr>
<td>DeepID2</td>
<td>91.1</td>
<td>61.6</td>
</tr>
<tr>
<td>DeepID2+</td>
<td><strong>95.0</strong></td>
<td><strong>80.7</strong></td>
</tr>
</tbody>
</table>


# Face Verification on YouTube Faces

<table>
<thead>
<tr>
<th>Methods</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LM3L [1]</td>
<td>81.3 ± 1.2</td>
</tr>
<tr>
<td>DDML (LBP) [2]</td>
<td>81.3 ± 1.6</td>
</tr>
<tr>
<td>DDML (combined) [2]</td>
<td>82.3 ± 1.5</td>
</tr>
<tr>
<td>EigenPEP [3]</td>
<td>84.8 ± 1.4</td>
</tr>
<tr>
<td>DeepFace [4]</td>
<td>91.4 ± 1.1</td>
</tr>
<tr>
<td>DeepID2+</td>
<td>93.2 ± 0.2</td>
</tr>
</tbody>
</table>

- Linear transform
- Pooling
- Nonlinear mapping
<table>
<thead>
<tr>
<th><strong>Unified subspace analysis</strong></th>
<th><strong>Joint deep learning</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Identification signal is in $S_b$; verification signal is in $S_w$</td>
<td>• Learn features by joint identification-verification</td>
</tr>
<tr>
<td>• Maximize distance between classes under constraint that intrapersonal variation is constant</td>
<td>• Minimize intra-personal variation under constraint that the distance between classes is constant</td>
</tr>
<tr>
<td>• Linear feature mapping</td>
<td>• Hierarchical nonlinear feature extraction</td>
</tr>
<tr>
<td></td>
<td>• Generalization power increases with more training identities</td>
</tr>
</tbody>
</table>
What has been learned by DeepID2+?

Properties owned by neurons?

- Moderate sparse
- Selective to identities and attributes
- Robust to data corruption

These properties are naturally owned by DeepID2+ through large-scale training, without explicitly adding regularization terms to the model.
Biological Motivation

- Monkey has a face-processing network that is made of six interconnected face-selective regions
- Neurons in some of these regions were view-specific, while some others were tuned to identity across views
- View could be generalized to other factors, e.g. expressions?

Deeply learned features are moderately space

• For an input image, about half of the neurons are activated
• An neuron has response on about half of the images
Deeply learned features are moderately space

- The binary codes on activation patterns of neurons are very effective on face recognition
- Activation patterns are more important than activation magnitudes in face recognition

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Joint Bayesian (%)</th>
<th>Hamming distance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single model (real values)</td>
<td>98.70</td>
<td>n/a</td>
</tr>
<tr>
<td>Single model (binary code)</td>
<td>97.67</td>
<td>96.46</td>
</tr>
<tr>
<td>Combined model (real values)</td>
<td>99.47</td>
<td>n/a</td>
</tr>
<tr>
<td>Combined model (binary code)</td>
<td>99.12</td>
<td>97.47</td>
</tr>
</tbody>
</table>
Deeply learned features are selective to identities and attributes

• With a single neuron, DeepID2 reaches 97% recognition accuracy for some identity and attribute
Deeply learned features are selective to identities and attributes

- With a single neuron, DeepID2 reaches 97% recognition accuracy for some identity and attribute

Identity classification accuracy on LFW with one single DeepID2+ or LBP feature. GB, CP, TB, DR, and GS are five celebrities with the most images in LFW.

Attribute classification accuracy on LFW with one single DeepID2+ or LBP feature.
Deeply learned features are selective to identities and attributes

- Excitatory and inhibitory neurons

Histograms of neural activations over identities with the most images in LFW
Deeply learned features are selective to identities and attributes

- Excitatory and inhibitory neurons

Histograms of neural activations over gender-related attributes (Male and Female)

Histograms of neural activations over race-related attributes (White, Black, Asian and India)
Histogram of neural activations over age-related attributes (Baby, Child, Youth, Middle Aged, and Senior)

Histogram of neural activations over hair-related attributes (Bald, Black Hair, Gray Hair, Blond Hair, and Brown Hair.)
DeepID2+

High-dim LBP
Deeply learned features are selective to identities and attributes

- Visualize the semantic meaning of each neuron

![Images showing high and low responses for different attributes: Gender, Hair Color, Age, Race, Face Shape, Eye Shape]
Deeply learned features are selective to identities and attributes

- Visualize the semantic meaning of each neuron

Neurons are ranked by their responses in descending order with respect to test images.
DeepID2 features for attribute recognition

- Features at top layers are more effective on recognizing identity-related attributes.
- Features at lower layers are more effective on identity-non-related attributes.
DeepID2 features for attribute recognition

- DeepID2 features can be directly used for attribute recognition
- Use DeepID2 features as initialization (pre-trained result), and then fine tune on attribute recognition
- Average accuracy on 40 attributes on CelebA and LFWA datasets

<table>
<thead>
<tr>
<th>Method</th>
<th>CelebA</th>
<th>LFWA</th>
</tr>
</thead>
<tbody>
<tr>
<td>FaceTracer [1] (HOG+SVM)</td>
<td>81</td>
<td>74</td>
</tr>
<tr>
<td>PANDA-W [2] (Parts are automatically detected)</td>
<td>79</td>
<td>71</td>
</tr>
<tr>
<td>PANDA-L [2] (Parts are given by ground truth)</td>
<td>85</td>
<td>81</td>
</tr>
<tr>
<td>DeepID2</td>
<td><strong>84</strong></td>
<td><strong>82</strong></td>
</tr>
<tr>
<td>Fine-tune (w/o DeepID2)</td>
<td>83</td>
<td>79</td>
</tr>
<tr>
<td>DeepID2 + fine-tune</td>
<td><strong>87</strong></td>
<td><strong>84</strong></td>
</tr>
</tbody>
</table>

Deeply learned features are robust to occlusions

- Global features are more robust to occlusions
Outline

• Deep learning for object recognition on ImageNet

• Deep learning for face recognition
  – Learn identity features from joint verification-identification signals
  – Learn 3D face models from 2D images
Deep Learning Multi-view Representation from 2D Images

• Inspired by brain behaviors [Winrich et al. Science 2010]
• Identity and view represented by different sets of neurons
• Given an image under arbitrary view, its viewpoint can be estimated and its full spectrum of views can be reconstructed

Deep Learning Multi-view Representation from 2D Images

x and y are input and output images of the same identity but in different views;

v is the view label of the output image;

$h^{id}$ are neurons encoding identity features

$h^{v}$ are neurons encoding view features

$h^{r}$ are neurons encoding features to reconstruct the output images
<table>
<thead>
<tr>
<th>Method</th>
<th>Avg.</th>
<th>0°</th>
<th>-15°</th>
<th>+15°</th>
<th>-30°</th>
<th>+30°</th>
<th>-45°</th>
<th>+45°</th>
<th>-60°</th>
<th>+60°</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw Pixels+LDA</td>
<td>36.7</td>
<td>81.3</td>
<td>59.2</td>
<td>58.3</td>
<td>35.5</td>
<td>37.3</td>
<td>21.0</td>
<td>19.7</td>
<td>12.8</td>
<td>7.63</td>
</tr>
<tr>
<td>LBP [1]+LDA</td>
<td>50.2</td>
<td>89.1</td>
<td>77.4</td>
<td>79.1</td>
<td>56.8</td>
<td>55.9</td>
<td>35.2</td>
<td>29.7</td>
<td>16.2</td>
<td>14.6</td>
</tr>
<tr>
<td>Landmark LBP [6]+LDA</td>
<td>63.2</td>
<td>94.9</td>
<td>83.9</td>
<td>82.9</td>
<td>71.4</td>
<td>68.2</td>
<td>52.8</td>
<td>48.3</td>
<td>35.5</td>
<td>32.1</td>
</tr>
<tr>
<td>CNN+LDA</td>
<td>58.1</td>
<td>64.6</td>
<td>66.2</td>
<td>62.8</td>
<td>60.7</td>
<td>63.6</td>
<td>56.4</td>
<td>57.9</td>
<td>46.4</td>
<td>44.2</td>
</tr>
<tr>
<td>FIP [28]+LDA</td>
<td>72.9</td>
<td>94.3</td>
<td>91.4</td>
<td>90.0</td>
<td>78.9</td>
<td>82.5</td>
<td>66.1</td>
<td>62.0</td>
<td>49.3</td>
<td>42.5</td>
</tr>
<tr>
<td>RL [28]+LDA</td>
<td>70.8</td>
<td>94.3</td>
<td>90.5</td>
<td>89.8</td>
<td>77.5</td>
<td>80.0</td>
<td>63.6</td>
<td>59.5</td>
<td>44.6</td>
<td>38.9</td>
</tr>
<tr>
<td>MTL+RL+LDA</td>
<td>74.8</td>
<td>93.8</td>
<td>91.7</td>
<td>89.6</td>
<td>80.1</td>
<td>83.3</td>
<td>70.4</td>
<td>63.8</td>
<td>51.5</td>
<td>50.2</td>
</tr>
</tbody>
</table>

Face recognition accuracies across views and illuminations on the Multi-PIE dataset. The first and the second best performances are in bold.


Deep Learning Multi-view Representation from 2D Images

• Interpolate and predict images under viewpoints unobserved in the training set

The training set only has viewpoints of 0°, 30°, and 60°. (a): the reconstructed images under 15° and 45° when the input is taken under 0°. (b) The input images are under 15° and 45°.
Outline

- Introduction to deep learning
- Deep learning for object recognition
- **Deep learning for object segmentation**
- Deep learning for object detection
- Open questions and future works
Whole-image classification vs pixelwise classification

• Whole-image classification: predict a single label for the whole image

• Pixelwise classification: predict a label at every pixel
  – Segmentation, detection, and tracking

• CNN, forward and backward propagation were originally proposed for whole-image classification

• Such difference was ignored when CNN was applied to pixelwise classification problems, therefore it encountered efficiency problems
Pixelwise Classification

- Image patches centered at each pixel are used as the input of a CNN, and the CNN predicts a class label for each pixel
  - A lot of redundant computation because of overlap between patches

Farabet et al. TPAMI 2013  Pinheiro and Collobert ICML 2014
Classify Segmentation Proposal

- Determines which segmentation proposal can best represent objects of interest

Direct Predict Segmentation Maps

Direct Predict Segmentation Maps

- Classifier is location sensitive has no translation invariance
  - Prediction not only depends on the neighborhood of the pixel, but also its location
- Only suitable for images with regular structures, such as faces and humans
Efficient Forward-Propagation of Convolutional Neural Networks

- Generate the same result as patch-by-patch scanning, with 1500 times speedup for both forward and backward propagation.

The layewise timing and speedup results of the forward and backward propagation by our proposed algorithm on the RCNN model with 3X410X410 images as inputs.

\[
\text{Speedup} = O\left(\frac{s^2 m^2}{(s + m)^2}\right) \quad s^2 \text{ is image size and } m^2 \text{ is patch size}
\]
Fully convolutional neural network

- Replace fully connected layers in CNN with 1 x 1 convolution kernel just like “network in network” (Lin, Chen and Yan, arXiv 2013)
- Take the whole images as inputs and directly output segmentation map
- Has translation invariance like patch-by-patch scanning, but with much lower computational cost
- Once FCNN is learned, it can process input images of any sizes without warping them to a standard size

Fully convolutional neural network

(a) CNN Patch-scanning  (b) CNN Regression  (c) FCNN Segmentation  (d) FCNN Feature Maps

Convolution-pooling layers  Fully connected layers  “Fusion” convolutional layers implemented by 1 x 1 kernel
Summary

- Deep learning significantly outperforms conventional vision systems on large scale image classification.
- Feature representation learned from ImageNet can be well generalized to other tasks and datasets.
- In face recognition, identity preserving features can be effectively learned by joint identification-verification signals.
- 3D face models can be learned from 2D images; identity and pose information is encoded by different sets of neurons.
- In segmentation, larger patches lead to better performance because of the large learning capacity of deep models. It is also possible to directly predict the segmentation map.
- The efficiency of CNN based segmentation can be significantly improved by considering the differences between whole-image classification and pixelwise classification.
References

• Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. CVPR, 2015.
Outline

• Introduction to deep learning
• Deep learning for object recognition
• Deep learning for object segmentation
• Deep learning for object detection
• Open questions and future works
Part IV: Deep Learning for Object Detection

- Pedestrian Detection
- Human part localization
- General object detection

Object detection → Deep learning → Face alignment
Pedestrian detection → Deep learning → Human pose estimation
Part IV: Deep Learning for Object Detection

- Jointly optimize the detection pipeline
- Multi-stage deep learning (cascaded detectors)
- Mixture components
- Integrate segmentation and detection to depress background clutters
- Contextual modeling
- Pre-training
- Model deformation of object parts, which are shared across classes
Joint Deep Learning:

- Jointly optimize the detection pipeline
What if we treat an existing deep model as a black box in pedestrian detection?

ConvNet−U−MS

Results on ETHZ

Results on Caltech Test


Our Joint Deep Learning Model

Modeling Part Detectors

- Design the filters in the second convolutional layer with variable sizes

Part models learned from HOG

Part models

Learned filtered at the second convolutional layer
Our Joint Deep Learning Model

- Convolutional layer 1
  - Image data: 3x3
  - Filtered data map: 19x19

- Average pooling: 4x4

- Convolutional layer 2
  - Extracted feature map: 64x64

- Deformation layer
  - Part detection map: 5x5
  - Part score: 20

Deformation handling

Visibility reasoning and classification

\( y \)
Deformation Layer

Summed map

Part score

Global max pooling

Low value
High value

$B_p$

$M_p$
Part detection map

$D_{1,p}$

$D_{2,p}$

$D_{3,p}$

$D_{4,p}$
Deformation maps
Visibility Reasoning with Deep Belief Net
Results on Caltech Test

Results on ETHZ
Multi-Stage Contextual Deep Learning:

- Train different detectors for different types of samples
- Model contextual information
- Stage-by-stage pretraining strategies

Motivated by Cascaded Classifiers and Contextual Boost

• The classifier of each stage deals with a specific set of samples
• The score map output by one classifier can serve as contextual information for the next classifier

Conventional cascaded classifiers for detection
• Simulate the cascaded classifiers by mining hard samples to train the network stage-by-stage
• Cascaded classifiers are jointly optimized instead of being trained sequentially
• The deep model keeps the score map output by the current classifier and it serves as contextual information to support the decision at the next stage
• To avoid overfitting, a stage-wise pre-training scheme is proposed to regularize optimization
Training Strategies

• Unsupervised pre-train $W_{h,i+1}$ layer-by-layer, setting $W_{s,i+1} = 0, F_{i+1} = 0$
• Fine-tune all the $W_{h,i+1}$ with supervised BP
• Train $F_{i+1}$ and $W_{s,i+1}$ with BP stage-by-stage
• A correctly classified sample at the previous stage does not influence the update of parameters
• Stage-by-stage training can be considered as adding regularization constraints to parameters, i.e. some parameters are constrained to be zeros in the early training stages

Log error function:

$$E = -l \log y - (1 - l) \log (1 - y)$$

Gradients for updating parameters:

$$d\theta_{i,j} = -\frac{\partial E}{\partial \theta_{i,j}} = -\frac{\partial E}{\partial y} \frac{\partial y}{\partial \theta_{i,j}} = -(y - l) \frac{\partial y}{\partial \theta_{i,j}}$$
Experimental Results

Caltech

ETHZ
Comparison of Different Training Strategies

Network-BP: use back propagation to update all the parameters without pre-training

PretrainTransferMatrix-BP: the transfer matrices are unsupervised pretrained, and then all the parameters are fine-tuned

Multi-stage: our multi-stage training strategy
Switchable Deep Network

- Use mixture components to model complex variations of body parts
- Use salience maps to depress background clutters
- Help detection with segmentation information

Switchable Deep Network for Pedestrian Detection

- *Background clutter* and large variations of pedestrian appearance.
- **Proposed Solution.** A Switchable Deep Network (SDN) for learning the foreground map and removing the effect of background clutter.
Switchable Deep Network for Pedestrian Detection

- Switchable Restricted Boltzmann Machine

\[
E(x, y, h, s, m; \Theta) = - \sum_{k=1}^{K} s_k h_k^T (W_k (x \circ m_k) + b_k) - \sum_{k=1}^{K} s_k c_k^T (x \circ m_k) - y^T U \sum_{k=1}^{K} s_k h_k - d^T y,
\]

(a) RBM

(b) Switchable RBM
Switchable Deep Network for Pedestrian Detection

• Switchable Restricted Boltzmann Machine
Switchable Deep Network for Pedestrian Detection

(a) Performance on Caltech Test  
(b) Performance on ETH
Human Part Localization

- Contextual information is important to segmentation as well as detection
Human part localization

- Facial Keypoint Detection
- Human pose estimation

Sun et al. CVPR’ 13
Ouyang et al. CVPR’ 14
Facial Keypoint Detection

Comparison with Liang et al. [6], Valstar et al. [7], Luxand Face SDK [1] and Microsoft Research Face SDK [2] on BioID and LFPW.

Relative improvement = \frac{\text{reduced average error}}{\text{average error of the method in comparison}}.

Benefits of Using Deep Model

• The first network that takes the whole face as input needs deep structures to extract high-level features
• Take the full face as input to make full use of texture context information over the entire face to locate each keypoint
• Since the networks are trained to predict all the keypoints simultaneously, the geometric constraints among keypoints are implicitly encoded
Human pose estimation

Multiple information sources

• Appearance
Multiple information sources

- Appearance
- Appearance mixture type
Multiple information sources

- Appearance
- Appearance mixture type
- Deformation
Multi-source deep model
Experimental results

<table>
<thead>
<tr>
<th>Method</th>
<th>Torso</th>
<th>U.leg</th>
<th>L.leg</th>
<th>U.arm</th>
<th>L.arm</th>
<th>head</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yang &amp; Ramanan CVPR’11</td>
<td>82.9</td>
<td>68.8</td>
<td>60.5</td>
<td>63.4</td>
<td>42.4</td>
<td>82.4</td>
<td>63.6</td>
</tr>
<tr>
<td>Multi-source deep learning</td>
<td>89.3</td>
<td>78.0</td>
<td>72.0</td>
<td>67.8</td>
<td>47.8</td>
<td>89.3</td>
<td>71.0</td>
</tr>
<tr>
<td><strong>PARSE</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Torso</th>
<th>U.leg</th>
<th>L.leg</th>
<th>U.arm</th>
<th>L.arm</th>
<th>head</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yang &amp; Ramanan CVPR’11</td>
<td>81.8</td>
<td>65.0</td>
<td>55.1</td>
<td>46.8</td>
<td>37.7</td>
<td>79.8</td>
<td>57.0</td>
</tr>
<tr>
<td>Multi-source deep learning</td>
<td>89.1</td>
<td>72.9</td>
<td>62.4</td>
<td>56.3</td>
<td>47.6</td>
<td>89.1</td>
<td>65.6</td>
</tr>
<tr>
<td><strong>UIUC People</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Torso</th>
<th>U.leg</th>
<th>L.leg</th>
<th>U.arm</th>
<th>L.arm</th>
<th>head</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yang &amp; Ramanan CVPR’11</td>
<td>82.9</td>
<td>70.3</td>
<td>67.0</td>
<td>56.0</td>
<td>39.8</td>
<td>79.3</td>
<td>62.8</td>
</tr>
<tr>
<td>Multi-source deep learning</td>
<td>85.8</td>
<td>76.5</td>
<td>72.2</td>
<td>63.3</td>
<td>46.6</td>
<td>83.1</td>
<td>68.6</td>
</tr>
<tr>
<td><strong>LSP</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Up to 8.6 percent accuracy improvement with global geometric constraints
Experimental results

Left: mixture-of-parts (Yang & Ramanan CVPR'11)
Right: Multi-source deep learning
General Object Detection

- Pretraining
- Model deformation of object parts, which are shared across classes
- Contextual modeling
Object detection

Pascal VOC
~ 20 object classes
Training: ~ 5,700 images
Testing: ~10,000 images

Image-net ILSVRC
~ 200 object classes
Training: ~ 395,000 images
Testing: ~ 40,000 images
SIFT, HOG, LBP, DPM ...

[Regionlets. Wang et al. ICCV’13] [SegDPM. Fidler et al. CVPR’13]
With CNN features

R-CNN: regions + CNN features

Region:
91.6%/98% recall rate on ImageNet/PASCAL
Selective Search [van de Sande, Uijlings et al. IJCV 2013].

Deep model from Krizhevsky, Sutskever & Hinton. NIPS 2012

SVM: Liblinear
RCNN: deep model training

• Pretrain for the 1000-way ILSVRC image classification task (1.2 million images)
• Fine-tune the CNN for detection
  ➢ Transfer the representation learned from ILSVRC Classification to PASCAL (or ImageNet) detection

Network from Krizhevsky, Sutskever & Hinton. NIPS 2012
Also called “AlexNet”
Experimental results on ILSVRC 2013

ILSVRC2013 detection test set mAP

- *R-CNN BB: 31.4%
- *OverFeat (2): 24.3%
- UvA-Euvision: 22.6%
- *NEC–MU: 20.9%
- *OverFeat (1): 19.4%
- Toronto A: 11.5%
- SYSU_Vision: 10.5%
- GPU_UCLA: 9.8%
- Delta: 6.1%
- UIUC–IFP: 1.0%

Mean average precision (mAP) in %

Red: post competition result
Blue: competition result
Experimental results on ILSVRC 2014

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Model average</td>
<td>0.439</td>
<td>0.439</td>
<td>0.405</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Single model</td>
<td>0.380</td>
<td>0.427</td>
<td>0.402</td>
<td>0.354</td>
<td>0.345</td>
<td>0.314</td>
</tr>
</tbody>
</table>
DeepID-Net: deformable deep convolutional neural networks for generic object detection

RCNN

Image → Selective search → Proposed bounding boxes → AlexNet+ SVM → Detection results → Bounding box regression → Refined bounding boxes

Detection results

Refined bounding boxes

person

horse
RCNN

Mean ap 31.4 \rightarrow 40.67 (new result on)

DeepID-Net
RCNN

Image → Selective search → Proposed bounding boxes → AlexNet+ SVM → Detection results → Bounding box regression → Refined bounding boxes

DeepID-Net

Image → Selective search → Proposed bounding boxes → Box rejection → Remaining bounding boxes → DeepID-Net

Pretrain, deep-pooling layer, sub-box, hinge-loss → Context modeling

Model averaging
DeepID-Net
RCNN

DeepID-Net
Deep model training – pretrain

- RCNN (Cls+Det)
  - Pretrain on image-level annotation with 1000 classes
  - Finetune on object-level annotation with 200 classes
  - Gap: classification vs. detection, 1000 vs. 200

Image classification

Object detection
Result and discussion

• Investigation
  • Better pretraining on 1000 classes
  • Object-level annotation is more suitable for pretraining

• Conclusions
  • The supervisory tasks should match at the pre-training and fine-turning stages
  • Although an application only involves detecting a small number of classes, it is better to pretraining with many classes outside the application

<table>
<thead>
<tr>
<th></th>
<th>Image annotation</th>
<th>Object annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>200 classes (Det)</td>
<td>20.7</td>
<td>28.0</td>
</tr>
<tr>
<td>1000 classes (Cls-Loc)</td>
<td>31.8</td>
<td>36</td>
</tr>
</tbody>
</table>
Deformation

– Learning deformation [a] is effective in computer vision society.
– Missing in deep model.
– We propose a new deformation constrained pooling layer.

Deformation Layer [b]

\[ B_p = M_p + \sum_{n=1}^{N} c_{n,p} D_{n,p} \quad s_p = \max \limits_{(x,y)} b_p^{(x,y)} \]

Modeling Part Detectors

- Different parts have different sizes
- Design the filters with variable sizes

Part models learned from HOG

Part models

Learned filtered at the second convolutional layer
Deformation layer for repeated patterns

<table>
<thead>
<tr>
<th>Pedestrian detection</th>
<th>General object detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assume no repeated pattern</td>
<td>Repeated patterns</td>
</tr>
</tbody>
</table>

![Images of objects with bounding boxes showing pedestrian detection and general object detection differences.]
Deformation layer for repeated patterns

<table>
<thead>
<tr>
<th>Pedestrian detection</th>
<th>General object detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assume no repeated pattern</td>
<td>Repeated patterns</td>
</tr>
<tr>
<td>Only consider one object class</td>
<td>Patterns shared across different object classes</td>
</tr>
</tbody>
</table>

![Pedestrian detection examples](image1)

![General object detection examples](image2)
Deformation constrained pooling layer

Can capture multiple patterns simultaneously

\[ b(x, y) = \max_{i, j \in \{-R, \ldots, R\}} \left\{ m(k_x \cdot x + i, k_y \cdot y + j) \right\} - \sum_{n=1}^{N} c_n d_{n,i,j} \]
Our deep model with deformation layer

<table>
<thead>
<tr>
<th>Training scheme</th>
<th>Cls+Det</th>
<th>Loc+Det</th>
<th>Loc+Det</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net structure</td>
<td>AlexNet</td>
<td>Clarifai</td>
<td>Clarifai+Def layer</td>
</tr>
<tr>
<td>Mean AP on val2</td>
<td>0.299</td>
<td>0.360</td>
<td>0.385</td>
</tr>
</tbody>
</table>
Context modeling

• Use the 1000 class Image classification score.
• ~1% mAP improvement.
Context modeling

• Use the 1000-class Image classification score.
  – ~1% mAP improvement.
  – Volleyball: improve ap by 8.4% on val2.
RCNN

DeepID-Net
Model averaging

• Not only change parameters
  – Net structure: AlexNet(A), Clarifai (C), Deep-ID Net (D), DeepID Net2 (D2)
  – Pretrain: Classification (C), Localization (L)
  – Region rejection or not
  – Loss of net, softmax (S), Hinge loss (H)
  – Choose different sets of models for different object class

<table>
<thead>
<tr>
<th>Model</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net structure</td>
<td>A</td>
<td>A</td>
<td>C</td>
<td>C</td>
<td>D</td>
<td>D</td>
<td>D2</td>
<td>D</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>Pretrain</td>
<td>C</td>
<td>C+L</td>
<td>C</td>
<td>C+L</td>
<td>C+L</td>
<td>C+L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>Reject region?</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Loss of net</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>Mean ap</td>
<td>0.31</td>
<td>0.312</td>
<td>0.321</td>
<td>0.336</td>
<td>0.353</td>
<td>0.36</td>
<td>0.37</td>
<td>0.37</td>
<td>0.371</td>
<td>0.374</td>
</tr>
</tbody>
</table>
Component analysis

<table>
<thead>
<tr>
<th>Detection Pipeline</th>
<th>RCNN</th>
<th>Box rejection</th>
<th>Clarifai</th>
<th>Loc+ Det</th>
<th>+Def layer</th>
<th>+cont ext</th>
<th>+bbox regr.</th>
<th>Model avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>mAP on val2</td>
<td>29.9</td>
<td>30.9</td>
<td>31.8</td>
<td>36.0</td>
<td>38.5</td>
<td>39.2</td>
<td>40.1</td>
<td>42.4</td>
</tr>
<tr>
<td>mAP on test</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>38.0</td>
<td>38.6</td>
</tr>
</tbody>
</table>

DeepID-Net

Image → Selective search → Proposed bounding boxes → Box rejection → Remaining bounding boxes → DeepID-Net

Pretrain, def-pooling layer, sub-box, hinge-loss → Context modeling → Model averaging
Summary

• Bounding rejection. Save feature extraction by about 10 times, slightly improve mAP (~1%).
• Pre-training with object-level annotation, more classes. 4.2% mAP
• Def-pooling layer. 2.5% mAP improvement
• Contextual modeling. 1% mAP improvement
• Model averaging. 2.3% mAP improvement. Different model designs and training schemes lead to high diversity
Reference

• W. Ouyang and X. Wang, "Joint Deep Learning for Pedestrian Detection," ICCV 2013
Reference

Outline

• Introduction to deep learning
• Deep learning for object recognition
• Deep learning for object segmentation
• Deep learning for object detection
• Open questions and future works
“Concerns” on deep learning

• C1: Weak on theoretical support (convergence, bound, local minimum, why it works)
  – It’s true. That’s why deep learning papers were not accepted by the computer vision/image processing community for a long time. Any theoretical studies in the future are important.
Most computer vision/multimedia papers

- Motivations
- New objective function
- **New optimization algorithm**
- Theoretical analysis
- Experimental results

Deep learning papers for computer vision/multimedia

- Motivations
- New network structure and new objective function
- Back propagation (standard)
- Super experimental results

That’s probably one of the reasons that computer vision and image processing people think deep learning papers are lack of novelty and theoretical contribution 😊
“Concerns” on deep learning

- C2: It is hard for computer vision/image processing people to have innovative contributions to deep learning. Our job becomes preparing the data + using deep learning as a black box. That’s the end of our research life.
  - That’s not true. Computer vision and image processing researchers have developed many systems with deep architectures. But we just didn’t know how to jointly learn all the components. Our research experience and insights can help to design new deep models and pre-training strategies.
  - Many machine learning models and algorithms were motivated by computer vision and image processing applications. However, computer vision and multimedia did not have close interaction with neural networks in the past 15 years. We expect fast development of deep learning driven by applications.
“Concerns” on deep learning

• C3: Since the goal of neural networks is to solve the general learning problem, why do we need domain knowledge?
  – The most successful deep model on image and video related applications is convolutional neural network, which has used domain knowledge (filtering, pooling)
  – Domain knowledge is important especially when the training data is not large enough
“Concerns” on deep learning

• C4: Good results achieved by deep learning come from manually tuning network structures and learning rates, and trying different initializations
  – That’s not true. One round evaluation may take several weeks. There is no time to test all the settings.
  – Designing and training deep models does require a lot of empirical experience and insights. There are also a lot of tricks and guidance provided by deep learning researchers. Most of them make sense intuitively but without strict proof.
“Concerns” on deep learning

• C5: Deep learning is more suitable for industry rather than research groups in universities
  – Industry has big data and computation resources
  – Research groups from universities can contribute on model design, training algorithms and new applications
“Concerns” on deep learning

• C6: Deep learning has different behaviors when the scale of training data is different
  – Pre-training is useful when the training data small, but does not make big difference when the training data is large enough
  – So far, the performance of deep learning keep increasing with the size of training data. We don’t see its limit yet.
  – Shall we spend more effort on data annotation or model design?
Future works

• Explore deep learning in new applications
  – Worthy to try if the applications require features or learning, and have enough training data
  – We once had many doubts on deep. (Does it work for vision? Does it work for segmentation? Does it work for low-level vision?) But deep learning has given us a lot of surprises.
  – Applications will inspire many new deep models

• Incorporate domain knowledge into deep learning

• Integrate existing machine learning models with deep learning
Future works

• Deep learning to extract dynamic features for video analysis
• Deep models for structured data
• Theoretical studies on deep learning
• Quantitative analysis on how to design network structures and how to choose nonlinear operations of different layers in order to achieve feature invariance
• New optimization and training algorithms
• Parallel computing systems and algorithm to train very large and deep networks with larger training data
# Multimodal Laboratory

## Projects / Deep Learning

**Description**

A demo code that allows you to input a pedestrian image and then compute the label map.

**Reference:**

A demo code that shows you how the frontal-view face image of a query face image is reconstructed.

**Reference:**

Matlab training and testing source code for pedestrian detection using the proposed approach. Models trained on INRIA and Caltech are provided.

**Reference:**

Executable files for the face detector and facial point detector.

**Reference:**

---

http://mmlab.ie.cuhk.edu.hk/project_deep_learning.html
Thank you!

http://mmlab.ie.cuhk.edu.hk/

http://www.ee.cuhk.edu.hk/~xgwang/