Identify BP segmentation in ultrasound images based on u-net architecture neural network

BP: brachial plexus

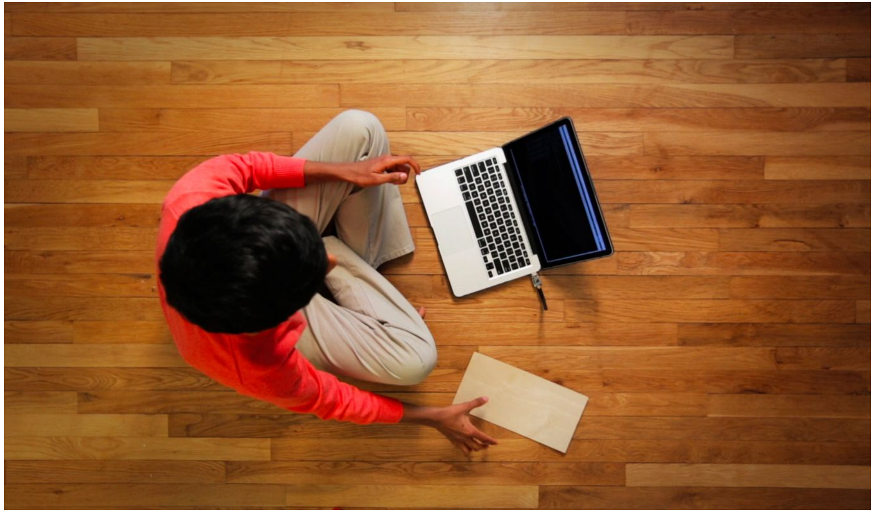
Boya Zhou, Zhiyuan Liu, You Zhou, Zhenyu Wan

Department of Computer Science, Department of Robotics

Reference

[1] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS. pp. 1106–1114 (2012)
[2] LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Computation 1(4), 541–551 (1989)
[3] Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Heidelberg (2015).
[4] Jonathan Masci, Ueli Meier, Dan Cire¸san, and J¨urgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In Proceedings of the 21th international conference on Artificial neural networks - Volume Part I, ICANN'11, pages 52–59, Berlin, Heidelberg, 2011. Springer-Verlag.
[5] Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
[6] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Y. Ng, "Large scale distributed deep networks," in Advances in Neural Information Processing Systems. MIT Press, 2012, pp. 1232–1240
[7] Ivan Bruha: Pre- and Post-processing in Machine Learning and Data Mining. In: Machine Learning and Its Applications, Advanced Lectures. Pp. 258-266 Jan.(2001)

Motivation



Many diseases $\xrightarrow{\text{precise surgical procedures}}$ cured

- **Patients**: cringes at the mention of such a process.
- **Doctors**:

to manage the pain $\longrightarrow$ using narcotics $\longrightarrow$ unwanted side effects

- **Result**:

Pain management is of great importance in the advanced research.

Intro & Background

Kaggle Competition Sponsor:

Use indwelling catheters $\xrightarrow{\text{block/mitigate pain}}$ (1) reduce dependence on narcotics

(2) speed up patient recovery

What we need to do is identifying the BP segmentations based on ultrasound images and improving the placement of catheters.

The problem and challenge

- Accurately and precisely identifying nerve structures in ultrasound images to high-effectively insert a patient's pain management catheter.

- We are challenged to build a model that can identify nerve structures accurately through ultrasound images.

Method

Each scientist uses different
methods of experimentation

Methods & Workflow

Our methods generally contain four parts:

Data preprocessing, Modeling, Data post processing and Evaluation

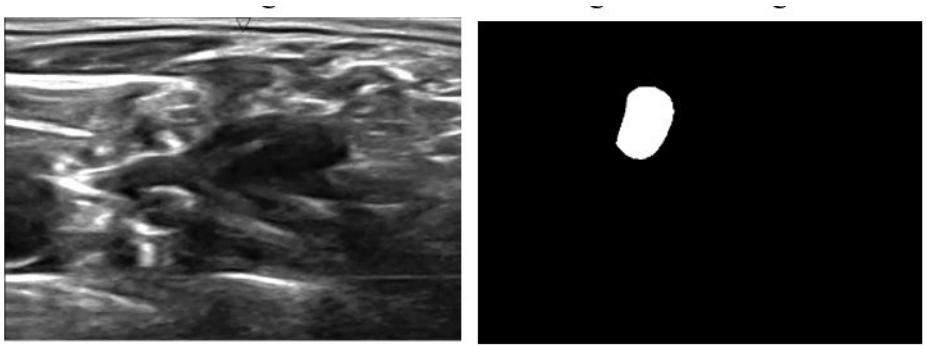
The major workflow is as below:

Data preparation, Standardization, Delete conflicts, Data augmentation, Deep convolutional network, Data post processing and Test.

Description of data

- Training data: contains the training set images.
- Test data: contains the test set images.

Based on nerve ultrasound image, detect BP segmentation and predict the mask.



A) example nerve ultrasound image B) Corresponding BP segmentation

Figure 1. Example of ultrasound nerve images

Data preprocessing---Standardization

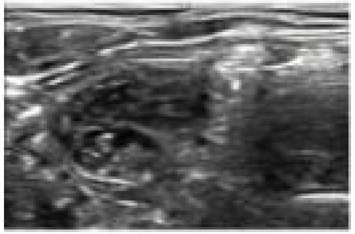
- **Zero score:**

  X=(X-mean) / std

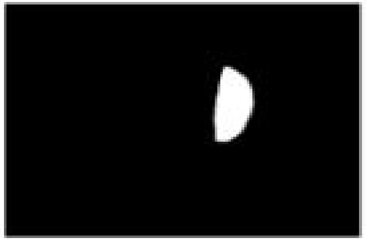
- We scale mask to [0,1] by dividing 255. Now the image size is 1x128x160

Data preprocessing---Delete conflicts

- There are many images “look” (by human or computer) very similar, but they have different masks.
- We use histogram of images to compare images. With the threshold we set, similar images with different result are discovered and we decide to remove images without BP segmentation.



A) ultrasound image 6_88.tif in dataset



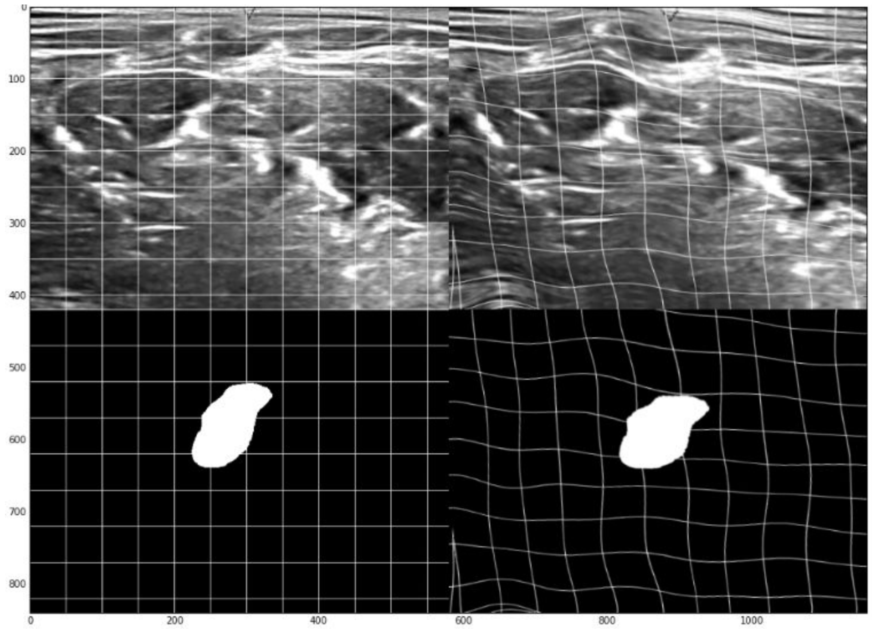
B) ultrasound mask image 6_88_mask.tif

A) image 6_101.tif

B) image 6_101_mask.tif

Data preprocessing---Augmentation

- To obtain more data, we need to implement geometric transformation on image data and add the new images into training data.
- The skills we have tried:
  - Flipping, random rotation, and elastic transformation.
- Elastic transformation improves the result and the main idea of ET is distortion.



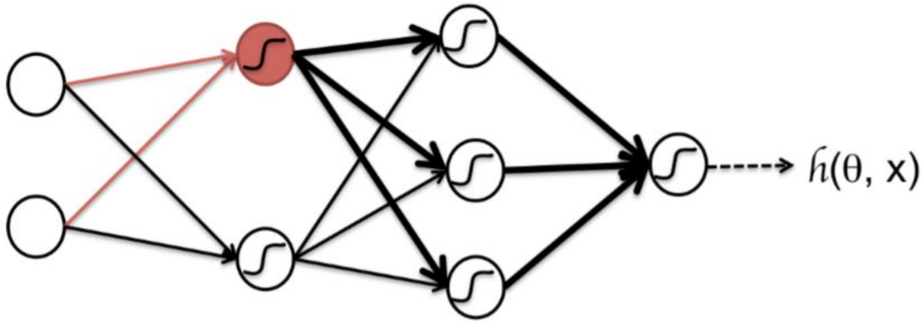
Evaluation

The evaluation is based on the mean Dice coefficient. Dice coefficient can be used to compare the pixel-wise agreement between a predicted segmentation and its corresponding ground truth. The formula is given by:

$$\frac{2 * |X \cap Y|}{|X| + |Y|}$$

Model Architecture

- CNN
  - image recognition
- U-net
  - image segmentation
- Our model
  - modified U-net



CNN

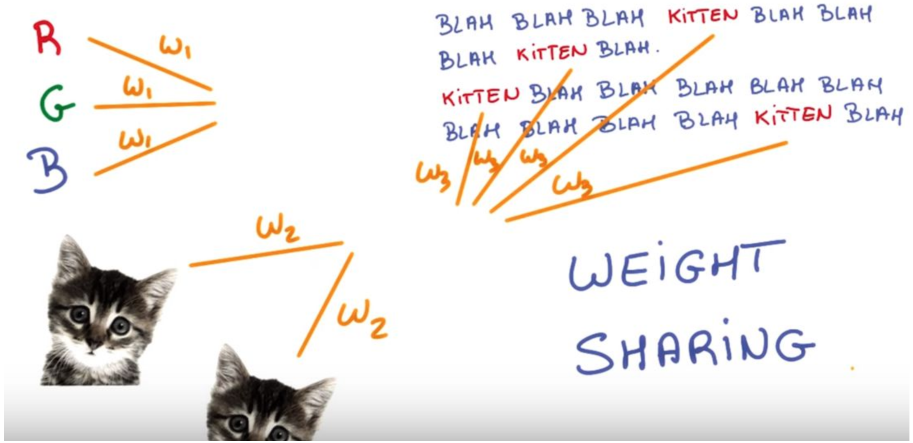
Remarkable improvement in object recognition for ImageNet, 2012

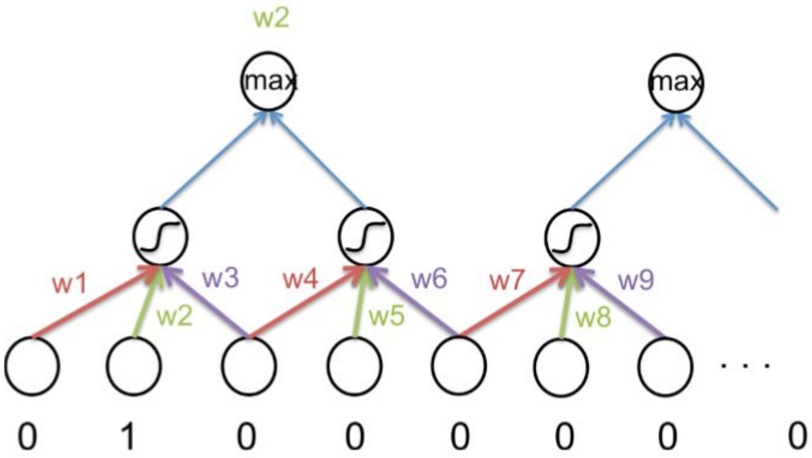
- local networks

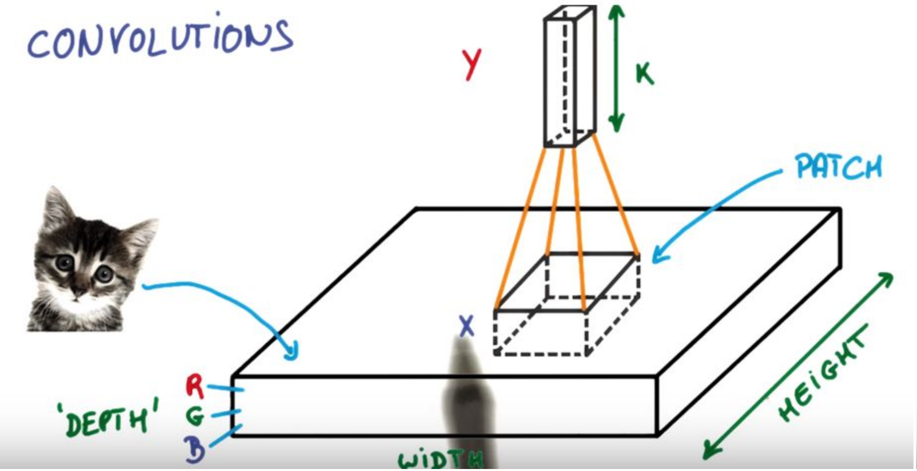
  deal with high dimensions

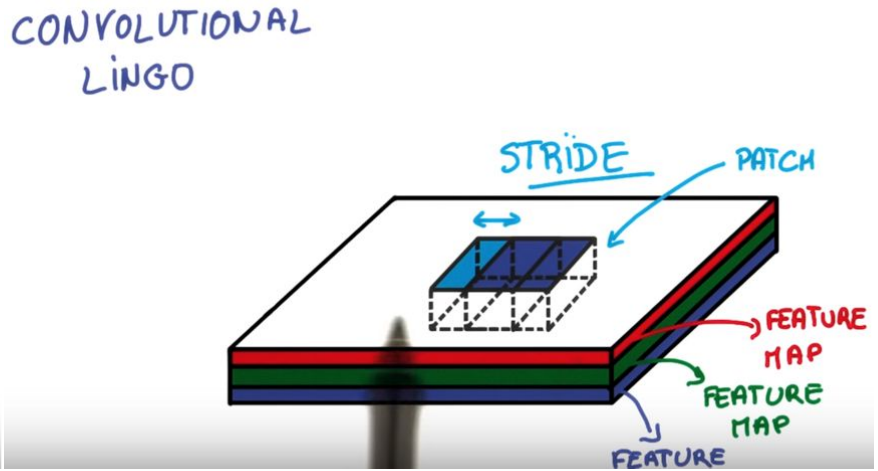
- weight sharing

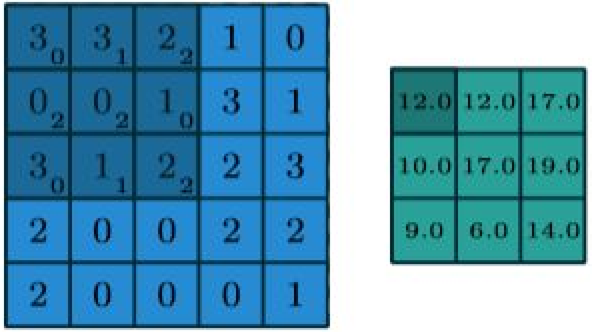
  translational invariance

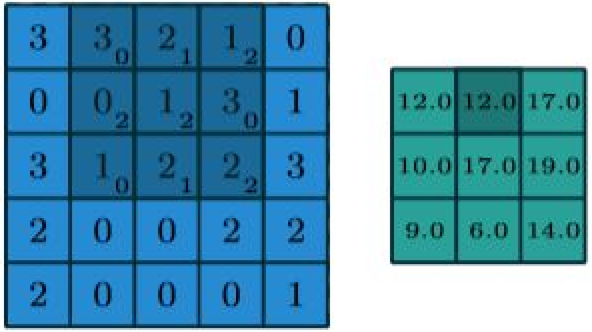





CONVOLUTIONS
Y
K
PATCH
X
R
G
B
'DEPTH'
WIDTH
HEIGHT

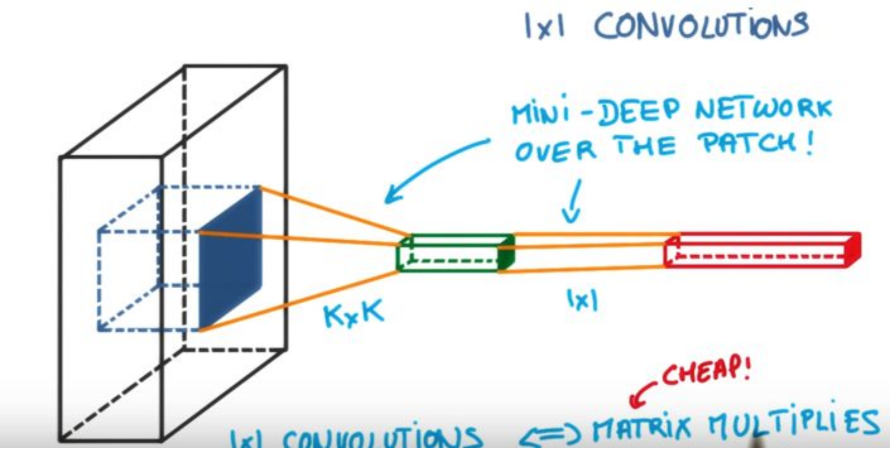

CONVOLUTIONAL
LINGO
STRIDE
PATCH
FEATURE
MAP
FEATURE
MAP
FEATURE


3 3 2 1 0
0 0 1 3 1
3 1 2 2 3
2 0 0 2 2
2 0 0 0 1
12.0 12.0 17.0
10.0 17.0 19.0
9.0 6.0 14.0


3 3 2 1 0
0 0 1 3 1
3 1 2 2 3
2 0 0 2 2
2 0 0 0 1
12.0 12.0 17.0
10.0 17.0 19.0
9.0 6.0 14.0

U-net: image segmentation

Outputs are images: deconvolution



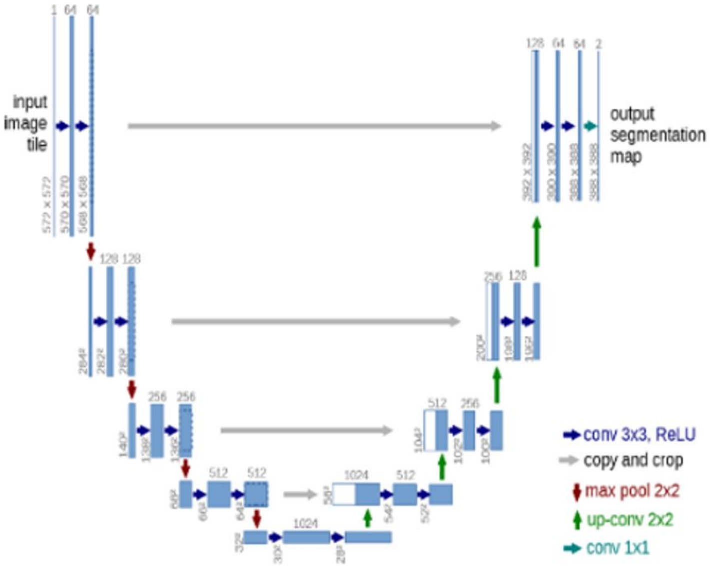
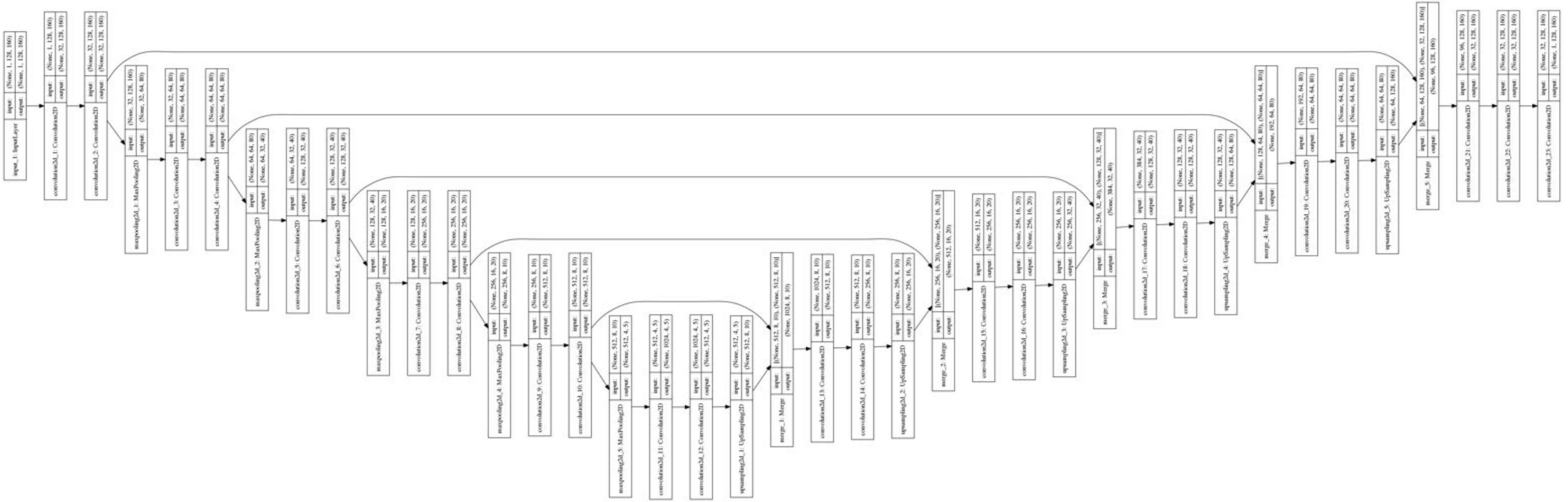
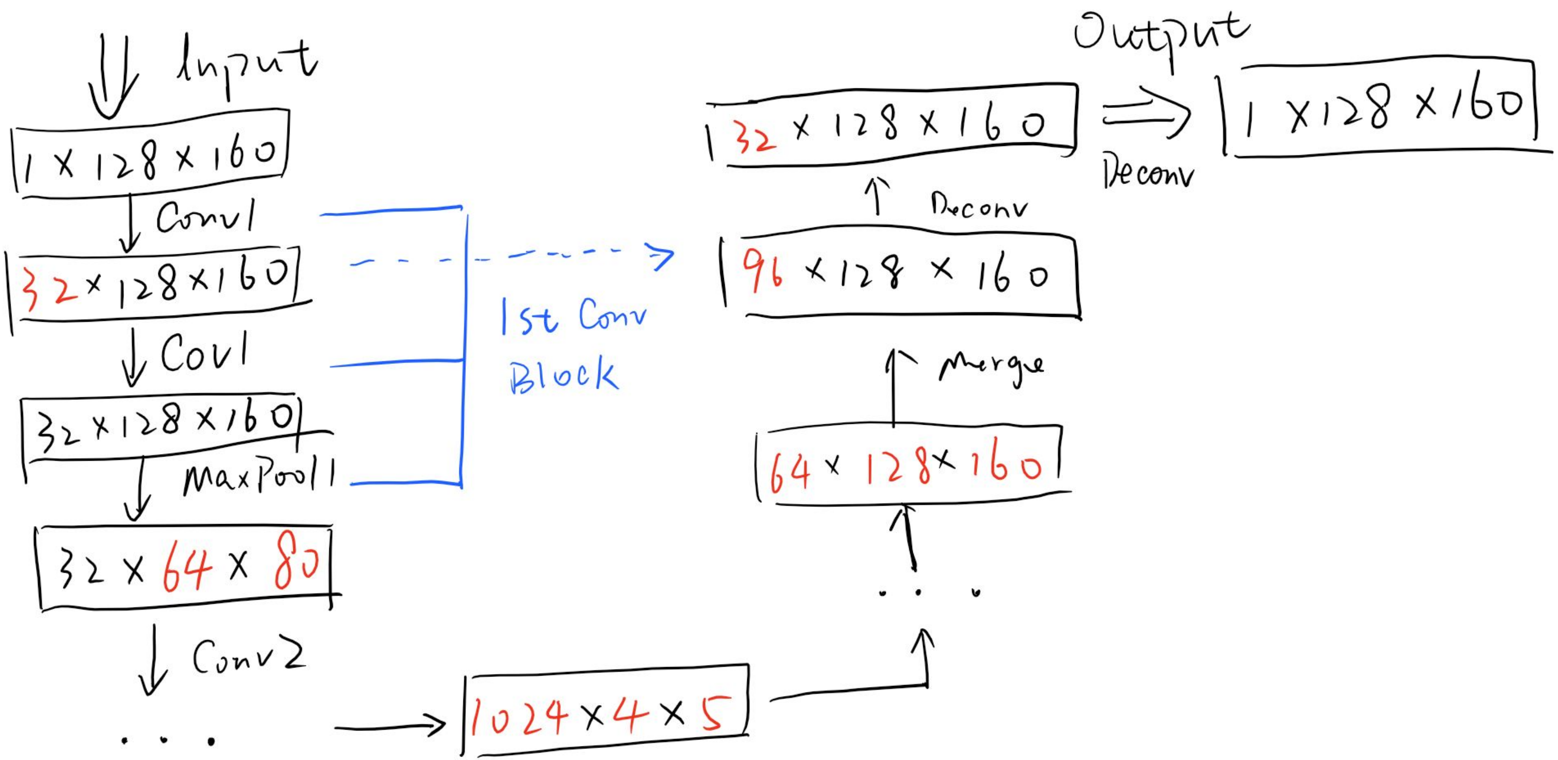


Figure 3. Example of U-net

Our Model Architecture



Our Model Architecture



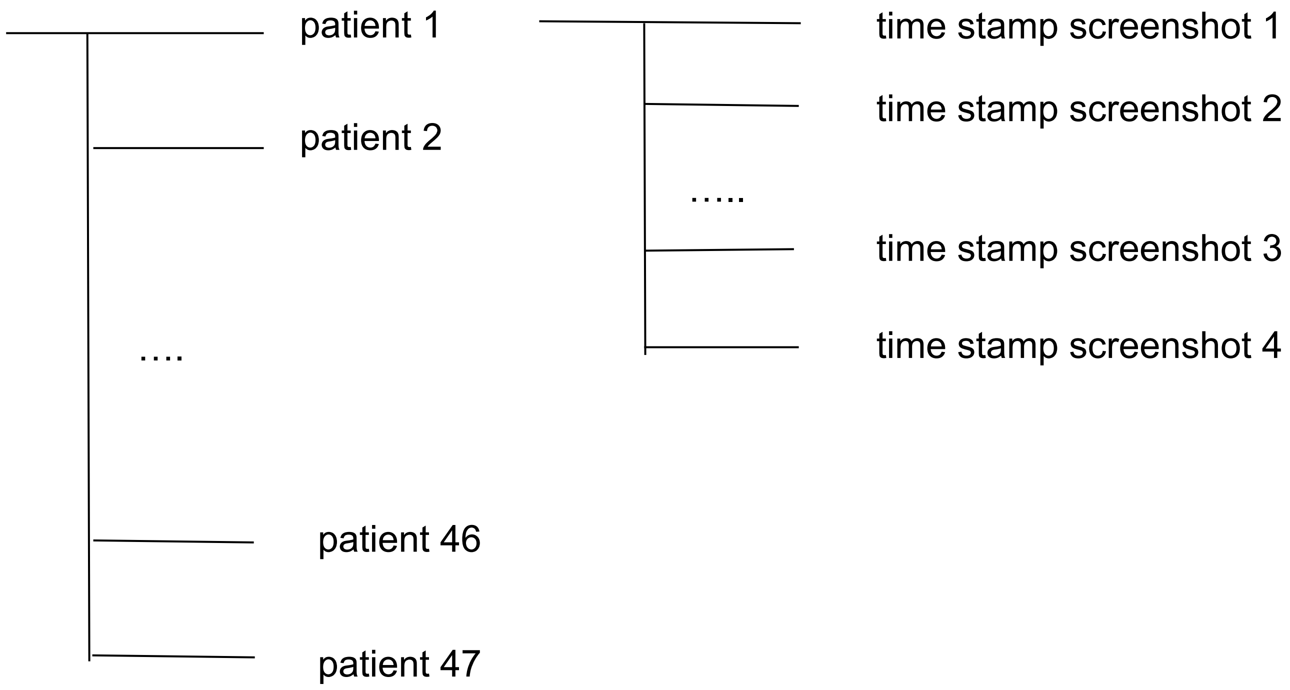
Result

Training detail and
comparing result

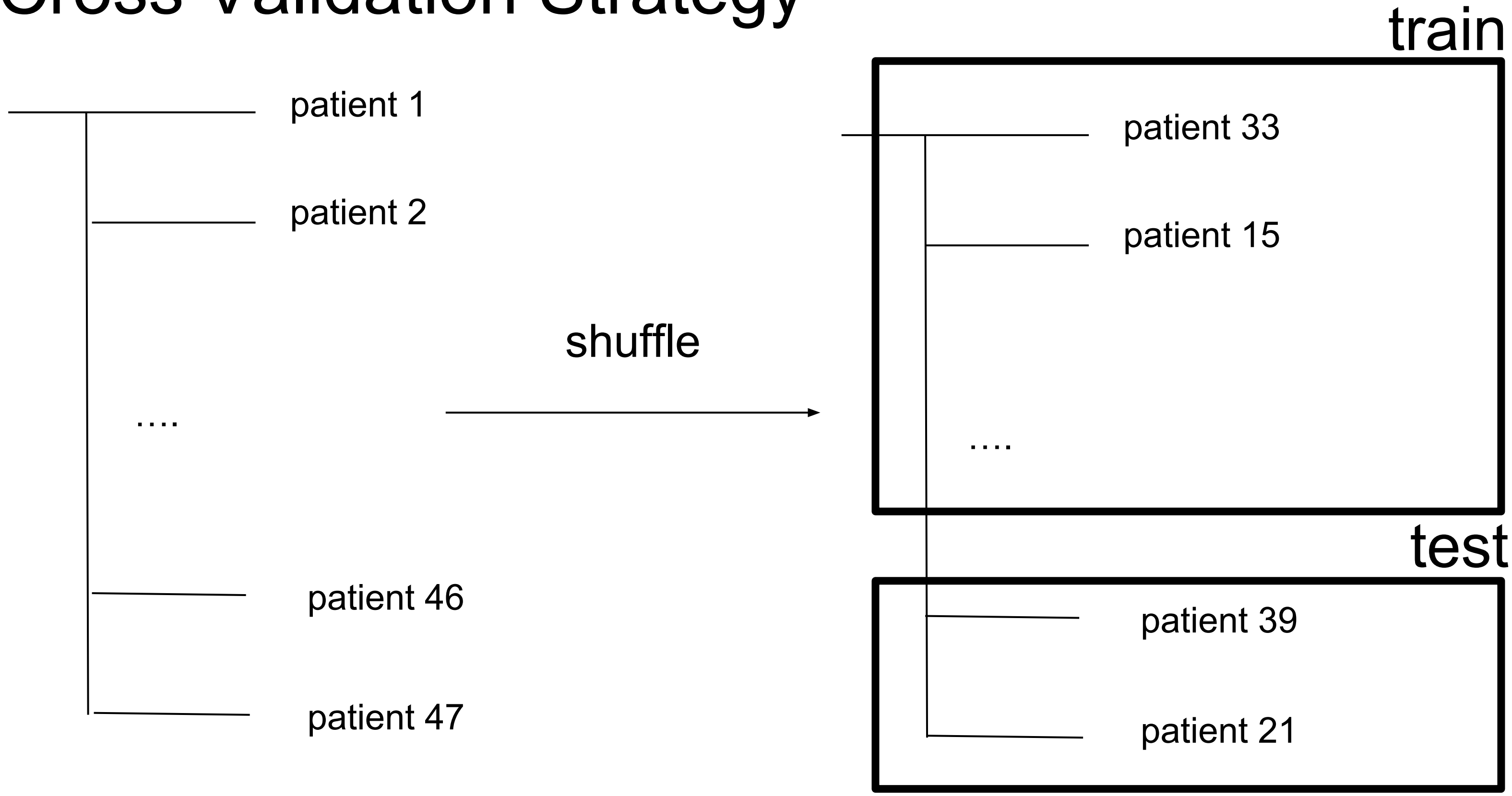
Where do we get evaluation results

- cross validation results
  - training : 37 patients (4508 images)
  - testing : 10 patients (450 images)
- private leaderboard result from kaggle (less intuitive but much more authentic and accurate)
  - no train-test split by ourself

Cross Validation Strategy: unique file structure



Cross Validation Strategy



Results: from Kaggle private leaderboard

	Statements	dice coefficient
1	Baseline	0.53449
2	Simple convolutional neural network no merge	0.56337
3	CNN random augmentation	0.53449
4	u-Net ConvNet	0.57646
5	Modified u-Net ConvNet	0.59124
6	Set loss function to cross entropy	0.57543
7	(5) with data augmentation	0.61191
8	(7) with removing conflict images	0.62574
9	(8) with post-processing technique	0.62741

Table 1. Comparatives of different models' results on ultrasound nerve problem

Results

	Statements	dice coefficient
1	Baseline	0.53449
2	Simple convolutional neural network no merge	0.56337
3	CNN random augmentation	0.53449
4	u-Net ConvNet	0.57646
5	Modified u-Net ConvNet	0.59124
6	Set loss function to cross entropy	0.57543
7	(5) with data augmentation	0.61191
8	(7) with removing conflict images	0.62574
9	(8) with post-processing technique	0.62741

Table 1. Comparatives of different models' results on ultrasound nerve problem

Training detail

The training is carried out by optimizing the logistic regression objective using mini-batch gradient descent based on back-propagation with no momentum.

These parameter are fixed for comparing method:

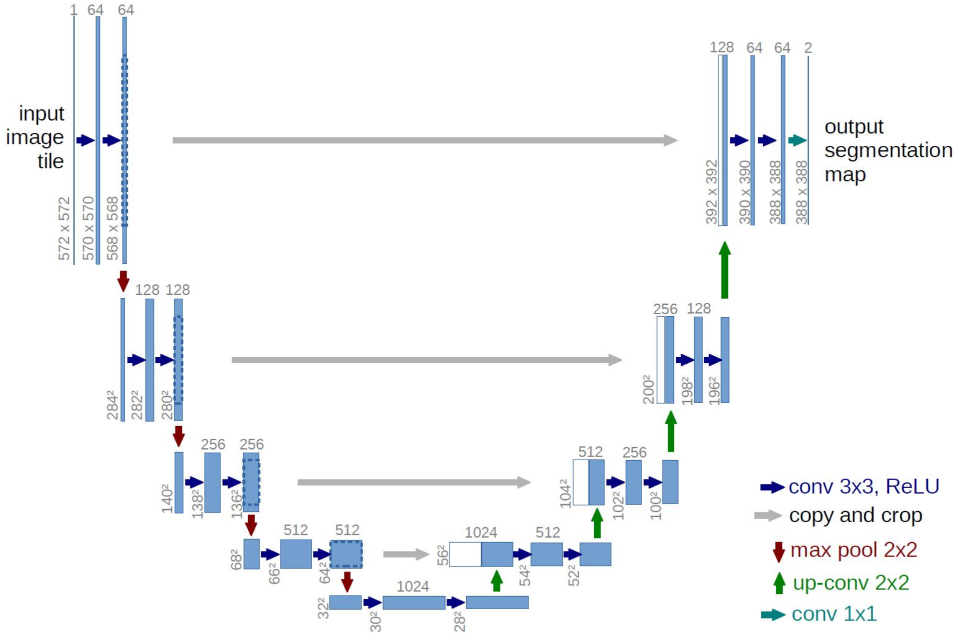
- Loss function: dice coefficient
- batch size :32
- learning rate : 0.00001
- epoch : 20

Results: Baseline

Set all test data to pure negative result
The result is 0.53449

The best final result on Kaggle is 0.73226

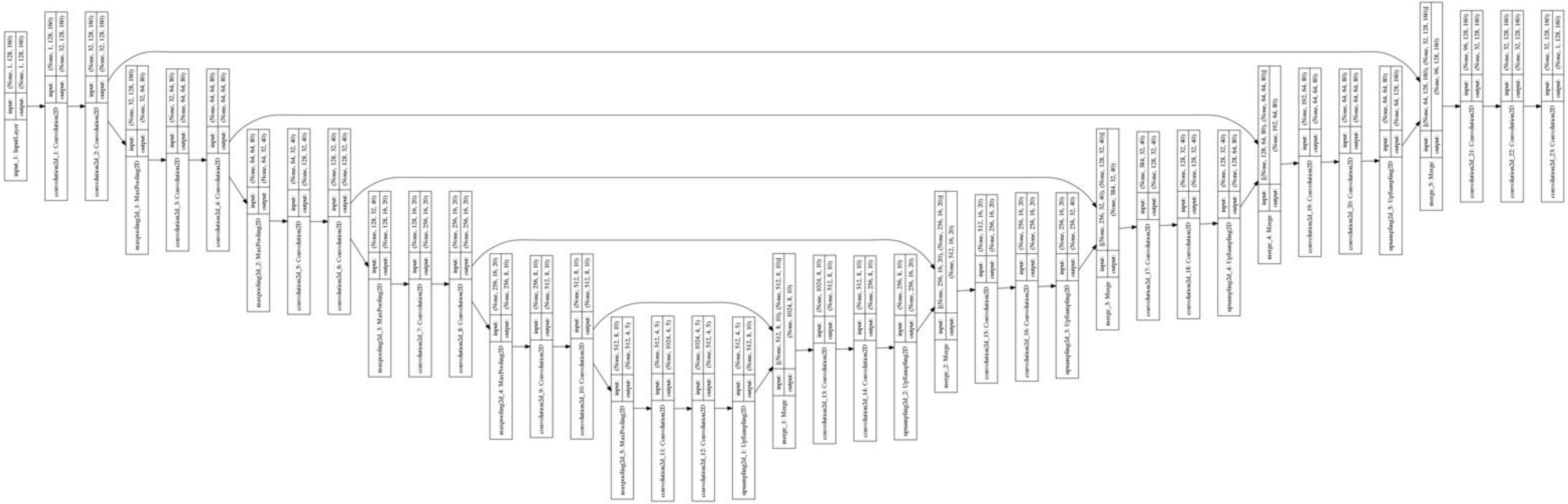
Results: U-net ConvNet



The settings are as follows:

- Image input resolution was resized from 580 * 420 to 80 * 64
- Result : improve from 0.53449 to 0.57646

Results: Modified u-Net ConvNet



The settings are as follows:

- Image input resolution was resized from 580 * 420 to 160 * 128
- one more deep layer, max channel from 512 to 1024
- Result : improve from 0.57 to 0.59124

Results: U-net with other processing technique

- Data augmentation
  - elastic transform on each image
  - double the training data
  - result increase from 0.59124 to 0.61191
- Delete conflict data
  - delete around 0.5% “outlier”
  - result increase from 0.61191 to 0.62574
- Post-processing
  - set test image with “small mask” to pure negative result
  - result increase from 0.62574 to 0.62741

Conclusion

- the U-net convolutional neural network is powerful in biomedical image segmentation problem
- Several data processing skills can help ConvNet get better performance

What will we do next?

What will you do with your findings next?

set high weight to border pixel

Thanks for all kagglers' help
And thanks for listening!