

# An empirical comparison of CNNs and other methods for classification of protein subcellular localization with microscopy images

Mengli Xiao, Wei Pan

Division of Biostatistics  
University of Minnesota

June 5, 2018

# Outline

- 1 Background
  - Protein subcellular localization
  - Data description
  - Implementation
- 2 A Convolutional Neural Network: DeepYeast
  - CNN model structure
  - Result
- 3 Residual Neural Network
  - ResNet model structures
  - Result
- 4 Feature extraction and transfer learning
  - Definition
  - Result
- 5 Summary
  - Comparison of different methods
  - Discussion

- 1 Background
  - Protein subcellular localization
  - Data description
  - Implementation
- 2 A Convolutional Neural Network: DeepYeast
  - CNN model structure
  - Result
- 3 Residual Neural Network
  - ResNet model structures
  - Result
- 4 Feature extraction and transfer learning
  - Definition
  - Result
- 5 Summary
  - Comparison of different methods
  - Discussion

# Protein subcellular localization

- A protein's subcellular localization  $\Leftrightarrow$  function
- Spatial temporal variation of a protein's location results from genetic and environmental perturbations
- High-throughput imaging: image classification
  - proteins are fluorescently labeled to track their locations within a cell;
- Why **automating**?
  - Manual: labor-intensive and error-prone.
- Large, but not so large, amounts of data: deep learning? others?

# Data description

- Data: Pärnamaa and Parts (2017);
- Each image contains a single cell.

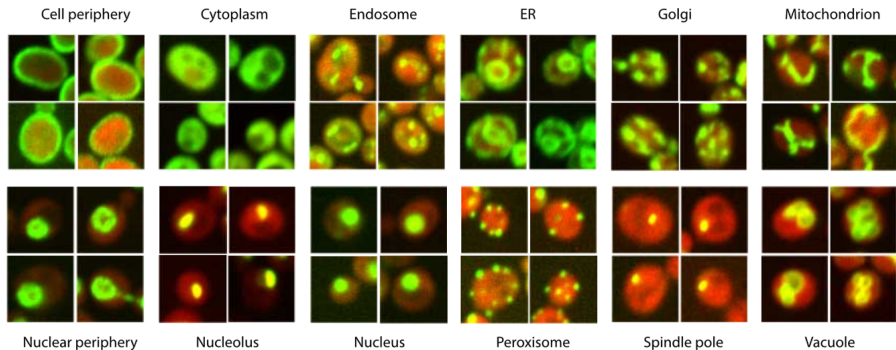


Figure 1: DeepYeast dataset overview with 4 images per category (Pärnamaa and Parts, 2017)

# Data description

Table 1: Data: sample sizes

Subcellular categories	training	validation	test
Cell periphery	6924	961	1569
Cytoplasm	6935	1223	1276
Endosome	2692	697	689
ER	6195	1393	1755
Golgi	2770	208	382
Mitochondria	6547	1560	1243
Nuclear Periphery	6661	1252	1164
Nucleolus	7014	1147	1263
Nuclei	6440	1312	1627
Peroxisome	1683	297	164
Spindle	4713	1517	781
Vacuole	6426	936	587
Total	65000	12500	12500

# Implementation

- Keras in Tensorflow - CNNs
- Python sklearn - RF, XGBoost
- R - CATCH (Pan et al., 2018a,b)

- 1 Background
  - Protein subcellular localization
  - Data description
  - Implementation
- 2 A Convolutional Neural Network: DeepYeast
  - CNN model structure
  - Result
- 3 Residual Neural Network
  - ResNet model structures
  - Result
- 4 Feature extraction and transfer learning
  - Definition
  - Result
- 5 Summary
  - Comparison of different methods
  - Discussion



# DeepYeast (11-layer CNN) Model structure

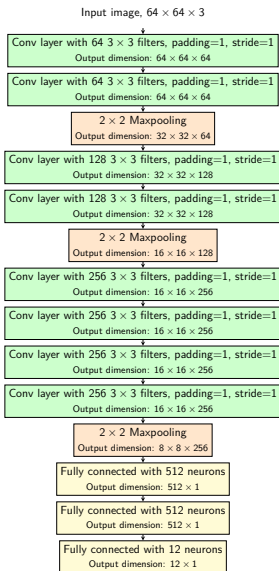
- A 11-layered CNN; similar to the first few layers of VGG-19;
- VGG-19 was trained on the (ImageNet) ILSVRC dataset consisting of natural objects, aircraft, etc.
- Several papers: similar CNNs for the current problem.

# VGG-19 and DeepYeast (11-layered) model structure

Table 2: VGG-19 and DeepYeast model structure

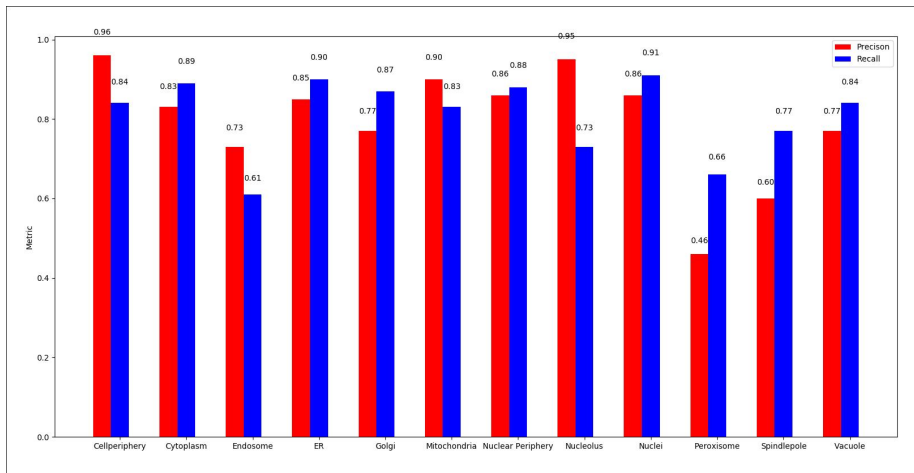
VGG-19	DeepYeast
19 weight layers	11 weight layers
Input: $224 \times 224 \times 3$	Input: $64 \times 64 \times 3$
conv3-64	conv3-64
conv3-64	conv3-64
maxpool $2 \times 2$	maxpool $2 \times 2$
conv3-128	conv3-128
conv3-128	conv3-128
maxpool $2 \times 2$	maxpool $2 \times 2$
conv3-256	conv3-256
conv3-256	conv3-256
conv3-256	conv3-256
conv3-256	conv3-256
maxpool $2 \times 2$	maxpool $2 \times 2$
conv3-512	Fully-connected layer-512
conv3-512	Dropout-0.5
conv3-512	Fully-connected layer-512
conv3-512	Dropout-0.5
maxpool $2 \times 2$	Fully-connected layer-12 (softmax)
conv3-512	(BN added except for the last FC layer)
conv3-512	
conv3-512	
maxpool $2 \times 2$	
Fully-connected layer-4096	
Dropout-0.5	
Fully-connected layer-4096	
Dropout-0.5	
Fully-connected layer-1000 (softmax)	
# of parameters is 144,000,000	# of parameters is 3,128,908

# CNN Model structure



# Result

Base CNN (DeepYeast) performance on the subcellular localization dataset



The test accuracy is 0.8512 (vs 0.8671 in the paper).

- 1 Background
  - Protein subcellular localization
  - Data description
  - Implementation
- 2 A Convolutional Neural Network: DeepYeast
  - CNN model structure
  - Result
- 3 Residual Neural Network**
  - **ResNet model structures**
  - **Result**
- 4 Feature extraction and transfer learning
  - Definition
  - Result
- 5 Summary
  - Comparison of different methods
  - Discussion

# Motivation

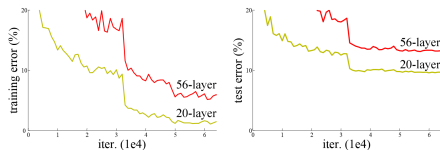


Figure 2: Poorer performance with deeper layers (He et al., 2016)

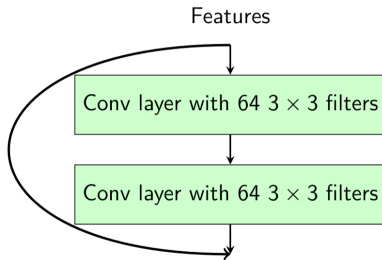


Figure 3: Convolution layer learns the residual features left by the identity skip connection/shortcut

# Residual neural networks

- Convolutional layer blocks; no fully-connected layers;
- Identity shortcut was shown to perform well.
- We tried 18- and 50-layered ResNet, Res18 and Res50.

# ResNetwork model structures

Table 3: Model structure

Block name	DeepYeast	Res18 (ours)	ResNet 50	Res50 (ours)	W40-4	W40-2
conv1_x	$[3 \times 3, 64] \times 2$	$[7 \times 7, 64]$	$[7 \times 7, 64]$	$[7 \times 7, 64]$	$[3 \times 3, 16]$	$[3 \times 3, 16]$
conv2_x	$[3 \times 3, 128] \times 2$	$[3 \times 3, 64] \times 2$ $[3 \times 3, 64]$	$[1 \times 1, 64]$ $[3 \times 3, 64] \times 3$ $[1 \times 1, 256]$	$[1 \times 1, 64]$ $[3 \times 3, 64] \times 3$ $[1 \times 1, 64]$	$[3 \times 3, 16 \times 4] \times 6$ $[3 \times 3, 16 \times 4]$	$[3 \times 3, 16 \times 2] \times 6$ $[3 \times 3, 16 \times 2]$
conv3_x	$[3 \times 3, 256] \times 4$	$[3 \times 3, 64] \times 2$ $[3 \times 3, 64]$	$[1 \times 1, 128]$ $[3 \times 3, 128] \times 4$ $[1 \times 1, 512]$	$[1 \times 1, 64]$ $[3 \times 3, 64] \times 2$ $[1 \times 1, 64]$	$[3 \times 3, 32 \times 4] \times 6$ $[3 \times 3, 32 \times 4]$	$[3 \times 3, 32 \times 2] \times 6$ $[3 \times 3, 32 \times 2]$
conv4_x		$[3 \times 3, 64] \times 2$ $[3 \times 3, 64]$	$[1 \times 1, 256]$ $[3 \times 3, 256] \times 6$ $[1 \times 1, 1024]$	$[1 \times 1, 64]$ $[3 \times 3, 64] \times 2$ $[1 \times 1, 64]$	$[3 \times 3, 64 \times 4] \times 6$ $[3 \times 3, 64 \times 4]$	$[3 \times 3, 64 \times 2] \times 6$ $[3 \times 3, 64 \times 2]$
conv5_x		$[3 \times 3, 64] \times 2$ $[3 \times 3, 64]$	$[1 \times 1, 512]$ $[3 \times 3, 512] \times 3$ $[1 \times 1, 2048]$	$[1 \times 1, 64]$ $[3 \times 3, 64] \times 3$ $[1 \times 1, 64]$		
	max pooling [512-d fc] $\times$ 2 12-d fc (softmax)			average pooling, 12-d fc (softmax)		



# Test accuracy of residual neural networks

- Res18 and Res50 performed better than their plain versions;
- Plain 50 worse than plain 18; but Res50 better than Res18;
- More benefits with 50 layers.

Table 4: Comparison of accuracy among different methods

Network	Training time	Test accuracy
plain 18	1.75 h	0.8432
Res 18	1.75 h	0.8708
plain 50	13 h	0.8190
Res 50	12.75 h	0.8856

- 1 Background
  - Protein subcellular localization
  - Data description
  - Implementation
- 2 A Convolutional Neural Network: DeepYeast
  - CNN model structure
  - Result
- 3 Residual Neural Network
  - ResNet model structures
  - Result
- 4 Feature extraction and transfer learning**
  - Definition**
  - Result**
- 5 Summary
  - Comparison of different methods
  - Discussion

# Definition and advantages

- The last one or few layers of a pretrained neural network are replaced by new classifiers.
- More accurate and faster (vs. without feature extraction).

## Use trained network as a feature extractor

- Replace the last fully-connected layer of the base CNN model (DeepYeast) with a random forest and an XGBoost:
  - Compared to using vectorizing-image input, the test accuracy is improved (0.85 vs 0.6)
  - Faster: with 512 extracted features vs the original 12288 (=  $64 \times 64 \times 3$ ) features
- Replace all the fully-connected layers of the VGG-19 model with new fully-connected layers, a random forest and an XGBoost respectively,
  - Very quick compared to training a neural network from scratch; decent test accuracy (0.73, 0.66 , 0.72)

- 1 Background
  - Protein subcellular localization
  - Data description
  - Implementation
- 2 A Convolutional Neural Network: DeepYeast
  - CNN model structure
  - Result
- 3 Residual Neural Network
  - ResNet model structures
  - Result
- 4 Feature extraction and transfer learning
  - Definition
  - Result
- 5 Summary
  - Comparison of different methods
  - Discussion

# Summary

Table 5: Comparison of accuracy between different methods

Network	Training time	Test accuracy
DeepYeast (11-layer CNN)	6 h	0.851
CNN (18-layer)	1.75 h	0.843
Res18	1.75 h	0.871 – 0.891
ResNet 18 (He et al., 2016)	2.45 h	0.853
CNN (50-layer)	13 h	0.819
ResNet 50 (He et al.2016)	12.75 h	0.886
Wide ResNet (widening factor 2)	46 h	0.853
Random Forest (v-images; 1000 trees)	1.68 h	0.600
XGBoost (v-images 800 trees)	10 h	0.6
Feature extraction by DeepYeast:		
Random Forest	10 min	0.850
XGBoost	1 h	0.840
Feature extraction by VGG-19 (transfer learning):		
FC layers	3 min	0.730
Random Forest (800 trees)	12 min	0.660
XGBoost (1000 trees)	14 h	0.722

# Discussion

- CNNs performed best, though not a thorough evaluation!  
Why? images, a large dataset, ...
- Other statistical methods: RF, Boosting, SVM good, but not tailored to images...  
BUT, some new stat methods: could not even run ...

# Acknowledgment

- Funded by NIH, NSF.
- **Thank you!**



## References

- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Pan, Y., Mai, Q., and Zhang, X. (2018a). *catch: Covariate-Adjusted Tensor Classification in High-Dimensions*. R package version 1.0.
- Pan, Y., Mai, Q., and Zhang, X. (2018b). Covariate-adjusted tensor classification in high-dimensions. *arXiv preprint arXiv:1805.04421*.
- Pärnamaa, T. and Parts, L. (2017). Accurate classification of protein subcellular localization from high-throughput microscopy images using deep learning. *G3: Genes, Genomes, Genetics*, 7(5):1385–1392.