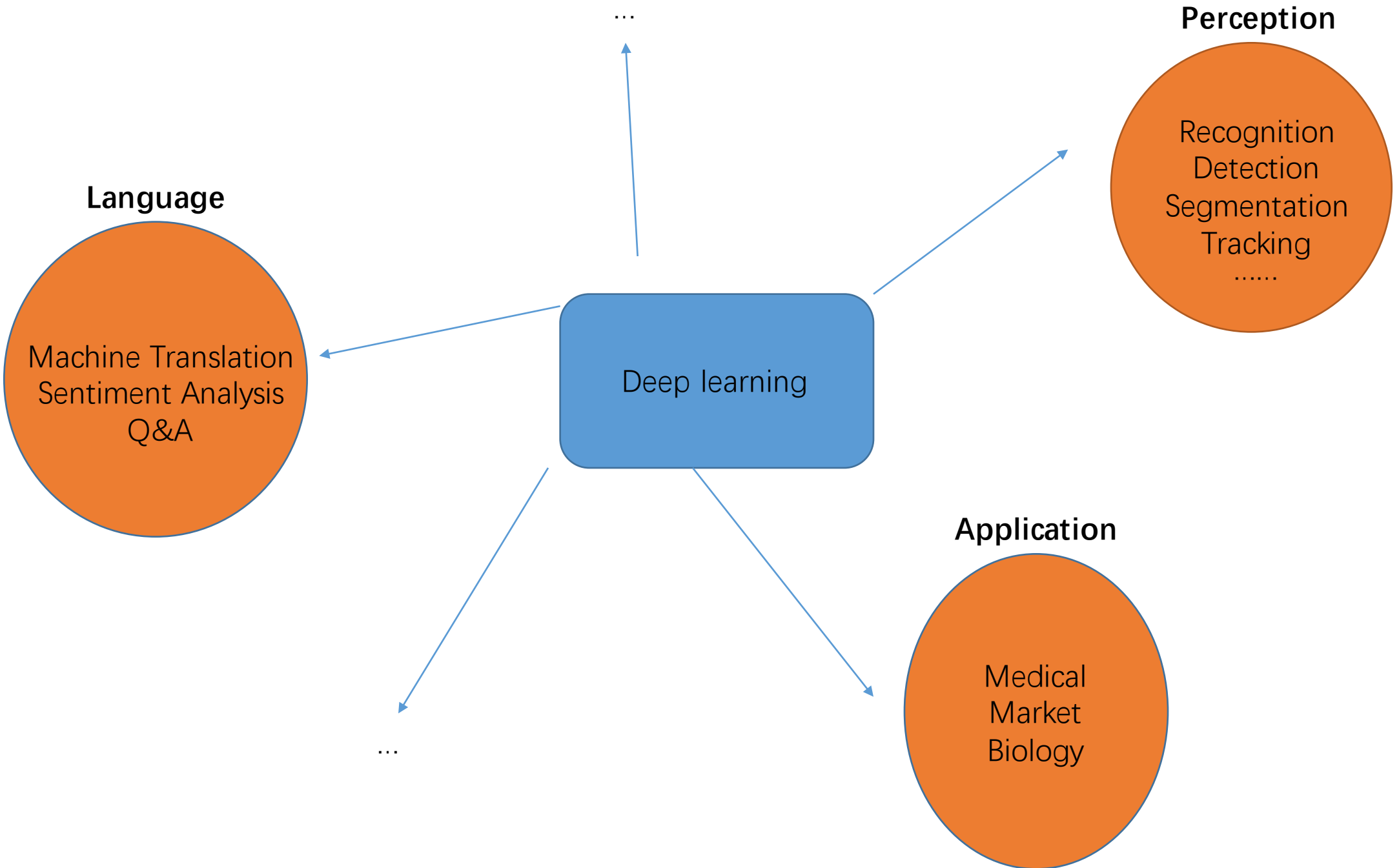
MATH6380O Project1

Mini experiment on MNIST

Shangzhe Wu, Xinnan Du, Gengmo Qi, Boyu Liu

Deep Neural Networks

- Popular
- Strong
- “Invade” and “conquer” multiple areas


...
Perception
Recognition
Detection
Segmentation
Tracking
.....
Language
Machine Translation
Sentiment Analysis
Q&A
Deep learning
...
Application
Medical
Market
Biology

Why? What happened? Alchemy（炼金）？

- Still, just a non-linear function
- Y = f(X)
- But, with good properties (like scarcity)
- Of course, BIG data & powerful GPUs

And where goes the classical ML techs?

Classical framework

"An honorable Professor"

Where lots of smart ideas appeared

Feature Engineering;
Dimension Reduction;
Analysis;
...

Modeling → **Output**

Where classical ML models come in

Deep learning?

Where lots of smart ideas appeared

Feature Engineering;
Dimension Reduction;
Analysis;
...

Modeling → Output

Where classical ML models come in

Neural Networks!!! → Output

- **Simple!** （One stage）
- **Simple!** (No need to design features)
- **Powerful!** (model capacity)
- **Note:** Last generation of ML tech, Kernel SVM, is kind of having these properties

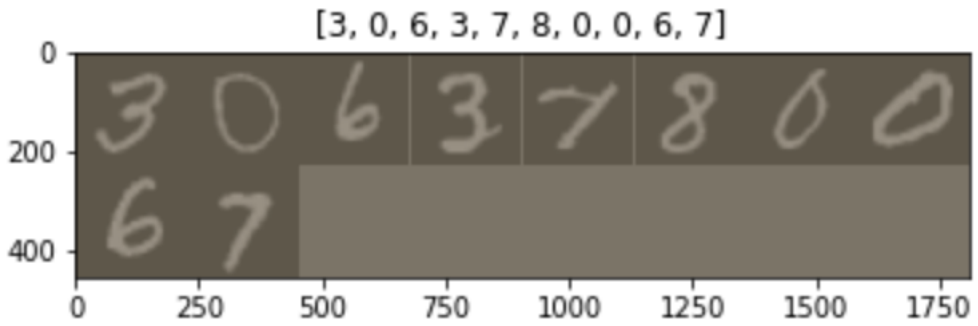
MNIST experiments on this “SSP” tech

- ResNet18 trained on ImageNet

- 1. Feature Extractor (vs ScatNet)

- 2. Modeling (vs Classical ML techs)

MNIST dataset recap



1. Feature extractor

- ResNet18 (trained on imagenet)

versus

- ScatNet (human-defined filters)

Quantitative result (accuracy using multiple models)

- ScatNet:

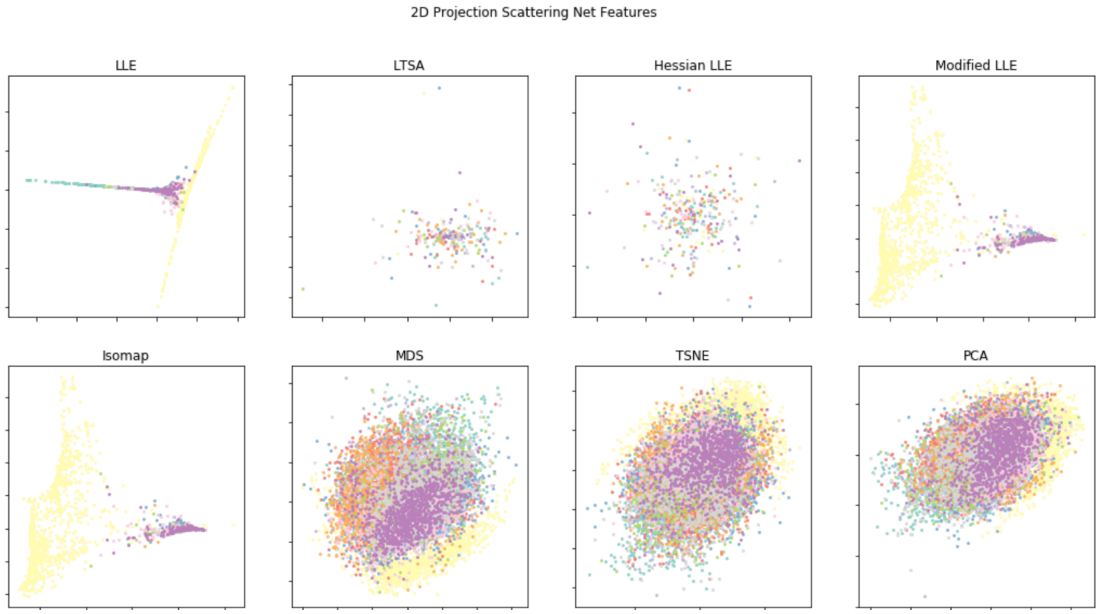
	Average Fitting Time	Train Accuracy	Average Test Time	Test Accuracy
LDA	3.67757e-05	0.9583	1.10083e-06	0.958
LGR	0.000686586	0.918967	1.00079e-06	0.9239
SVM	0.0495049	0.727	0.0201156	0.7451
RFC	0.000264731	0.998517	7.40516e-06	0.9071

- ResNet18:

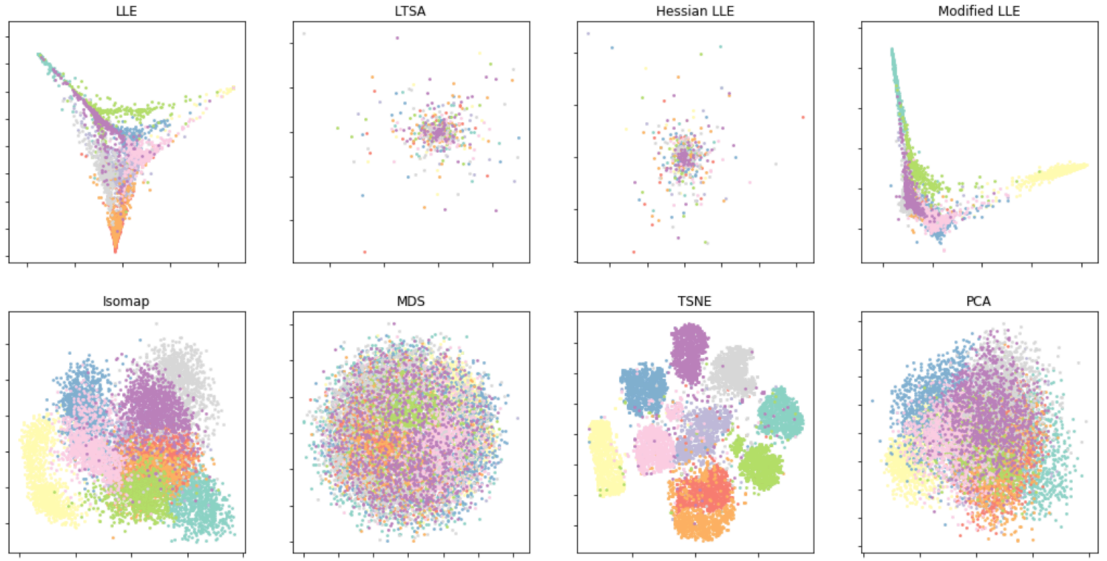
	Average Fitting Time	Train Accuracy	Average Test Time	Test Accuracy
LDA	2.90342e-05	0.977167	1.3603e-06	0.9746
LGR	0.00161714	0.991933	1.36828e-06	0.982
SVM	0.00924093	0.98325	0.00590839	0.9799
RFC	0.000200609	0.99905	2.67529e-06	0.9137

Qualitative result (visualization)

ScatNet

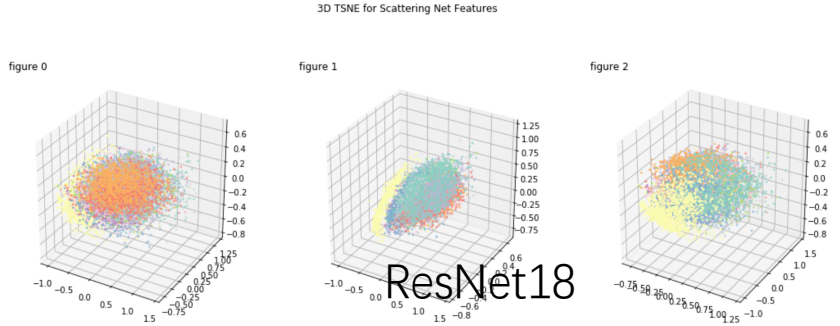


ResNet18

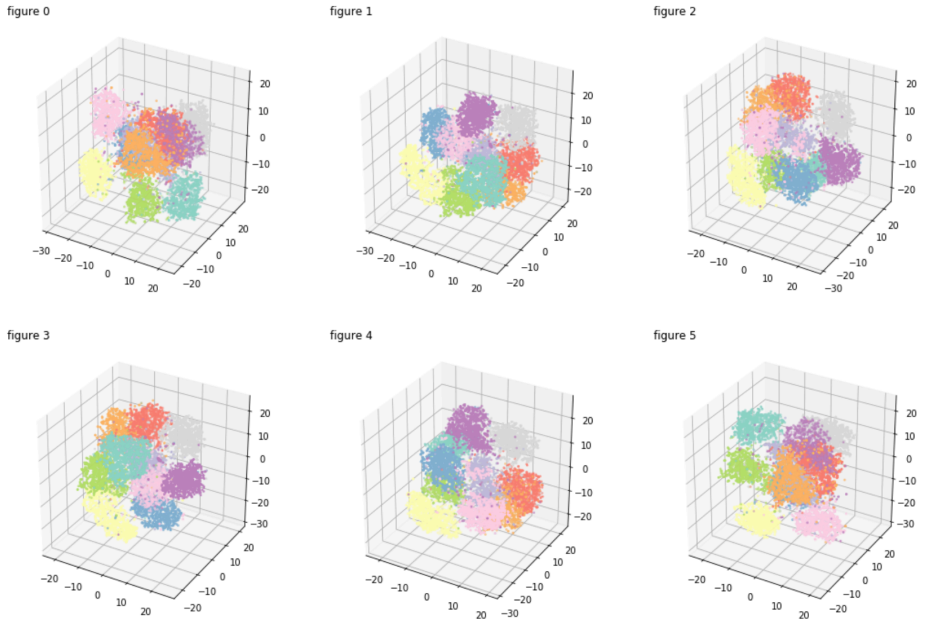


TSNE

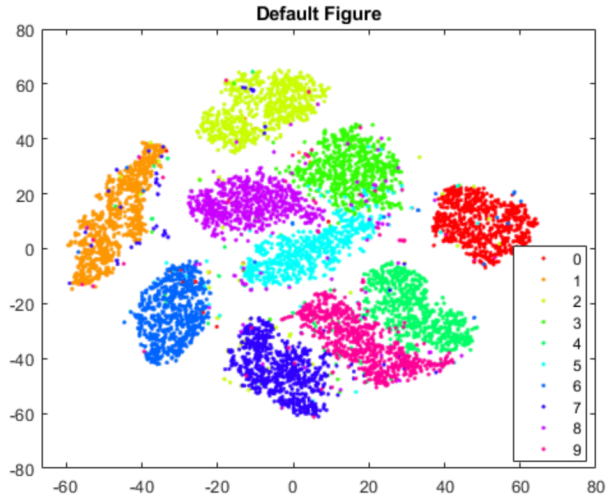
ScatNet



ResNet18



Direct PCA+TSNE on image



Winner goes to…

ResNet18!

- Amazingly, never seen MNIST, the features are still very good. It seems that deep learning really have its power in extracting features. And these features could generalize well to another dataset.

Where lots of smart ideas appeared

Feature Engineering;
Dimension Reduction;
Analysis;
...

Modeling → Output

Where classical ML models come in

Neural Networks!!! → Output

2. Modeling

- ResNet18 (In this context, we refers to using fine-tuned ResNet18 to inference end-to-end)

Versus

- Classical ML (Here we use ResNet18 features as the input)

Results

- Fine-tuned ResNet18: 99.6% on test set after 5 epoch
- Classical ML:

Quantitative result (accuracy using multiple models)

- ScatNet:

	Average Fitting Time	Train Accuracy	Average Test Time	Test Accuracy
LDA	3.67757e-05	0.9583	1.10083e-06	0.958
LGR	0.000686586	0.918967	1.00079e-06	0.9239
SVM	0.0495049	0.727	0.0201156	0.7451
RFC	0.000264731	0.998517	7.40516e-06	0.9071

This one?
No! Not fair!

- ResNet18:

	Average Fitting Time	Train Accuracy	Average Test Time	Test Accuracy
LDA	2.90342e-05	0.977167	1.3603e-06	0.9746
LGR	0.00161714	0.991933	1.36828e-06	0.982
SVM	0.00924093	0.98325	0.00590839	0.9799
RFC	0.000200609	0.99905	2.67529e-06	0.9137

- Use fine-tuned resnet18 to extract features again

	Average Fitting Time	Train Accuracy	Average Test Time	Test Accuracy
LDA	5.34711e-05	0.999367	4.60317e-06	0.9957
LGR	0.00102568	1	4.90353e-06	0.9963
SVM	0.00135654	0.999717	0.000985597	0.9965
RFC	0.000314965	0.999983	4.60334e-06	0.9941

- Fine-tuned ResNet18: 99.6% on test set after 5 epoch

- Comparable!!

- Surprise?
- Or not?

- After all, ResNet18 only use 1 fully-connected layer to perform classification.

- **Classical ML>ResNet18:** SVMs obviously have stronger power than 1 fully-connected layer

- **ResNet18> Classical ML**: features are adapted to the 1fully-connected layer classifier during training.

So, why in practice, not using CNN to extract features and SVM to do classification?

Neural Networks(CNN)

Features

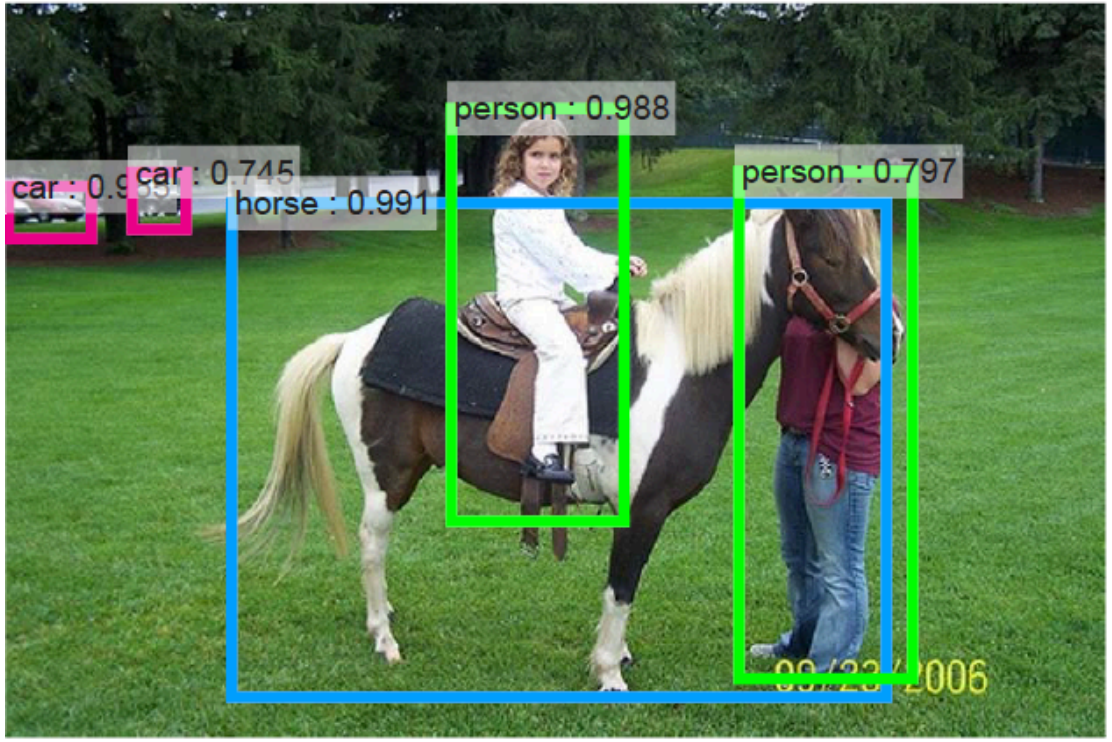
SVM

"An honorable Professor"

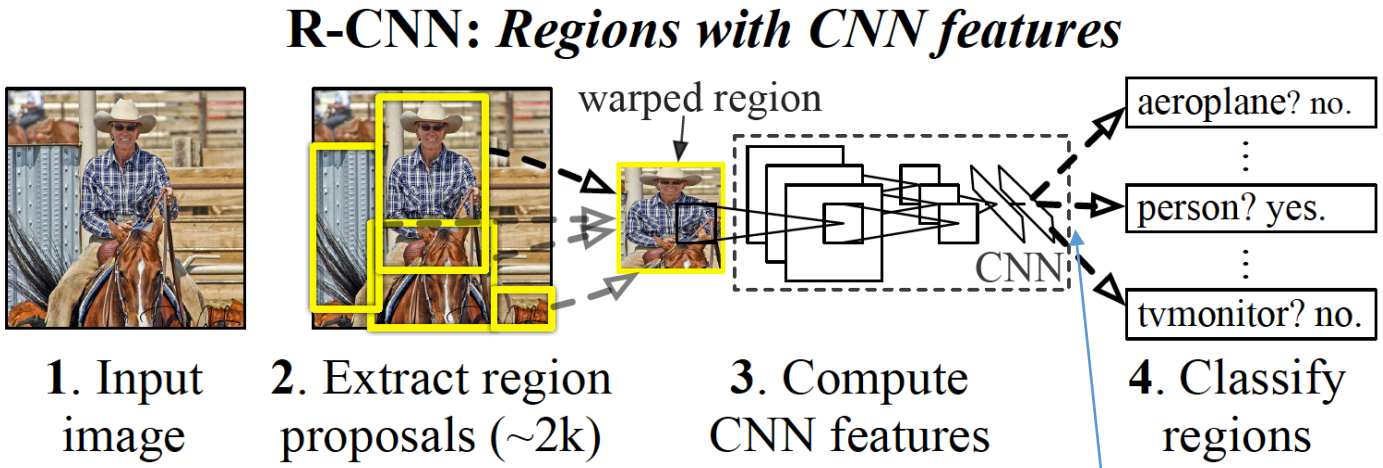
Let’s take a look at famous RCNN family

- RCNN(2014), Fast-RCNN(2015), Faster-RCNN(2016)

- Detection and classification task



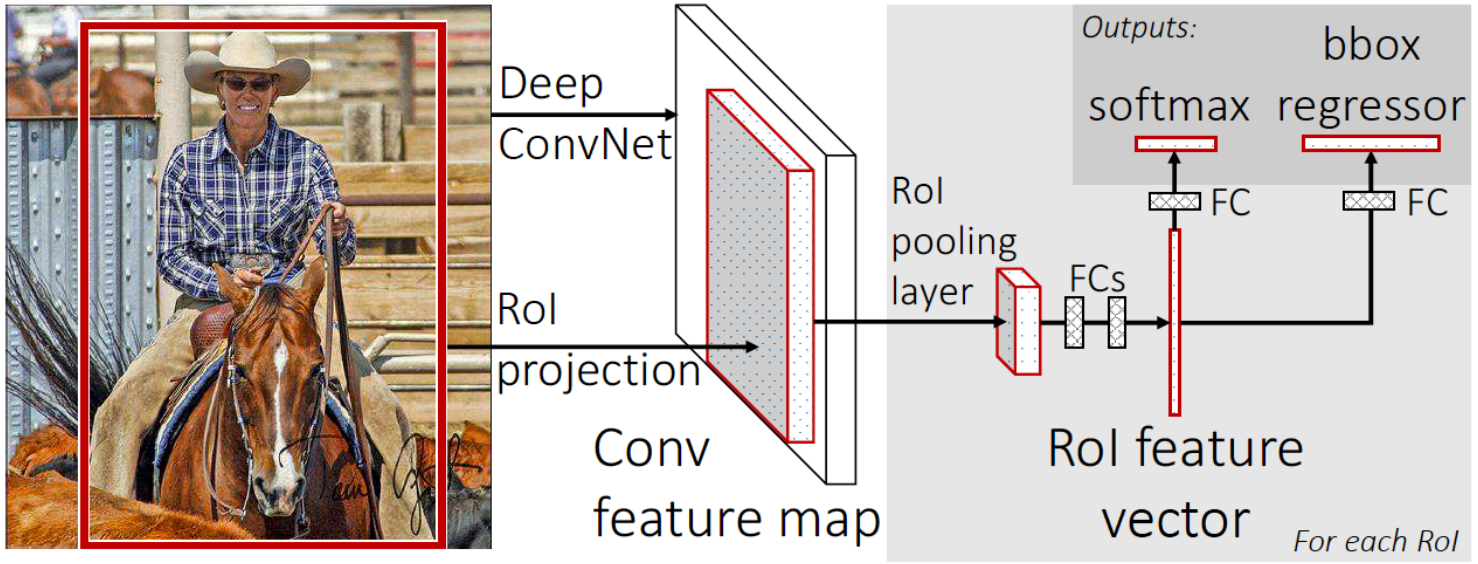
RCNN(2014)



Here it use SVM to do classification

Just like we suggested!!!

Fast-RCNN (2015)



SVM is abandoned!! And It only use FC(fully-connected) layer to perform classification

WHY?

Fast-RCNN says…

- 1. In RCNN version, SVM and feature extractor (CNNs) are trained separately, which makes training slower.

- 2. SVM is much slower than FC layer (just matrix multiplication) during inference.

- 3. FC layer(with softmax) outperform SVM by a very small margin.

method	classifier	S	M	L
R-CNN [9, 10]	SVM	**58.5**	**60.2**	66.0
FRCN [ours]	SVM	56.3	58.7	66.8
FRCN [ours]	softmax	57.1	59.2	**66.9**

Table 8. Fast R-CNN with softmax vs. SVM (VOC07 mAP).

We believe the following reasons also contributes to the choice

- 1. Currently, there are not well-designed algorithms to train classical ML techs(like SVM) together with neural networks (end-to-end back-propagation)
- This lead to the result that features are not fully adapted to these ML techs
- 2. Neural networks, especially CNNs, have already enough power to extract features that are easily separable. Thus SVMs would not improve a lot than FC-layer.

Conclusion

- 1. ResNet18 outperform ScatNet as a feature extractor.
- 2. ResNet18 itself could achieve comparable results with classical ML techs by fine-tuning, even though the classifier is very simple.
- 3. With practical concerns and the fact that Neural Networks and Classical ML techs could not be trained together, Neural Networks is always used on its own


Q
&
A

Thanks!

- Shangzhe Wu
- Xinnan Du
- Gengmo Qi
- Boyu Liu

- ***Group 09 LiuQiDuWu***