Deep Learning in Medical Image Analysis

Dinggang Shen
Deep Learning
For imaging-based diagnosis, manual labeling is expensive, and hence ground truth data are limited.
<table>
<thead>
<tr>
<th>Methods</th>
<th>Limitations</th>
</tr>
</thead>
<tbody>
<tr>
<td>• PCA</td>
<td>➢ Linear&lt;br&gt;➢ Not optimal for non-Gaussian data</td>
</tr>
<tr>
<td>• Gaussian Mixture Models</td>
<td>➢ Require knowledge for the number of clusters&lt;br&gt;➢ Challenging when applied to high-dimensional data</td>
</tr>
<tr>
<td>• k-Means</td>
<td></td>
</tr>
<tr>
<td>• ICA</td>
<td>➢ Linear model</td>
</tr>
<tr>
<td>• Sparse Coding</td>
<td>➢ Shallow model (e.g., single-layer representation)</td>
</tr>
<tr>
<td>• Non-Linear Embedding</td>
<td></td>
</tr>
</tbody>
</table>
All these methods involve just one step of mapping.

Mapping is shallow, **not deep**!

Thus, not able to represent complex mapping!
Deep Learning – Why hot?

- Deep mapping and representation
- Won the 1st place in many competitions.
- Industrial applications (Google, IBM, Microsoft, Baidu, Facebook, Samsung, Yahoo, Intel, Apple, Nuance, BBN, ...)

![Deep learning diagram]
Deep Learning – Why hot?

- Deeper representations ➔ abstractions ➔ disentangling
- Manifolds are expanded and flattened

The following 7 Slides edited from Dr. Yoshua Bengio’s tutorial
Deep Learning – Why hot?

- Deeper representations ➔ abstractions ➔ disentangling
- Manifolds are expanded and flattened

[Diagram showing linear interpolation in pixel space and representation space]
Deep Learning – Why hot?

- Each level transforms the data into a representation which can be easily modeled → Unfolding it more will map the original data to a factorized (uniform-like) distribution.
Deep Learning – Why hot?

Performance increase with layers

Raw data
1 layer
2 layers
3 layers
4 layers

NIPS’2011
Transfer Learning Challenge
Paper: ICML’2012

ICML’2011 workshop on Unsup. & Transfer Learning
Deep Learning – Why hot?

Successive model layers learn deeper intermediate representations

(Lee, Largman, Pham & Ng, NIPS 2009)
(Lee, Grosse, Ranganath & Ng, ICML 2009)

Prior: underlying factors & concepts compactly expressed w/ multiple levels of abstraction
Neural Network – Why not working

- Issues with previous neural network (NN), such as Back-Propagation (BP)
  - Gradient-based method → Propagate errors from the last layer to the previous layers
  - Last layer represents high nonlinear function, i.e., a jump function in binary classification → unstable and large gradient in small range, but zero in most places → difficult to propagate errors to previous layers
Neural Network – Why not working

- Effect of Initial Conditions in Deep Nets
  - Random initialization
  - Unsupervised pre-training

- No two training trajectories end up in the same place → Huge number of effective local minima
- Pre-training: Transfer knowledge from previous learning (representation and explanatory factors) → cases with few examples → shared underlying explanatory factors, between $P(X)$ and $P(Y|X)$

- Random initialization
- Unsupervised pre-training
Deep Learning – Why working now

3 Main Reasons:

1) New layer-wise training algorithm (*Since 2006*)
   - Each time, train on simple task

2) Large data, compared to 20 years ago

3) Powerful computers
   - Previous algorithms may be theoretically working, but practically not converged to good local minima with the previous less-powerful computers.
Deep Learning

Restricted Boltzmann Machine (RBM)

Visible layer

Hidden layers

Directed belief nets

\[ P(v, h^1, h^2, ..., h^l) = P(v | h^1)P(h^1 | h^2)...P(h^{l-2} | h^{l-1})P(h^{l-1}, h^l) \]
First step:
- Construct an RBM with an input layer $v$ and a hidden layer $h$
- Train the RBM
Deep Learning – Greedy Training

• Second step:
  – Stack another hidden layer on top of the RBM to form a new RBM
  – Fix $W^1$, sample $h^1$ from $Q(h^1 | v)$ as input. Train $W^2$ as RBM.
Deep Learning – Greedy Training

- Third step:
  - Continue to stack layers on top of the network, train it as previous step, with sample sampled from $Q(h^2 | h^1)$

- And so on...
Stacked Auto-Encoder

Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton* and R. R. Salakhutdinov

High-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors. Gradient descent can be used for fine-tuning the weights in such "autoencoder" networks, but this works well only if the initial weights are close to a good solution. We describe an effective way of initializing the weights that allow deep autoencoder networks to learn low-dimensional codes that work much better than principal components analysis as a tool to reduce the dimensionality of data.
Stacked Auto-Encoder
Application 1:
Hippocampus Segmentation Using 7T MR Images

Importance

The volume of the hippocampus is an important trait for early diagnosis of neurological diseases (e.g., Alzheimer’s disease)

Challenges

- The hippocampus is small ($\approx 35 \times 15 \times 7 \text{mm}^3$)
- The hippocampus is surrounded by complex structures
- Low imaging resolution ($\approx 1 \times 1 \times 1 \text{mm}^3$) of 1.5T or 3T MRI scanners
Challenges in 7T images

- The characteristics of 7T MR images
  - Much richer structural information
  - Severe intensity inhomogeneity problem
  - Less partial volume effect

Comparison between 1.5T and 7T MR images

1.5T (1 x 1 x 1 mm³) 7.0T (0.35 x 0.35 x 0.35 mm³)
Hand-Crafted Features

- Limited discriminative power

Extracting patches from a 7T MR image

Responses of Haar filters for the image patches in a-c
Proposed method:
- Multi-atlas segmentation
- Classification using high-level shape information via auto-context models (ACM)
- Hierarchical feature representation via unsupervised deep learning

Basis filters generated by unsupervised deep learning
Context feature
Hierarchical Feature Extraction

Stacked two-layer convolutional ISA (Independent Subspace Analysis)

Image patches $X$

1st ISA layer

Basis filters $W$ in 1st layer

Activations $P$ in 1st layer

PCA

Learned basis filters by the 1st ISA

2nd ISA layer

Activations $P'$ in 2nd layer

Basis filters $W'$ in 2nd layer

Dimension-reduced activations from 1st layer

Department of Radiology and BRIC, UNC-Chapel Hill
Multi-Atlas Segmentation

Training Stage
- Aligned training images in each atlas space 1…N
- 2-layer ISA
- Learned features
- Classifier sequences 1…N
- Atlas spaces 1…N

Testing Stage
- 2-layer ISA
- Classification maps 1…N
- Adaptively weighted fusion
- Probability map
- Level set
- Segmentation result
- Subject image space
Qualitative Evaluation

Ground Truth

Haar + Texture Features

Hierarchical Features
Quantitative Evaluation

Overlap metrics

Precision

\[ P = \frac{V(A \cap B)}{V(B)} \]

Recall

\[ R = \frac{V(A \cap B)}{V(A)} \]

Relative overlap

\[ RO = \frac{V(A \cap B)}{V(A \cup B)} \]

Similarity index

\[ SI = \frac{V(A \cap B)}{(V(A) + V(B))/2} \]

Comparison Results Using 20 Leave-One-Out Cases

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>R</th>
<th>RO</th>
<th>SI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hand-Crafted Haar + Texture Features</td>
<td>0.843</td>
<td>0.847</td>
<td>0.772</td>
<td>0.865</td>
</tr>
<tr>
<td>Hierarchical Patch Representations</td>
<td>0.883</td>
<td>0.881</td>
<td>0.819</td>
<td>0.894</td>
</tr>
</tbody>
</table>
Application 2:
Registration of Brain MR Images

Image Registration

Determine accurate correspondences between images
Feature Extraction

- Hand-crafted features are **NOT reusable**
- New learning-based framework for determining **intrinsic** feature representations
Learned Features

(a) Template
(b) Subject
(c) Deformed subject

(d) By local patches
(e) By SIFT
(f) By unsupervised learning
Deep Learning

Input image patches

Morphological signatures for image registration

Representations

Encoder

Decoder

"x", "y(1)", "y(2)", ..., "z"
**Difficulty #1:** How to determine the number of hidden nodes and the number of layers?

**Solution:** Use affinity propagation to roughly estimate the number of hidden units.
**Deep Learning**

**Difficulty #2:** How to deal with high dimensional training data?

**Solution:** Use the convolutional RBM in each layer.
**Deep Learning**

**Difficulty #3:** How to deal with large number of training samples?

**Solution:** Select key points across training images to focus on distinctive image regions.

(a) Importance map  
(b) Key points
Symmetric Image Registration

$F = \Phi_1 \circ \Phi_2$  

$F^{-1} = \Phi_2 \circ \Phi_1$  

$T_0 \quad T_k \quad T_K \quad S_K \quad S_k \quad S_0$

$\Phi_1 \quad \Phi_1^k \quad \Phi_2^k \quad \Phi_2$
ADNI Dataset

Training Images: 10 images
Testing Images: 24 images
Patch Size: 21 x 21 x 21
Number of Layers: 7
Number of Features: 150

Dice ratio

<table>
<thead>
<tr>
<th>Methods</th>
<th>Ventricle</th>
<th>Gray Matter</th>
<th>White Matter</th>
<th>Hippocampus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demons</td>
<td>90.2</td>
<td>76.0</td>
<td>85.7</td>
<td>72.2</td>
</tr>
<tr>
<td>M+PCA</td>
<td>90.5</td>
<td>76.6</td>
<td>85.5</td>
<td>72.3</td>
</tr>
<tr>
<td>M+DP</td>
<td>90.9</td>
<td>76.5</td>
<td>85.8</td>
<td>72.5</td>
</tr>
<tr>
<td>HAMMER</td>
<td>91.5</td>
<td>75.5</td>
<td>85.4</td>
<td>75.5</td>
</tr>
<tr>
<td>H+PCA</td>
<td>91.7</td>
<td>76.9</td>
<td>86.5</td>
<td>75.6</td>
</tr>
<tr>
<td>H+DP</td>
<td>95.0</td>
<td>78.6</td>
<td>88.1</td>
<td>76.8</td>
</tr>
<tr>
<td></td>
<td>Demons</td>
<td>M+PCA</td>
<td>M+DP</td>
<td>HAMMER</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------</td>
<td>-------</td>
<td>------</td>
<td>--------</td>
</tr>
<tr>
<td>R sup. frontal gyrus</td>
<td>79.7</td>
<td>79.8</td>
<td>80.0</td>
<td>77.5</td>
</tr>
<tr>
<td>R middle frontal gyrus</td>
<td>77.4</td>
<td>78.1</td>
<td>78.2</td>
<td>79.5</td>
</tr>
<tr>
<td>R middle frontal gyrus</td>
<td>77.0</td>
<td>76.5</td>
<td>77.1</td>
<td>78.2</td>
</tr>
<tr>
<td>R inf. frontal gyrus</td>
<td>72.2</td>
<td>72.8</td>
<td>72.6</td>
<td>72.8</td>
</tr>
<tr>
<td>R precentral gyrus</td>
<td>67.9</td>
<td>68.5</td>
<td>68.4</td>
<td>68.6</td>
</tr>
<tr>
<td>R precentral gyrus</td>
<td>68.6</td>
<td>68.5</td>
<td>69.0</td>
<td>65.0</td>
</tr>
<tr>
<td>L middle orbitofrontal gyrus</td>
<td>66.9</td>
<td>66.1</td>
<td>67.1</td>
<td>69.8</td>
</tr>
<tr>
<td>R middle orbitofrontal gyrus</td>
<td>66.8</td>
<td>67.5</td>
<td>67.7</td>
<td>69.4</td>
</tr>
<tr>
<td>L lateral orbitofrontal gyrus</td>
<td>59.4</td>
<td>55.6</td>
<td>55.7</td>
<td>64.7</td>
</tr>
<tr>
<td>R gyrus rectus</td>
<td>68.7</td>
<td>66.1</td>
<td>66.9</td>
<td>67.9</td>
</tr>
<tr>
<td>R gyrus rectus</td>
<td>68.1</td>
<td>67.7</td>
<td>68.1</td>
<td>65.5</td>
</tr>
<tr>
<td>L postcentral gyrus</td>
<td>60.5</td>
<td>60.7</td>
<td>61.4</td>
<td>60.5</td>
</tr>
<tr>
<td>R postcentral gyrus</td>
<td>63.9</td>
<td>63.4</td>
<td>62.5</td>
<td>63.5</td>
</tr>
<tr>
<td>L sup. parietal gyrus</td>
<td>70.7</td>
<td>70.9</td>
<td>70.6</td>
<td>72.6</td>
</tr>
<tr>
<td>R sup. parietal gyrus</td>
<td>70.9</td>
<td>70.6</td>
<td>71.2</td>
<td>71.8</td>
</tr>
<tr>
<td>L supramarginal gyrus</td>
<td>63.8</td>
<td>63.4</td>
<td>64.1</td>
<td>63.1</td>
</tr>
<tr>
<td>R supramarginal gyrus</td>
<td>63.3</td>
<td>63.7</td>
<td>63.8</td>
<td>65.1</td>
</tr>
<tr>
<td>L angular gyrus</td>
<td>63.2</td>
<td>62.8</td>
<td>63.5</td>
<td>66.9</td>
</tr>
<tr>
<td>R angular gyrus</td>
<td>65.0</td>
<td>65.1</td>
<td>65.7</td>
<td>65.4</td>
</tr>
<tr>
<td>L precuneus</td>
<td>66.9</td>
<td>67.7</td>
<td>68.4</td>
<td>70.6</td>
</tr>
<tr>
<td>R precuneus</td>
<td>67.3</td>
<td>67.2</td>
<td>67.8</td>
<td>70.8</td>
</tr>
<tr>
<td>L sup. occipital gyrus</td>
<td>58.1</td>
<td>58.0</td>
<td>58.2</td>
<td>61.2</td>
</tr>
<tr>
<td>R sup. occipital gyrus</td>
<td>55.4</td>
<td>55.9</td>
<td>56.2</td>
<td>64.5</td>
</tr>
<tr>
<td>L middle occipital gyrus</td>
<td>68.7</td>
<td>68.5</td>
<td>68.4</td>
<td>72.6</td>
</tr>
<tr>
<td>R middle occipital gyrus</td>
<td>67.9</td>
<td>67.8</td>
<td>67.8</td>
<td>71.1</td>
</tr>
<tr>
<td>L inf. occipital gyrus</td>
<td>67.2</td>
<td>67.8</td>
<td>67.9</td>
<td>65.8</td>
</tr>
<tr>
<td>R inf. occipital gyrus</td>
<td>66.1</td>
<td>66.5</td>
<td>67.1</td>
<td>62.0</td>
</tr>
<tr>
<td>L cuneus</td>
<td>63.4</td>
<td>63.4</td>
<td>63.9</td>
<td>64.1</td>
</tr>
<tr>
<td>R cuneus</td>
<td>52.2</td>
<td>52.4</td>
<td>52.5</td>
<td>60.0</td>
</tr>
<tr>
<td>L sup. temporal gyrus</td>
<td>72.5</td>
<td>72.5</td>
<td>72.7</td>
<td>69.9</td>
</tr>
<tr>
<td>R sup. temporal gyrus</td>
<td>72.6</td>
<td>73.1</td>
<td>73.4</td>
<td>74.1</td>
</tr>
<tr>
<td>L middle temporal gyrus</td>
<td>66.4</td>
<td>66.8</td>
<td>66.8</td>
<td>67.1</td>
</tr>
<tr>
<td>R middle temporal gyrus</td>
<td>67.9</td>
<td>67.5</td>
<td>67.9</td>
<td>68.3</td>
</tr>
<tr>
<td>L inf. temporal gyrus</td>
<td>66.6</td>
<td>66.2</td>
<td>66.9</td>
<td>66.5</td>
</tr>
<tr>
<td>R inf. temporal gyrus</td>
<td>66.4</td>
<td>66.4</td>
<td>66.5</td>
<td>66.9</td>
</tr>
<tr>
<td>L parahippocampal gyrus</td>
<td>68.1</td>
<td>68.2</td>
<td>68.5</td>
<td>68.0</td>
</tr>
<tr>
<td>R parahippocampal gyrus</td>
<td>66.9</td>
<td>66.7</td>
<td>67.2</td>
<td>67.5</td>
</tr>
<tr>
<td>L lingual gyrus</td>
<td>69.7</td>
<td>69.8</td>
<td>68.9</td>
<td>69.4</td>
</tr>
<tr>
<td>R lingual gyrus</td>
<td>70.6</td>
<td>70.5</td>
<td>70.6</td>
<td>73.6</td>
</tr>
<tr>
<td>L fusiform gyrus</td>
<td>68.9</td>
<td>68.8</td>
<td>69.1</td>
<td>66.5</td>
</tr>
<tr>
<td>R fusiform gyrus</td>
<td>68.3</td>
<td>68.3</td>
<td>68.5</td>
<td>67.5</td>
</tr>
<tr>
<td>L insular cortex</td>
<td>76.4</td>
<td>76.1</td>
<td>76.5</td>
<td>77.5</td>
</tr>
<tr>
<td>R insular cortex</td>
<td>74.2</td>
<td>74.6</td>
<td>74.7</td>
<td>75.1</td>
</tr>
<tr>
<td>L lingulate gyrus</td>
<td>68.1</td>
<td>68.2</td>
<td>68.8</td>
<td>69.5</td>
</tr>
<tr>
<td>R lingulate gyrus</td>
<td>67.5</td>
<td>67.4</td>
<td>67.2</td>
<td>69.2</td>
</tr>
<tr>
<td>L cuneus</td>
<td>73.4</td>
<td>73.4</td>
<td>73.8</td>
<td>74.5</td>
</tr>
<tr>
<td>R cuneus</td>
<td>73.1</td>
<td>73.0</td>
<td>73.5</td>
<td>76.2</td>
</tr>
<tr>
<td>L putamen</td>
<td>76.3</td>
<td>76.5</td>
<td>76.7</td>
<td>77.0</td>
</tr>
<tr>
<td>R putamen</td>
<td>76.5</td>
<td>76.3</td>
<td>76.4</td>
<td>76.5</td>
</tr>
<tr>
<td>L hippocampus</td>
<td>72.7</td>
<td>72.6</td>
<td>72.8</td>
<td>74.7</td>
</tr>
<tr>
<td>R hippocampus</td>
<td>72.8</td>
<td>72.6</td>
<td>73.1</td>
<td>75.9</td>
</tr>
<tr>
<td>cerebellum</td>
<td>84.9</td>
<td>85.1</td>
<td>85.9</td>
<td>86.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demons</td>
<td>68.9</td>
</tr>
<tr>
<td>M+PCA</td>
<td>68.9</td>
</tr>
<tr>
<td>M+DP</td>
<td>69.2</td>
</tr>
<tr>
<td>HAMMER</td>
<td>70.2</td>
</tr>
<tr>
<td>H+PCA</td>
<td>70.6</td>
</tr>
<tr>
<td>H+DP</td>
<td>72.7</td>
</tr>
</tbody>
</table>
Results – IXI Dataset

STG: Superior temporal gyrus
M&ITG: Middle and inferior temporal gyrus
MFG: Middle frontal gyrus
AOG: Anterior orbital gyrus
IFG: Inferior frontal gyrus
SFG: Superior frontal gyrus
POG: Posterior gyrus
LG: Lingual gyrus
PCG: Precentral gyrus
Results

Development of registration algorithms for 7T MR Image

1.5T (1 x 1 x 1 mm³) 7.0T (0.35 x 0.35 x 0.35mm³)

Overlap ratio for hippocampus

78.4%  Our method
68.5%  Demons

Features learned in the first layer
Application 3: Disease Diagnosis


Alzheimer’s Disease

- The most common form of dementia
  - 6th leading cause-of-death in US
  - Mild Cognitive Impairment (MCI): prodromal stage of AD

- Treatments
  - Small symptomatic benefits for mild-to-moderate AD
  - Cannot delay or halt the progression of AD

Forecast of prevalence AD in the US

- 2000 (est): 4.5 Million
- 2030 (est): 8.6 Million
- 2050 (est): 16.0 Million

- **Neuroimaging modalities for diagnosis**
  - Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET), fMRI, etc.

- **Simple low-level features** [Previous work]
  - MRI: gray matter tissue volumes
  - PET: mean signal intensities
  - Cerebrospinal Fluid (CSF): biomarker measures

Vulnerable to noises and/or artifacts
Motivation

- Hidden or latent high-level information
  - Deep architecture can be efficiently used to discover latent or hidden representation in self-taught learning
  - Overcome the vulnerability to noise/artifacts in the data by encoding in a hierarchical feature space

- Unsupervised greedy training
  - Allows us to benefit from the target-unrelated samples to discover general latent feature representations
  - Leverages for enhancement of the accuracy
- Deep or hierarchical architecture to find the highly non-linear and complex patterns in data [Bengio, 2009]
- Stacked Auto-Encoder (SAE): discover latent information such as relations among features
Proposed Framework

Latent feature representation

Feature selection

Multi-modality fusion

Feature extraction
- MRI
- PET
- CSF

Template
- Pre-training
- Fine-tuning

Feature representation with stacked auto-encoder

Label prediction
- Clinical scores regression

Multi-task learning

Multi-kernel SVM learning
- MRI kernel
- PET kernel
- CSF kernel

AD/MCI diagnosis

Label
- MMSE
- ADAS-Cog
ADNI dataset: baseline MRI, PET, and CSF data
- 51 AD, 52 HC, 43 MCI-C, 56 MCI-NC

**Experimental Results**

- **LLF**: Low-Level Features
- **SAEF**: SAE Features

![Graphs showing experimental results for different comparisons: AD vs. HC, MCI vs. HC, AD vs. MCI, MCI vs. MCI-NC.](Image)
Fusing complementary information from multiple modalities helps enhance diagnostic accuracy

Previous approaches

- Simple concatenation into a long vector [Kohannim et al., 2010]
- Kernel methods [Hinrichs et al., 2011; Zhang et al., 2011; Suk and Shen, 2013]

Independent steps of feature extraction and modality fusion

Motivation

- High-level feature representation via deep learning
  - Successfully applied in medical imaging analysis
    [Shin et al., 2013; Liao et al., 2013; Ciresan et al., 2013; Suk and Shen, 2013]
  - Deep Boltzmann Machine (DBM) [Salakhutdinov and Hinton, 2012]
    - An undirected graphical model
    - Structured by stacking multiple restricted Boltzmann machines in a hierarchical manner

- Inherent relations between modalities of MRI and PET
  - Shared feature representation
  - Multi-Modal DBM (MM-DBM) [Srivastava and Salakhutdinov, 2012]
Proposed Framework

Multi-modal input images
$2@[I \times I \times I]$  

Patch extraction  
$2K@|w \times w \times w|$  

Patch-level feature learning  
$K@F_S$  

Image-level classifier learning

$I$: image size, $w$: patch size, $K$: # of selected patches, $m$: modality index,  
$F_G$: # of hidden units in Gaussian restricted Boltzmann machine,  
$F_S$: # of hidden units in the top-layer of multi-modal Deep Boltzmann Machine (DBM)
Learned Weights

Hidden Layer 1

Hidden Layer 2

MRI

PET

Department of Radiology and BRIC, UNC-Chapel Hill
## Experimental Results

### Comparison with state-of-the-art methods

<table>
<thead>
<tr>
<th>Methods</th>
<th>Dataset (AD/MCI/NC)</th>
<th>AD vs. NC (%)</th>
<th>MCI vs. NC (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kohannim et al., 2010</td>
<td>MRI+PET+CSF (40/83/43)</td>
<td>90.7</td>
<td>75.8</td>
</tr>
<tr>
<td>Walhovd et al., 2010</td>
<td>MRI+CSF (38/73/42)</td>
<td>88.8</td>
<td>79.1</td>
</tr>
<tr>
<td>Hinrichs et al., 2011</td>
<td>MRI+PET (48/119/66)</td>
<td>92.4</td>
<td>n/a</td>
</tr>
<tr>
<td>Westman et al., 2012</td>
<td>MRI+CSF (96/162/111)</td>
<td>91.8</td>
<td>77.6</td>
</tr>
<tr>
<td>Zhang and Shen, 2012</td>
<td>MRI+PET+CSF (51/99/52)</td>
<td>93.3</td>
<td>83.2</td>
</tr>
<tr>
<td><strong>Proposed method</strong></td>
<td>MRI+PET (93/204/101)</td>
<td><strong>93.5</strong></td>
<td><strong>85.19</strong></td>
</tr>
</tbody>
</table>
Application 4:
Prostate Labeling

Due to good soft tissue contrast of MR prostate images, there is an increasing interest in
- MR-guided transperineal prostate core biopsy
- MR-guided radiotherapy planning
- Quantitative analysis using MR images (e.g., volume measurement)

Accurate segmentation of prostate in MR images is a perquisite for all these steps.
Challenges

Large inter-subject anatomical appearance variability

Inhomogeneity

Large inter-subject shape variability
Multi-Atlas Label Propagation

How to find a good feature representation \( \tilde{f}_{\tilde{x}} \) for sparse label propagation?

- Hand-crafted features (e.g., Haar, HOG)
- Data-driven features (e.g., Deep Learning)

Label Propagation:

\[
L_{\text{new}}(\tilde{x}) = \frac{\sum_{t=1}^{T} \sum_{\tilde{y} \in \mathcal{N}_t(\tilde{x})} w_t(\tilde{x}, \tilde{y}) L_t(\tilde{y})}{\sum_{t=1}^{T} \sum_{\tilde{y} \in \mathcal{N}_t(\tilde{x})} w_t(\tilde{x}, \tilde{y})}
\]

Sparse Representation:

\[
E(\tilde{\beta}_{\tilde{x}}) = \frac{1}{2} \| \tilde{f}_{\tilde{x}} - A \tilde{\beta}_{\tilde{x}} \|^2_2 + \lambda \| \tilde{\beta}_{\tilde{x}} \|_1, \quad \tilde{\beta}_{\tilde{x}} \geq 0
\]

Sparse code \( \tilde{\beta}_{\tilde{x}} \) is used as weight, \( \tilde{f}_{\tilde{x}} \) is feature vector of voxel \( \tilde{x} \), \( A \) is a dictionary containing \( \tilde{f}_{\tilde{y}} \) for all \( \tilde{y} \in \mathcal{N}_t(\tilde{x}), t = 1 \cdots T \).
Simple units correspond to image filters, and pool units group similar image filters together to increase the robustness of learned features. The goal of ISA is to learn a feature representation that is: 1) sparse, and 2) diverse. Thus, the objective function is defined as:

\[
\arg\min_{W,V} \sum_{i=1}^{N} \sum_{j=1}^{m} R_j(x_i, W, V), \quad \text{where } WW^T = I, \quad \text{where } R_j(x_i, W, V) = \sqrt{\frac{\sum_{l=1}^{k} \sum_{p=1}^{d} W_{lp}^2 x_i^p}{V_{jl}}}^2
\]

\(d, k\) and \(m\) denotes the dimension of each input \(x_i\), number of simple units in the first layer, and the number of pooling units in the second layer, respectively.
Hierarchical ISA network by stacking one-layer ISA

Building hierarchical ISA network by:
1) Divide large patch into small overlapping patches
2) Learn the first layer ISA on small patches
3) Take the learned features of small patches as input to the second layer of ISA after PCA whitening
4) Learn the second-layer high-level features
Visualization of Filters

Gabor-like image filters learned by the first-layer ISA

Feature Difference Maps: Comparing the green-cross voxel with other image voxels using different features. **Blue** indicate low difference, and **red** indicate high difference.
## Quantitative Comparison

- **Comparisons between different multi-atlas based segmentation methods:**
  1. Klein’s Method
  2. Coupe’s Method
  3. Liao’s Method

- **Comparisons between different features:**
  - Haar: Haar features
  - HOG: Histogram of Oriented Gradients
  - LBP: Local Binary Patterns
  - Single ISA: Single Layer ISA
  - Stacked ISA: Two-Layer stacked ISA.
  - SL: Sparse Label Propagation

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean Dice + SD (in %)</th>
<th>Mean Hausdorff + SD (in mm)</th>
<th>Mean ASD + SD (in mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Klein’s Method</td>
<td>83.4±3.1</td>
<td>10.2±2.6</td>
<td>2.5±1.4</td>
</tr>
<tr>
<td>Coupe’s Method</td>
<td>81.7±2.7</td>
<td>12.4±2.8</td>
<td>3.6±1.6</td>
</tr>
<tr>
<td>Liao’s Method</td>
<td>84.4±1.6</td>
<td>11.5±2.2</td>
<td>2.3±1.5</td>
</tr>
<tr>
<td>Haar + SL</td>
<td>84.2±3.4</td>
<td>9.7±3.2</td>
<td>2.3±1.7</td>
</tr>
<tr>
<td>HOG + SL</td>
<td>80.5±2.7</td>
<td>12.2±4.5</td>
<td>3.8±2.2</td>
</tr>
<tr>
<td>LBP + SL</td>
<td>82.6±3.2</td>
<td>10.4±2.2</td>
<td>3.0±1.9</td>
</tr>
<tr>
<td>Haar + HOG + LBP + SL</td>
<td>84.9±3.6</td>
<td>9.8±3.3</td>
<td>2.5±1.8</td>
</tr>
<tr>
<td>Single ISA + SL</td>
<td>84.8±2.5</td>
<td>9.5±2.4</td>
<td>2.2±1.8</td>
</tr>
<tr>
<td>Stacked ISA + SL</td>
<td><strong>86.7±2.2</strong></td>
<td><strong>8.2±2.5</strong></td>
<td><strong>1.9±1.6</strong></td>
</tr>
</tbody>
</table>
Thank You!
For more details, please visit:
http://bric.unc.edu/ideagroup
Or google: idea unc