

# STACKED DENOISING AUTO-ENCODER

**2016. 08. 22.**

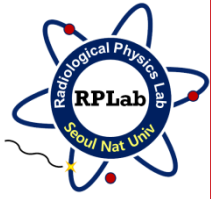
**JIMIN LEE**

# CONTENTS



1. Auto-Encoder
2. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion
3. Brain Tissues from Chinese Visible Human Dataset by Deep-Learned Features with **Stacked Autoencoder** (Medical Research Paper 1)
4. Adaptive **Multi-Column Deep Neural Networks** with Application to Robust Image Denoising (Medical Research Paper 2)

# AUTO-ENCODER



- **Auto-Encoder**

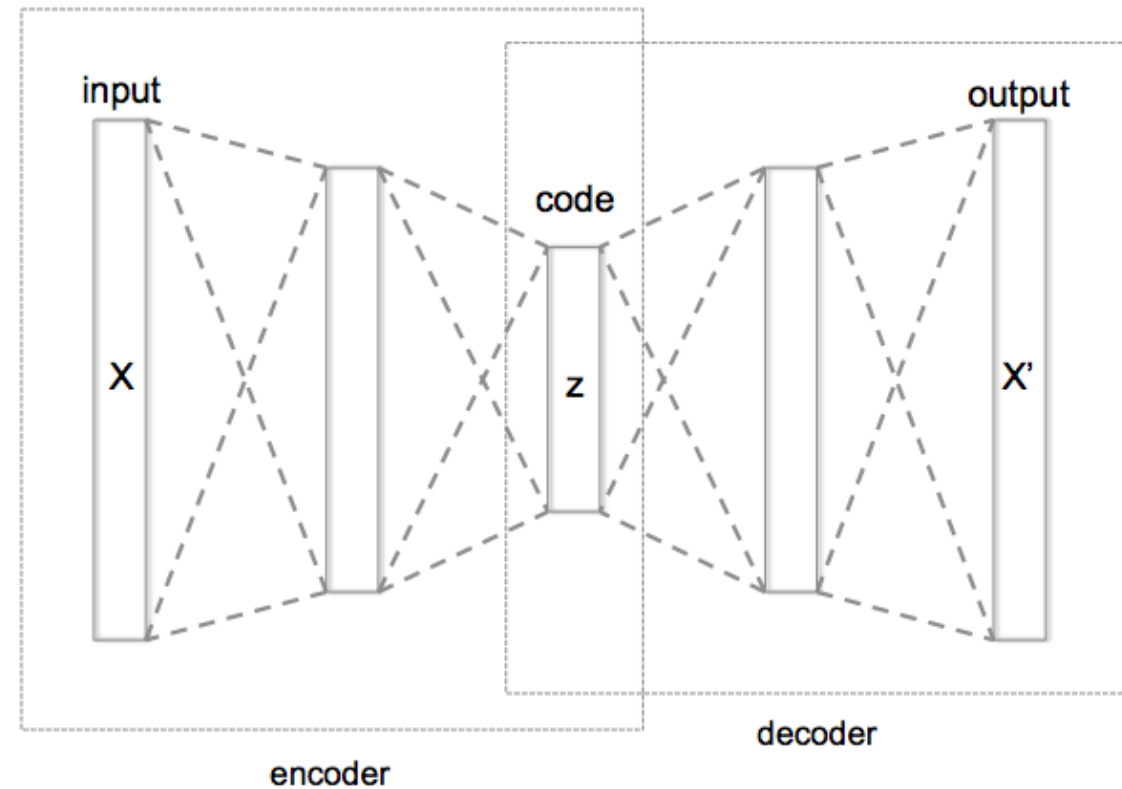
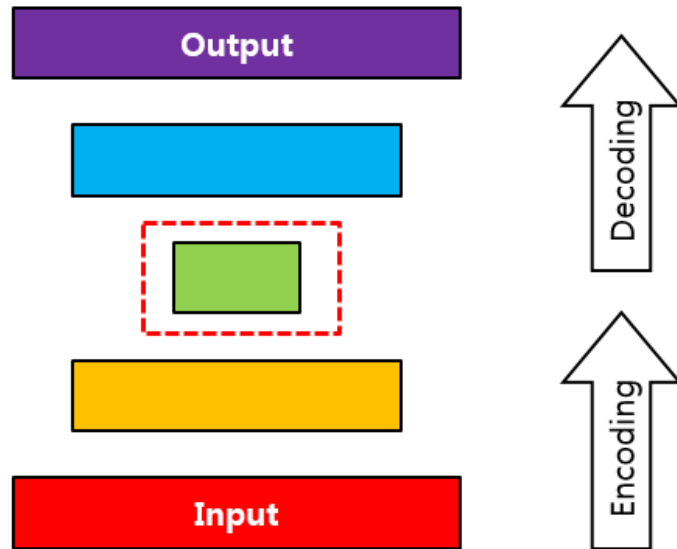
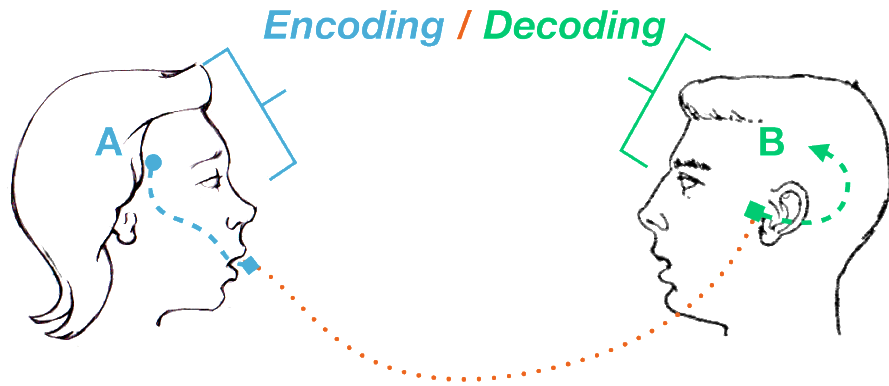
- Auto-Encoder
- Sparse Auto-Encoder
- Stacked Auto-Encoder (SAE)
- Denoising Auto-Encoder
- Stacked Denoising Auto-Encoder (SDAE)

⋮

- Linear Auto-Encoder
- Contractive Auto-Encoder

# AUTO-ENCODER

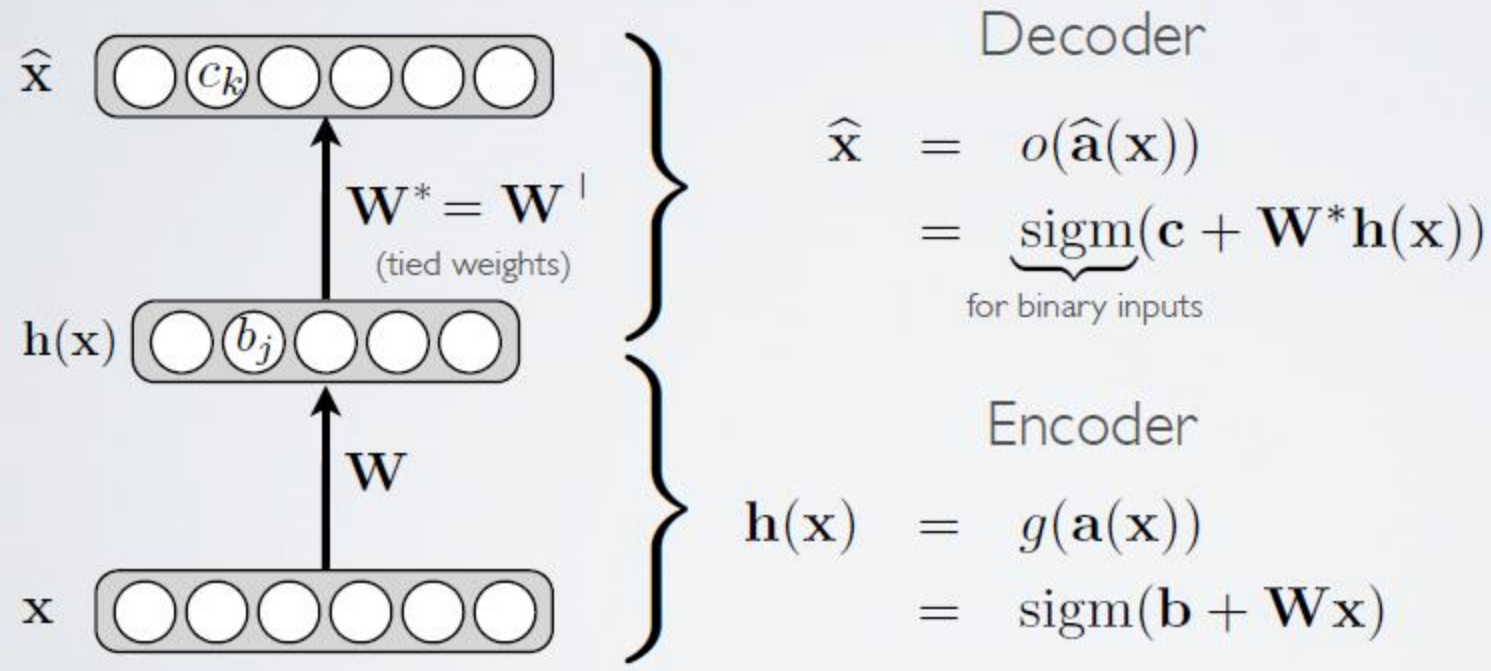
- Auto-Encoder



# AUTO-ENCODER

- Auto-Encoder

- Feed-forward neural network trained to reproduce its input at the output layer



# AUTO-ENCODER



- **Loss function**

**Topics:** loss function

- For binary inputs:

$$f(\mathbf{x}) \equiv \hat{\mathbf{x}}$$

$$l(f(\mathbf{x})) = - \sum_k (x_k \log(\hat{x}_k) + (1 - x_k) \log(1 - \hat{x}_k))$$

- cross-entropy (more precisely: sum of Bernoulli cross-entropies)

- For real-valued inputs:

$$l(f(\mathbf{x})) = \frac{1}{2} \sum_k (\hat{x}_k - x_k)^2$$

- sum of squared differences (squared euclidean distance)
- we use a linear activation function at the output



# AUTO-ENCODER

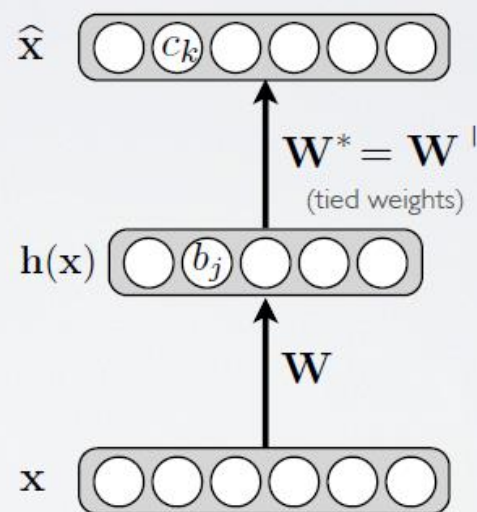
## ▪ Undercomplete / Overcomplete representation

- Hidden layer is undercomplete if smaller than the input layer

- ▶ hidden layer “compresses” the input
- ▶ will compress well only for the training distribution

- Hidden units will be

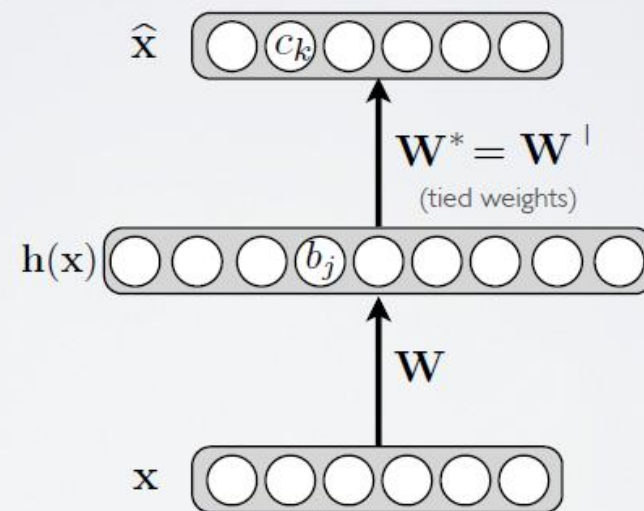
- ▶ good features for the training distribution → 
- ▶ but bad for other types of input → 



- Hidden layer is overcomplete if greater than the input layer

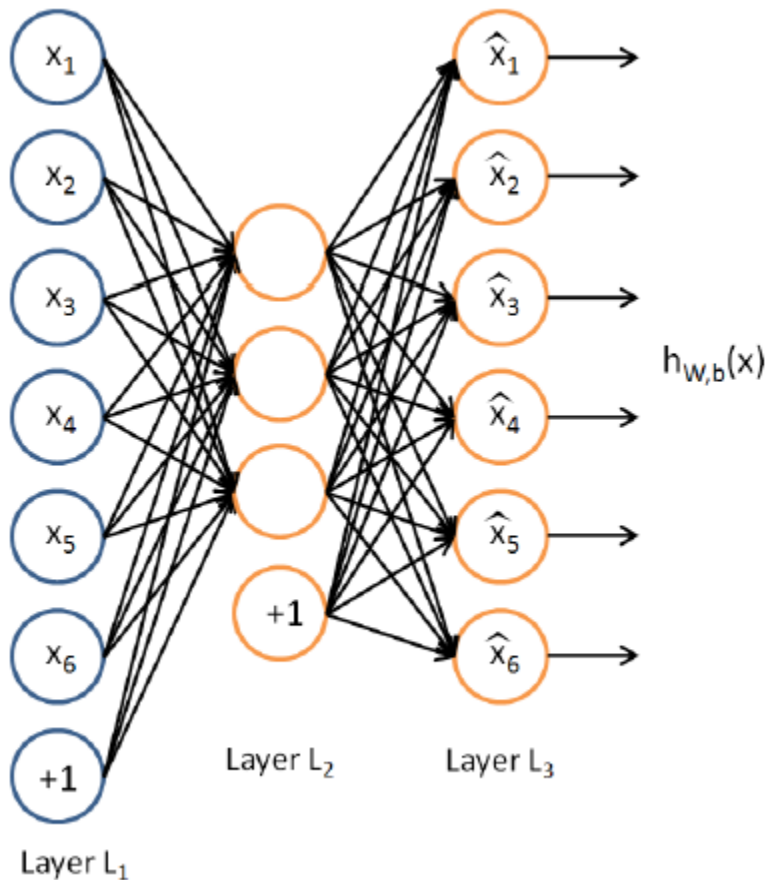
- ▶ no compression in hidden layer
- ▶ each hidden unit could copy a different input component

- No guarantee that the hidden units will extract meaningful structure



# AUTO-ENCODER

## ▪ Sparse Auto-Encoder



$$\hat{\rho}_j = \frac{1}{m} \sum_{i=1}^m \left[ a_j^{(2)}(x^{(i)}) \right]$$

$a_j^{(2)}$  : activation of hidden unit  $j$  in the autoencoder

$\hat{\rho}_j$  : average activation of hidden unit  $j$  (averaged over the training set)

$$\hat{\rho}_j = \rho$$

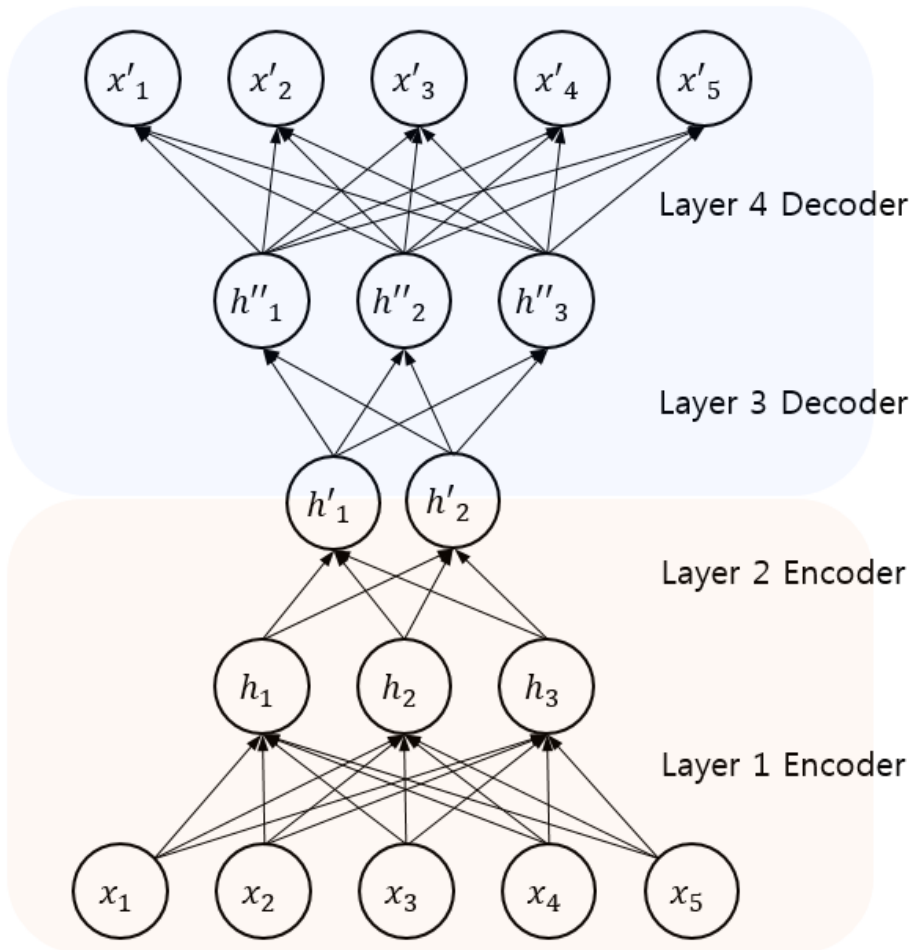
$\rho$  : **sparsity parameter**, typically a small value close to zero

If  $\rho$  is 0.05, the average activation of each hidden neuron  $j$  to be close to 0.05. To satisfy this constraint, **the hidden unit's activations must mostly be near 0.**



# AUTO-ENCODER

## Stacked Auto-Encoder (SAE)



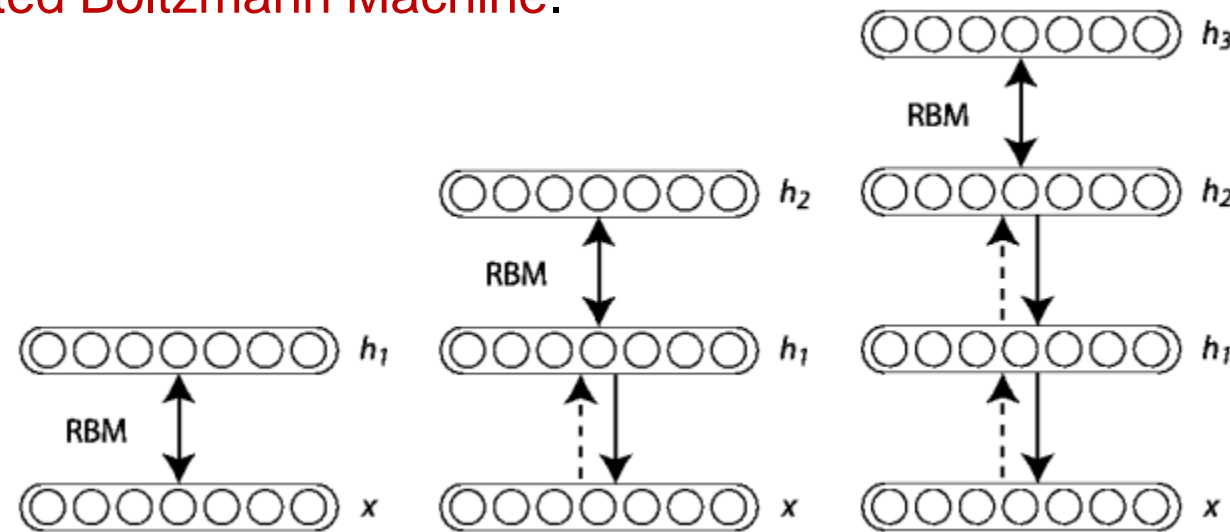
### ★ Stacked Auto-Encoder와 Auto-Encoder의 차이점

- **DBN (Deep Belief Network)**의 구조
- 기존 AE는 BP를 통해 Weight 학습
- Layer와 Unit의 개수가 많아질수록 방대한 계산량과 **Local Minima**에 빠질 위험, **Vanishing Gradient**의 문제 존재

# AUTO-ENCODER

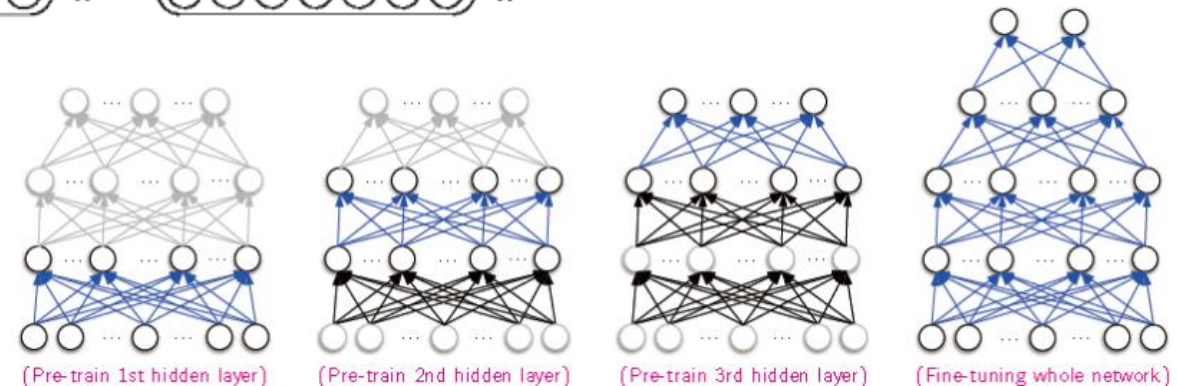
- **Stacked Auto-Encoder (SAE) ; DBN (Deep Belief Network)**

- Each layer of the network tries to model the distribution of its input, using unsupervised training in a **Restricted Boltzmann Machine**.



- It is trained in a **greedy layer-wise** fashion.

- First step : Pre-training
- Second step : Fine-tuning



# AUTO-ENCODER



## Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion

**Pascal Vincent**

PASCAL.VINCENT@UMONTREAL.CA

*Département d'informatique et de recherche opérationnelle  
Université de Montréal  
2920, chemin de la Tour  
Montréal, Québec, H3T 1J8, Canada*

**Hugo Larochelle**

LAROCHEH@CS.TORONTO.EDU

*Department of Computer Science  
University of Toronto  
10 King's College Road  
Toronto, Ontario, M5S 3G4, Canada*

**Isabelle Lajoie**

ISABELLE.LAJOIE.1@UMONTREAL.CA

**Yoshua Bengio**

YOSHUA.BENGIO@UMONTREAL.CA

**Pierre-Antoine Manzagol**

PIERRE-ANTOINE.MANZAGOL@UMONTREAL.CA

*Département d'informatique et de recherche opérationnelle  
Université de Montréal  
2920, chemin de la Tour  
Montréal, Québec, H3T 1J8, Canada*

**Editor:** Léon Bottou

# AUTO-ENCODER

- Denoising Auto-Encoder

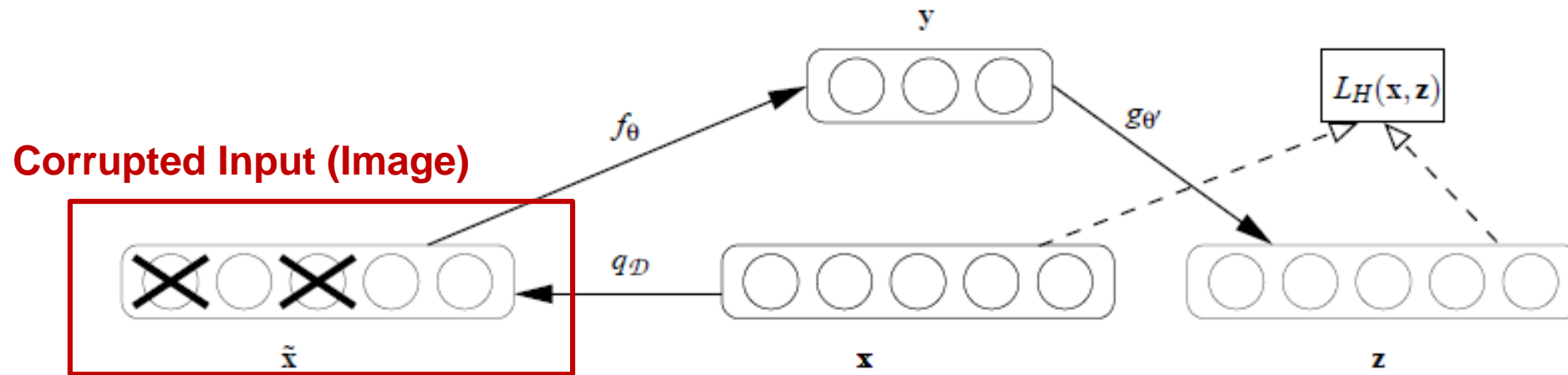


Figure 1: The denoising autoencoder architecture. An example  $\mathbf{x}$  is stochastically corrupted (via  $q_{\mathcal{D}}$ ) to  $\tilde{\mathbf{x}}$ . The autoencoder then maps it to  $y$  (via encoder  $f_{\theta}$ ) and attempts to reconstruct  $\mathbf{x}$  via decoder  $g_{\theta'}$ , producing reconstruction  $\mathbf{z}$ . Reconstruction error is measured by loss  $L_H(\mathbf{x}, \mathbf{z})$ .

# AUTO-ENCODER

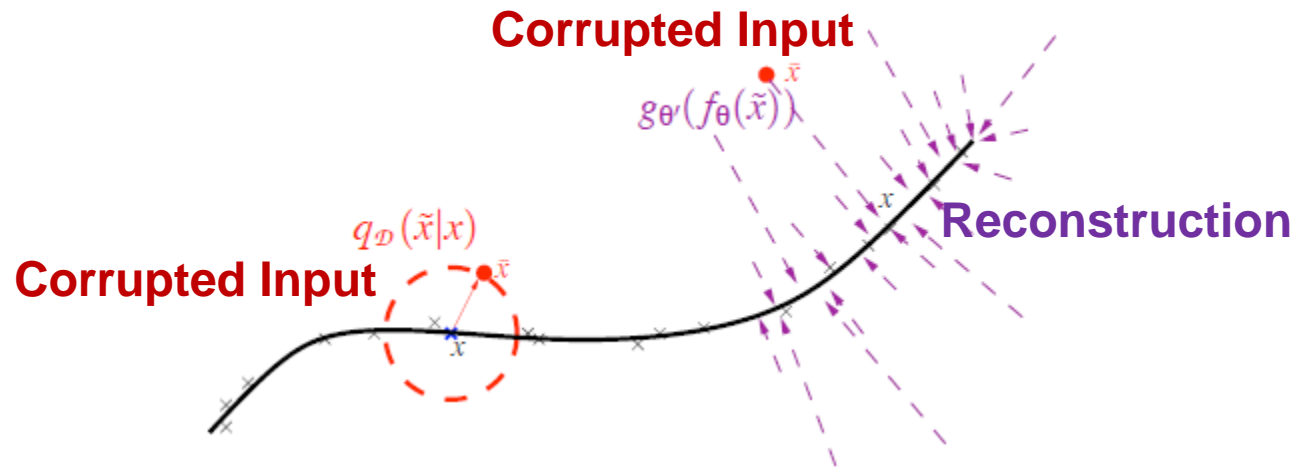


Figure 2: Manifold learning perspective. Suppose training data ( $\times$ ) concentrate near a low-dimensional manifold. Corrupted examples ( $\bullet$ ) obtained by applying corruption process  $q_{\mathcal{D}}(\tilde{X}|X)$  will generally lie farther from the manifold. The model learns with  $p(X|\tilde{X})$  to “project them back” (via autoencoder  $g'_{\theta}(f_{\theta}(\cdot))$ ) onto the manifold. Intermediate representation  $Y = f_{\theta}(X)$  may be interpreted as a coordinate system for points  $X$  on the manifold.

- Learns a vector field towards higher probability regions
- Minimizes variational lower bound on a generative model
- Corresponds to regularized score matching on an RBM

# AUTO-ENCODER

- Stacked Denoising Auto-Encoder (SDAE)

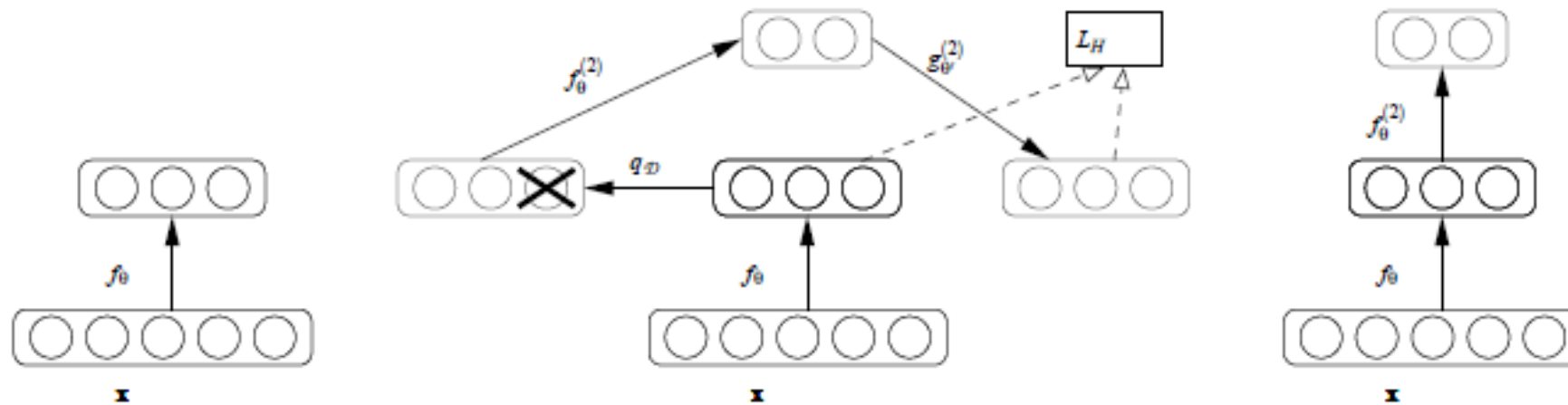
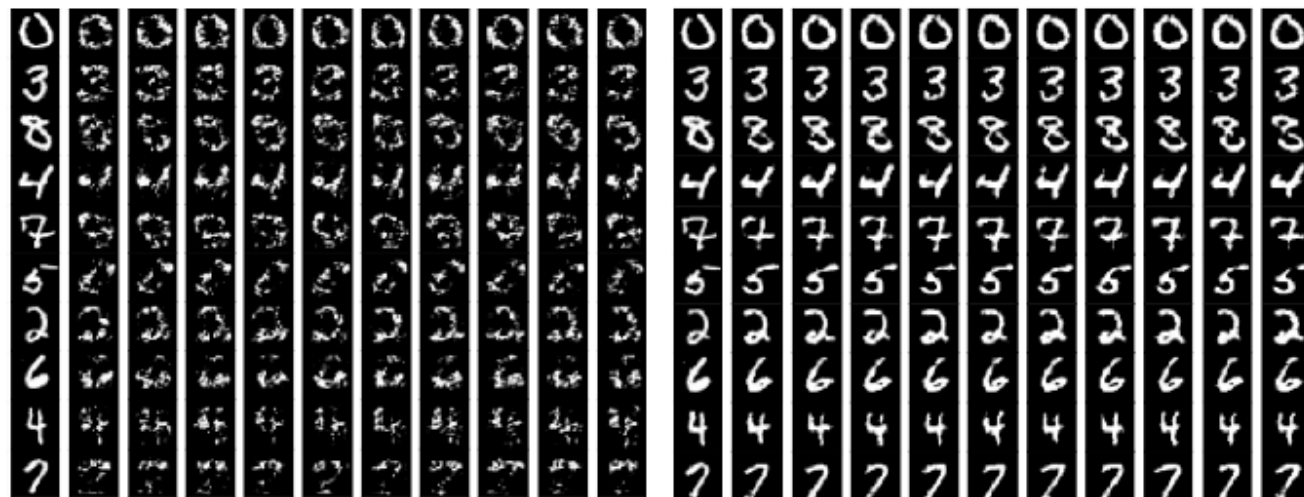


Figure 3: Stacking denoising autoencoders. After training a first level denoising autoencoder (see Figure 1) its learnt encoding function  $f_\theta$  is used on clean input (left). The resulting representation is used to train a second level denoising autoencoder (middle) to learn a second level encoding function  $f_\theta^{(2)}$ . From there, the procedure can be repeated (right).

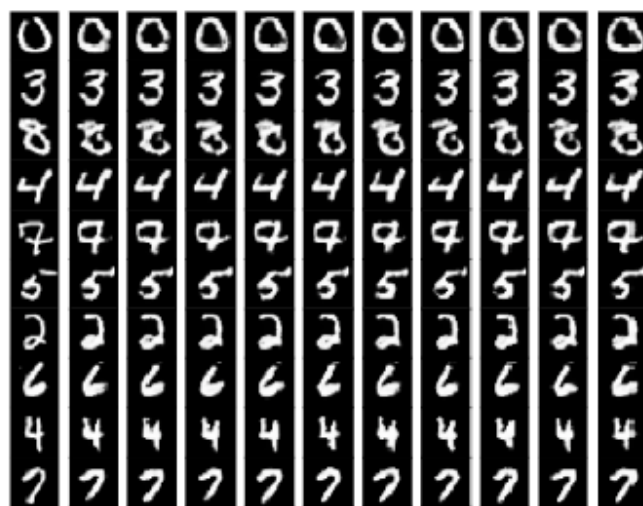


# AUTO-ENCODER



(a) SAE

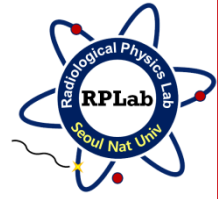
(b) SDAE



(c) DBN

Figure 15: Variability of the samples generated with 3-hidden-layer SAE, SDAE and DBN pre-trained models. Each sub-figure is to be read row-wise: the leftmost pattern in each row is a training set pattern. Following the sample generation depicted in Figure 14, it was provided as input to the network and its top-layer representation was computed by deterministic bottom up encoding. Patterns to its right were then generated independently given that top level representation. Clearly, SDAE trained networks, like DBNs, are able to regenerate high quality samples from their high level representation, contrary to SAE. SDAE and DBNs also appear to give rise to a similar level of variability in the bottom-up generated patterns (DBN patterns tending to be somewhat fatter). Note how SDAE puts back the missing hole in the loop of the regenerated 6, and sometimes straightens up the upper stroke of the last 7, suggesting that it did indeed capture meaningful specific characteristics. DBN and SDAE generated patterns can easily pass for samples from the unknown input distribution being modeled, unlike patterns generated by SAE.

# MEDICAL RESEARCH PAPER 1



Hindawi Publishing Corporation  
BioMed Research International  
Volume 2016, Article ID 5284586, 12 pages  
<http://dx.doi.org/10.1155/2016/5284586>



## Research Article

### Segmenting Brain Tissues from Chinese Visible Human Dataset by Deep-Learned Features with Stacked Autoencoder

Guangjun Zhao,<sup>1</sup> Xuchu Wang,<sup>1</sup> Yanmin Niu,<sup>2</sup> Liwen Tan,<sup>3</sup> and Shao-Xiang Zhang<sup>3</sup>

<sup>1</sup>Key Laboratory of Optoelectronic Technology and Systems of Ministry of Education, College of Optoelectronic Engineering, Chongqing University, Chongqing 400044, China

<sup>2</sup>College of Computer and Information Science, Chongqing Normal University, Chongqing 400050, China

<sup>3</sup>Institute of Digital Medicine, College of Biomedical Engineering, Third Military Medical University, Chongqing 400038, China

Correspondence should be addressed to Xuchu Wang; [seadrift.wang@gmail.com](mailto:seadrift.wang@gmail.com)

Received 2 November 2015; Revised 18 December 2015; Accepted 27 December 2015

Academic Editor: Sher Afzal Khan

Copyright © 2016 Guangjun Zhao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

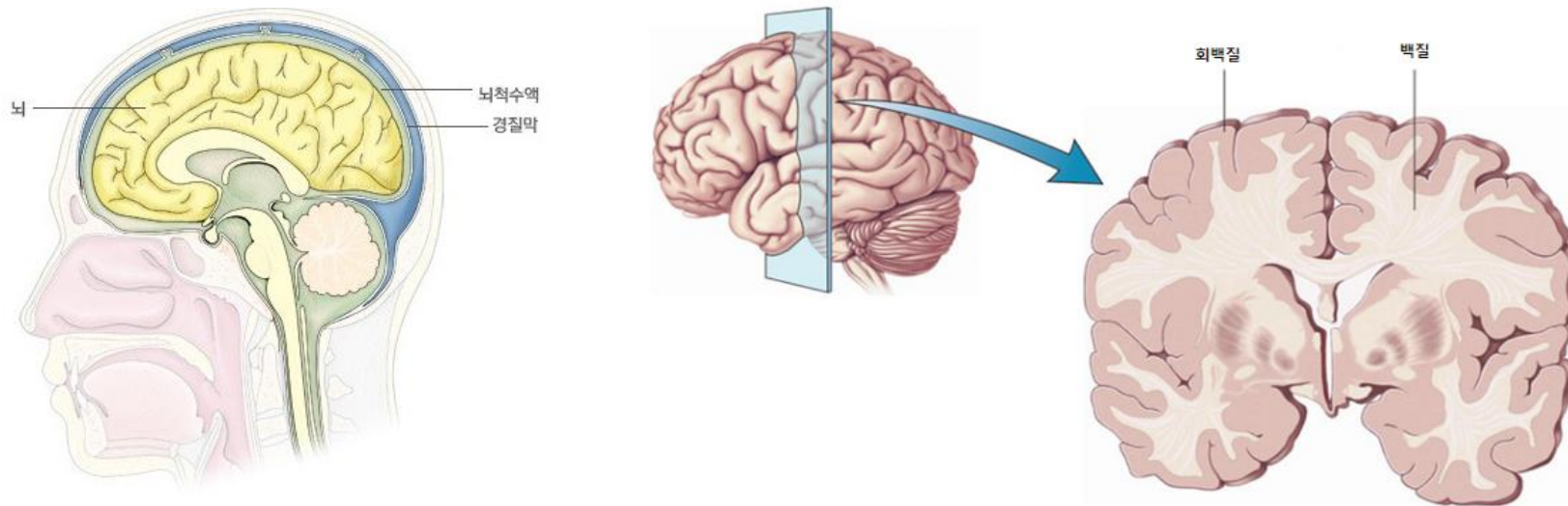
Cryosection brain images in Chinese Visible Human (CVH) dataset contain rich anatomical structure information of tissues because of its high resolution (e.g., 0.167 mm per pixel). Fast and accurate segmentation of these images into white matter, gray matter, and cerebrospinal fluid plays a critical role in analyzing and measuring the anatomical structures of human brain. However, most existing automated segmentation methods are designed for computed tomography or magnetic resonance imaging data, and they may not be applicable for cryosection images due to the imaging difference. In this paper, we propose a supervised learning-based CVH brain tissues segmentation method that uses stacked autoencoder (SAE) to automatically learn the deep feature representations. Specifically, our model includes two successive parts where two three-layer SAEs take image patches as input to learn the complex anatomical feature representation, and then these features are sent to Softmax classifier for inferring the labels. Experimental results validated the effectiveness of our method and showed that it outperformed four other classical brain tissue detection strategies. Furthermore, we reconstructed three-dimensional surfaces of these tissues, which show their potential in exploring the high-resolution anatomical structures of human brain.



# MEDICAL RESEARCH PAPER 1

## ■ Introduction

- To segment Chinese Visible Human (CVH) brain tissues into cerebrospinal fluid (CSF-뇌척수액), gray matter (GM-회백질) and white matter (WM-백질)
- Automatic or semiautomatic segmentation is helpful for alleviating the **laborious and time-consuming manual segment**.
- **Much noise** is introduced during CVH image acquisition and the **image contrast is low** at some positions.



# MEDICAL RESEARCH PAPER 1

## ■ Material and Methods

- Dataset : Successive cross-sectional images of human brain from the CVH dataset provided by the Third Military Medical University in China
- A total of 422 cross-sectional images of the head

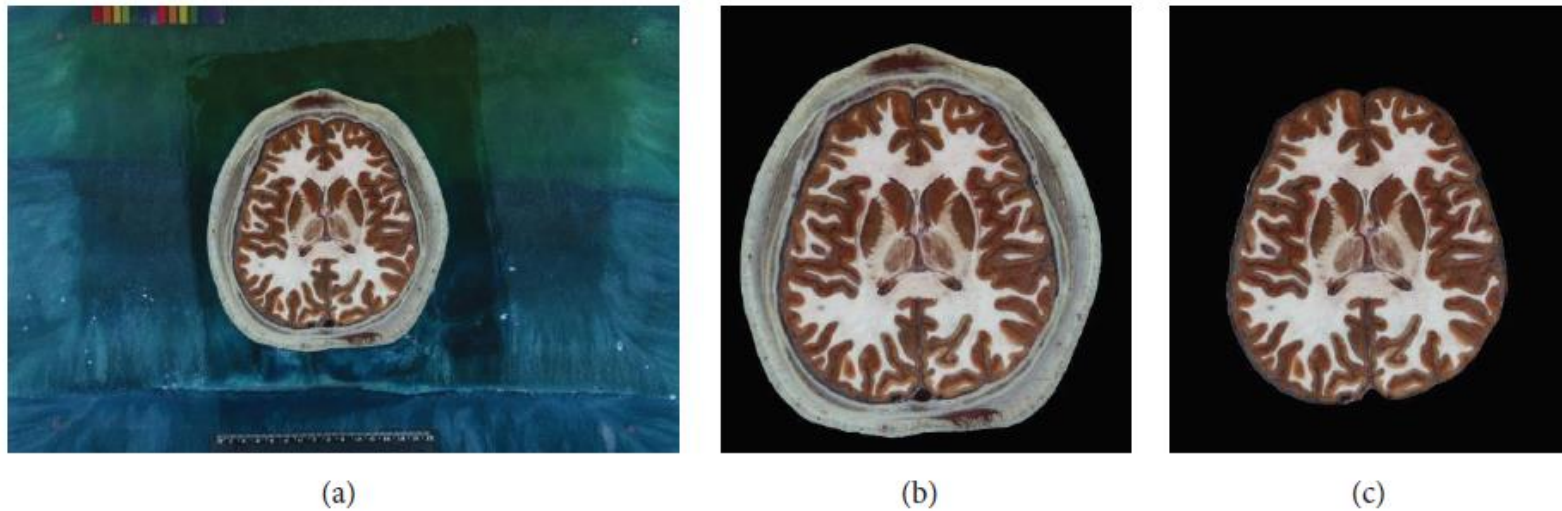


FIGURE 1: Preprocessing example of cryosection brain image. (a) Original image without any preprocessing ( $3,072 \times 2,048$  pixels). (b) Cropped image ( $1,252 \times 1,364$  pixels). (c) Skull stripped image.

# MEDICAL RESEARCH PAPER 1

## Material and Methods

- Method : Patch classification task
- SAE : To extract intrinsic feature representation of the input patches
- Softmax classifier : To generate a labels distribution based on deep features

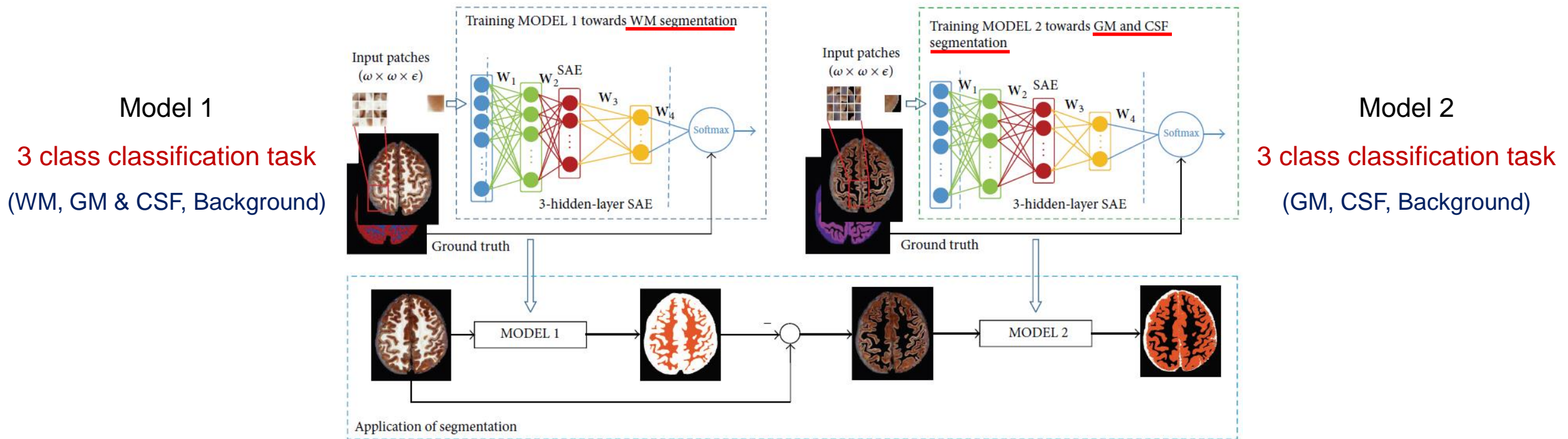


FIGURE 2: Flowchart of our segmentation model.

# MEDICAL RESEARCH PAPER 1

- **Material and Methods**
  - Stacked Auto-Encoder (SAE)

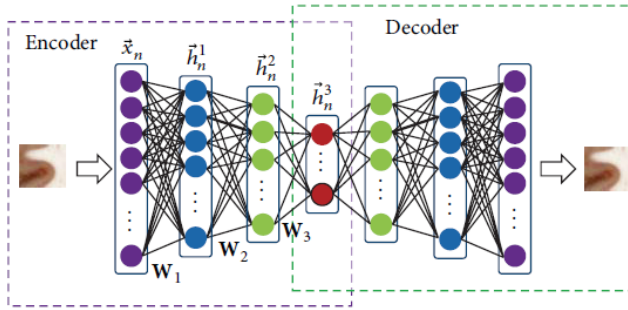


FIGURE 3: Proposed three-hidden-layer SAE. Note that the number of layers in our model is set via cross-validation.

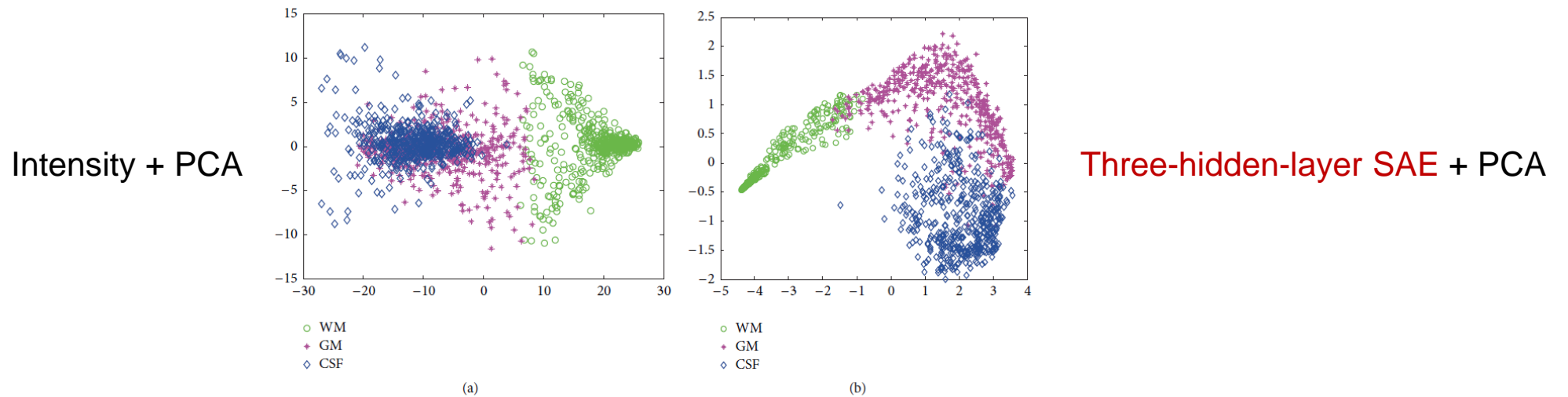


FIGURE 4: Two-dimensional feature representation for 500 patches of each brain tissue by (a) intensity + PCA and (b) intensity + three-hidden-layer SAE + PCA; here PCA is just for visualization of principle components.



# MEDICAL RESEARCH PAPER 1

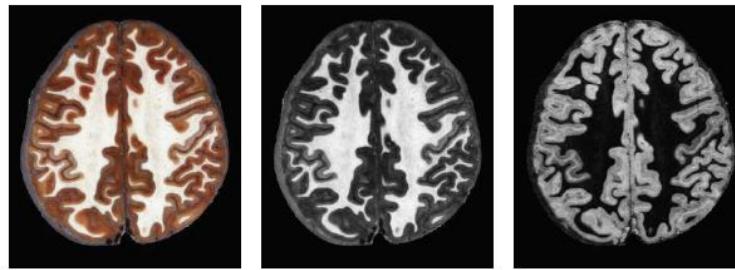
## Results and Discussion

TABLE 3: Mean and standard deviation of Dice ratio (in %) for the segmentations obtained by five feature representation methods.

Method	CSF	GM	WM
Intensity	88.76 ± 2.41	89.46 ± 2.12	95.49 ± 1.96
PCA	89.31 ± 1.89	90.87 ± 2.34	95.20 ± 2.01
HOG	86.77 ± 2.67	87.21 ± 2.86	94.92 ± 1.89
AE	88.32 ± 2.37	90.26 ± 2.36	95.60 ± 1.83
Proposed SAE	90.69 ± 2.14	91.24 ± 2.01	96.12 ± 1.23

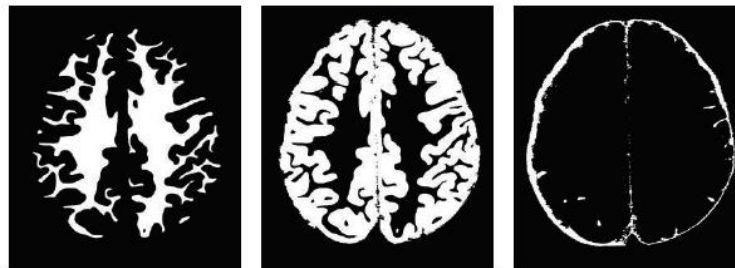
$$\text{Dice Ratio} = \frac{2|X \cap Y|}{|X| + |Y|}$$

Original Images



(a)

Manual Segmentation



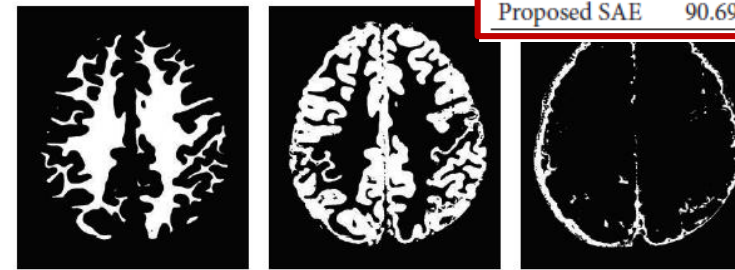
(b)

SAE



(c)

Intensity



(d)

PCA



(e)

HOG

(Histogram of Gradient)



(f)

# MEDICAL RESEARCH PAPER 1

- Results and Discussion
  - 3D surface-rendering reconstruction



FIGURE 9: Three-dimensional surface-rendering reconstruction results of WM, GM, and CSF based on our segmented images.

# MEDICAL RESEARCH PAPER 2



---

## Adaptive Multi-Column Deep Neural Networks with Application to Robust Image Denoising

---

(2013)

Forest Agostinelli      Michael R. Anderson      Honglak Lee  
Division of Computer Science and Engineering  
University of Michigan  
Ann Arbor, MI 48109, USA  
{agostifo, mrande, honglak}@umich.edu

### Abstract

Stacked sparse denoising autoencoders (SSDAs) have recently been shown to be successful at removing noise from corrupted images. However, like most denoising techniques, the SSDA is not robust to variation in noise types beyond what it has seen during training. To address this limitation, we present the adaptive multi-column stacked sparse denoising autoencoder (AMC-SSDA), a novel technique of combining multiple SSDAs by (1) computing optimal column weights via solving a nonlinear optimization program and (2) training a separate network to predict the optimal weights. We eliminate the need to determine the type of noise, let alone its statistics, at test time and even show that the system can be robust to noise not seen in the training set. We show that state-of-the-art denoising performance can be achieved with a single system on a variety of different noise types. Additionally, we demonstrate the efficacy of AMC-SSDA as a pre-processing (denoising) algorithm by achieving strong classification performance on corrupted MNIST digits.

# MEDICAL RESEARCH PAPER 2

## ■ AMC-SSDA

- SSDA(Stacked Sparse Denoising AE) is not robust to variation in noise types.
- To overcome this limitation, we present the **adaptive multi-column stacked sparse denoising AE (AMC-SSDA)**.

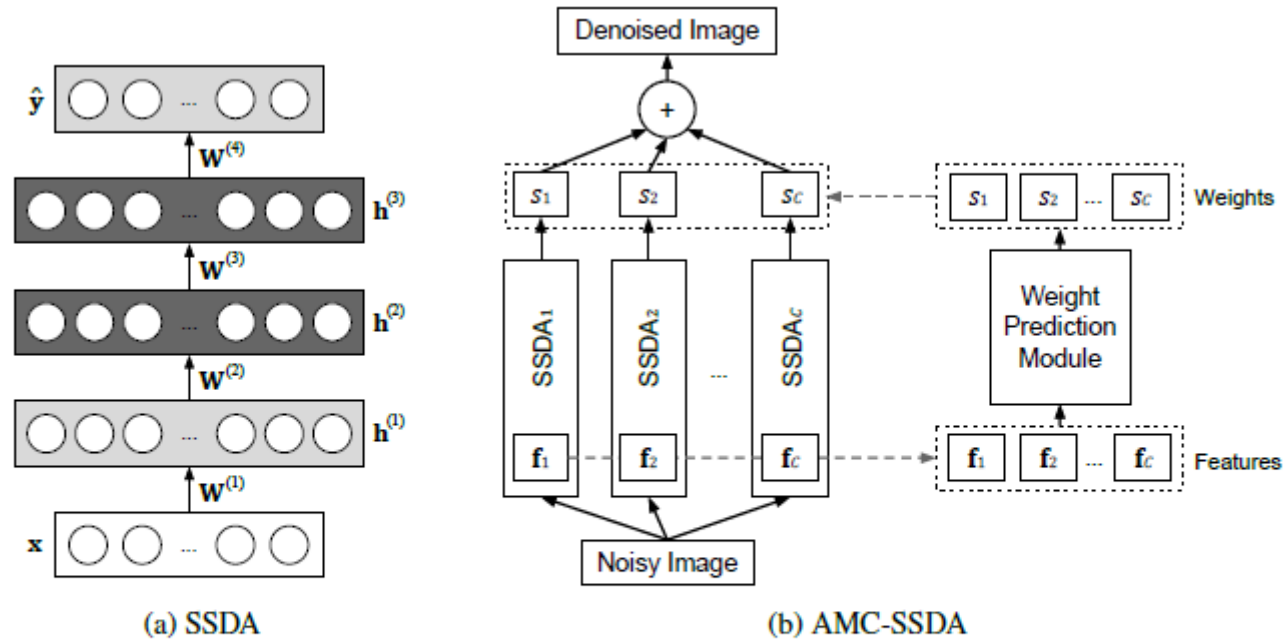


Figure 1: Illustration of the AMC-SSDA. We concatenate the activations of the first-layer hidden units of the SSDA in each column (i.e.,  $f_c$  denotes the concatenated hidden unit vectors  $h^{(1)}(x)$  and  $h^{(2)}(x)$  of the SSDA corresponding to  $c$ -th column) as input features to the weight prediction module for determining the optimal weight for each column of the AMC-SSDA. See text for details.



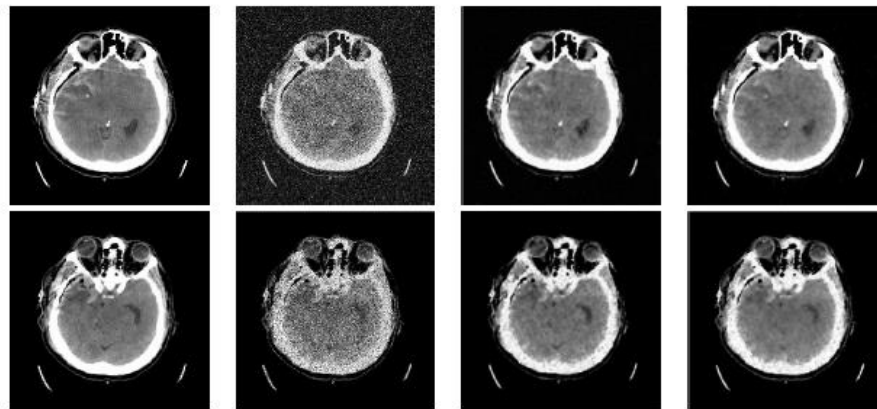
# MEDICAL RESEARCH PAPER 2

## Results

### - Noise type

Noise Type	1	2	3	4
Gaussian	$\sigma^2 = 0.01$	$\sigma^2 = 0.07$	$\sigma^2 = 0.1$	$\sigma^2 = 0.25$
Speckle	$\rho = 0.1$	$\rho = 0.15$	$\rho = 0.3$	$\rho = 0.4$
Salt & Pepper	$\rho = 0.1$	$\rho = 0.15$	$\rho = 0.3$	$\rho = 0.4$
Poisson	$\log(\lambda) = 24.4$	$\log(\lambda) = 25.3$	$\log(\lambda) = 26.0$	$\log(\lambda) = 26.4$
Uniform [-0.5, 0.5]	30%	50%	70%	100%

Table 2: Parameters of noise types used for testing. The Poisson and uniform noise types are not seen in the training set. The percentage for uniform noise denotes how many pixels are affected.  $\rho$  is the noise density.



(a) Original

(b) Noisy

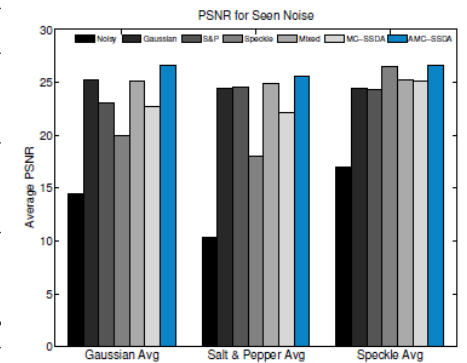
(c) Mixed-SSDA

(d) AMC-SSDA

Figure 2: Visualization of the denoising performance of the Mixed-SSDA and AMC-SSDA. Top: Gaussian noise. Bottom: speckle noise.

Noise Type	Noisy Image	Gaussian SSDA	S&P SSDA	Speckle SSDA	Mixed SSDA	MC-SSDA	AMC-SSDA
G 1	22.10	26.64	26.69	26.84	27.15	27.37	<b>29.60</b>
G 2	13.92	25.83	23.07	19.76	25.52	23.34	<b>26.85</b>
G 3	12.52	25.50	22.17	18.35	25.09	22.00	<b>26.10</b>
G 4	9.30	23.11	20.17	14.88	22.72	17.97	<b>23.66</b>
SP 1	13.50	25.86	26.26	22.27	26.32	25.84	<b>27.72</b>
SP 2	11.76	25.40	25.77	20.07	25.77	24.54	<b>26.77</b>
SP 3	8.75	23.95	23.96	15.88	24.32	20.42	<b>24.65</b>
SP 4	7.50	22.46	22.20	13.86	22.95	17.76	<b>23.01</b>
S 1	19.93	26.41	26.37	28.22	26.97	27.43	<b>28.59</b>
S 2	18.22	25.92	25.80	27.75	26.44	26.71	<b>27.68</b>
S 3	15.35	23.54	23.36	25.79	24.42	23.91	<b>25.72</b>
S 4	14.24	21.80	21.69	24.41	22.93	22.20	<b>24.35</b>
Avg	13.92	24.70	23.96	21.51	25.05	23.29	<b>26.23</b>

(a) PSNRs for previously seen noise, best values in bold.

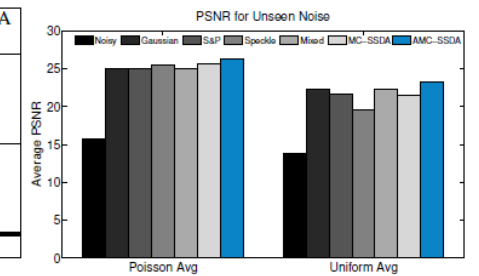


(b) Average PSNRs for specific noise types

Figure 3: Average PSNR values for denoised images of various previously seen noise types (G: Gaussian, S: Speckle, SP: Salt & Pepper).

Noise Type	Noisy Image	Gaussian SSDA	S&P SSDA	Speckle SSDA	Mixed SSDA	MC-SSDA	AMC-SSDA
P 1	19.90	26.27	26.48	27.99	26.80	27.35	<b>28.83</b>
P 2	16.90	25.77	25.92	26.94	26.01	26.78	<b>27.64</b>
P 3	13.89	24.61	24.54	24.65	24.43	25.11	<b>25.50</b>
P 4	12.11	23.36	23.07	22.64	23.01	23.28	<b>23.43</b>
U 1	17.20	23.40	23.68	25.05	23.74	24.71	<b>24.50</b>
U 2	16.04	26.21	25.86	23.21	26.28	26.13	<b>28.06</b>
U 3	12.98	23.24	21.36	17.83	22.89	21.07	<b>23.70</b>
U 4	8.78	16.54	15.45	12.01	16.04	14.11	<b>16.78</b>
Avg	14.72	23.67	23.29	22.54	23.65	23.57	<b>24.80</b>

(a) PSNR for unseen noise, best values in bold.



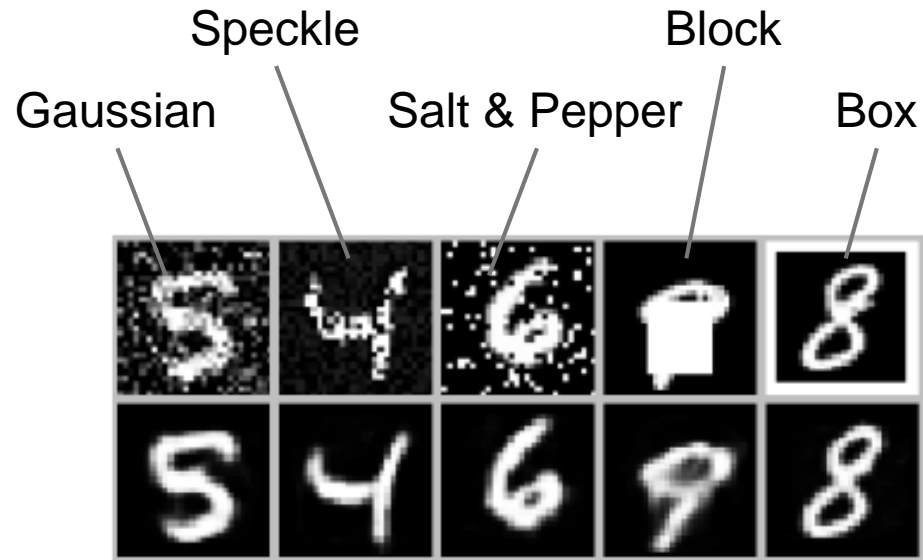
(b) Average results for noise types.

Figure 4: Average PSNR values for denoised images of various previously unseen noise types (P: Poisson noise; U: Uniform noise).

# MEDICAL RESEARCH PAPER 2

## Results

- Digit recognition from denoised images



Method / Noise Type	Clean	Gaussian	S & P	Speckle	Block	Border	Average
No denoising	<b>1.09%</b>	29.17%	18.63%	8.11%	25.72%	90.05%	28.80%
Gaussian SSDA	2.13%	<b>1.52%</b>	2.44%	5.10%	20.03%	8.69%	6.65%
Salt & Pepper SSDA	1.94%	1.71%	2.38%	4.78%	19.71%	2.16%	5.45%
Speckle SSDA	1.58%	5.86%	6.80%	<b>2.03%</b>	19.95%	7.36%	7.26%
Block SSDA	1.67%	5.92%	15.29%	7.64%	<b>5.15%</b>	1.81%	6.25%
Border SSDA	8.42%	19.87%	19.45%	13.89%	31.38%	<b>1.12%</b>	15.69%
AMC-SSDA	1.50%	<b>1.47%</b>	<b>2.22%</b>	<b>2.09%</b>	<b>5.18%</b>	<b>1.15%</b>	<b>2.27%</b>
Tang et al. [28]*	1.24%	-	-	-	19.09%	1.29%	-

Table 3: MNIST test classification error of denoised images. Rows denote the performance of different denoising methods, including: “no denoising,” SSDA trained on a specific noise type, and AMC-SSDA. Columns represent images corrupted with the given noise type. Percentage values are classification error rates for a set of test images corrupted with the given noise type and denoised prior to classification. Bold-faced values represent the best performance for images corrupted by a given noise type. \*Note: we compare the numbers reported from Tang et al. [28] (“7x7+denoised”).



**THANK YOU FOR YOUR ATTENTION**