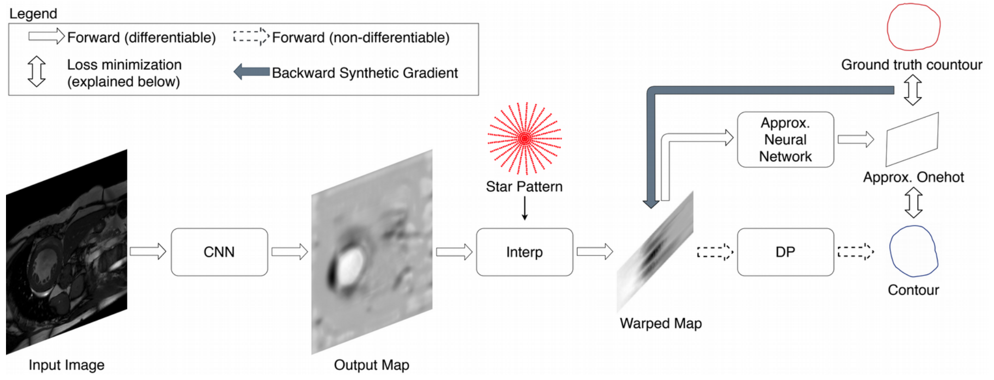
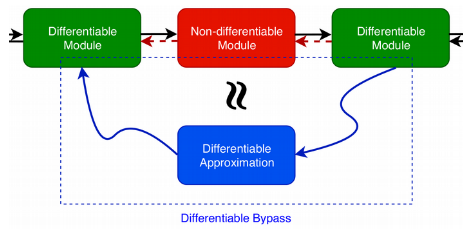
MIDL
Montréal 2020
6 - 9 July 2020

EDPCNN:
End-to-end learning of convolutional neural net and dynamic programming for left ventricle segmentation

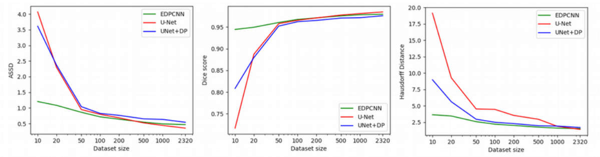
Nhat M. Nguyen & Nilanjan Ray
University of Alberta



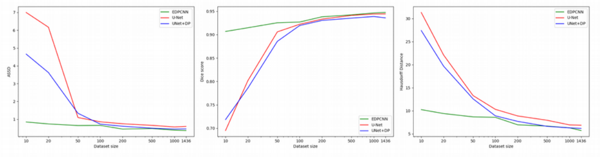


$$F(x) \approx G(x)$$

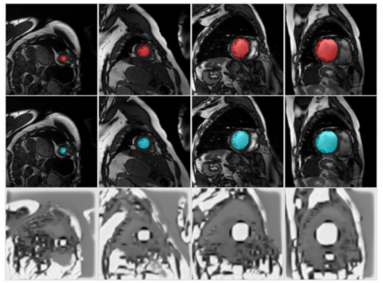
$$\nabla_x F \approx \nabla_x G$$



ACDC dataset

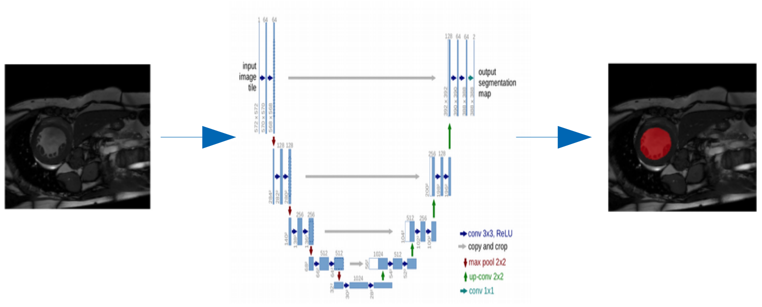


LVQuan2018 dataset



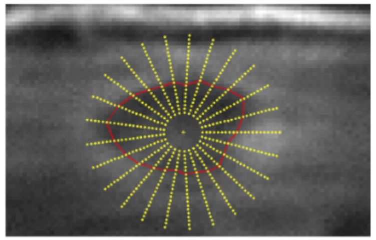
Motivation: Semantic segmentation

Deep Learning



(Example: U-Net [1])
- Powerful but need lots of data
- No trivival way to incoporate prior knowledge

Traditional Segmentation

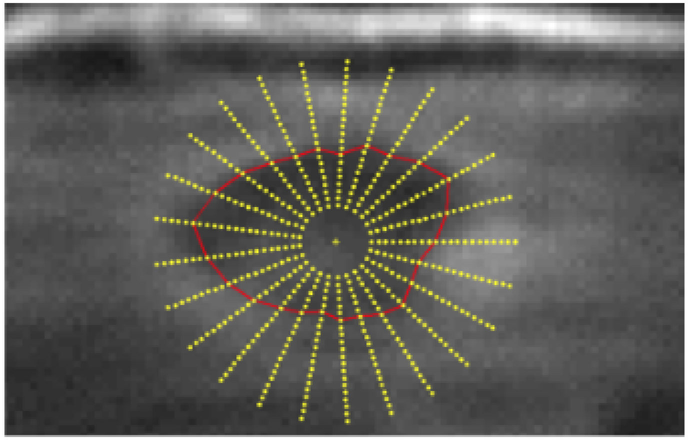


(Example: active countour)
- Handcrafted with prior knowledge
- Fixed performance

There’s a need to combine deep learning with traditional segmentation.

[1] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.

An example: Active Contour (Dynamic Programming)



- User provide a candidate position of the object
- Find the path that minimize sum of energy energy on a “Star Pattern” using Dynamic Programming

$$\min_{v_1,\ldots,v_N} E(N, v_N, v_1) + \sum_{n=1}^{N-1} E(n, v_n, v_{n+1})$$

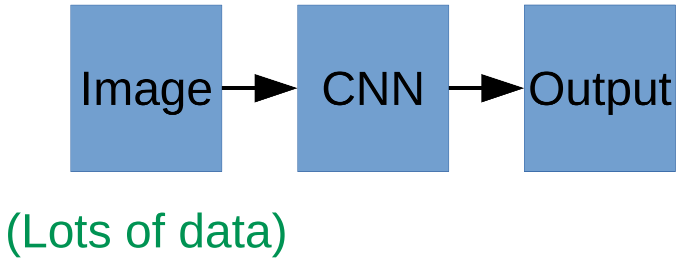
$$E(n,i,j) = \begin{cases} g(n,i) - g(n,i-1) + g(n \oplus 1, j) - g(n \oplus 1, j-1), & |i-j| \le \delta \\ \infty, & \text{otherwise,} \end{cases}$$

Energy function is the sum of the gradient between two consecutive lines.
Delta: smoothness constraint. Selected indices cannot change abruptly.

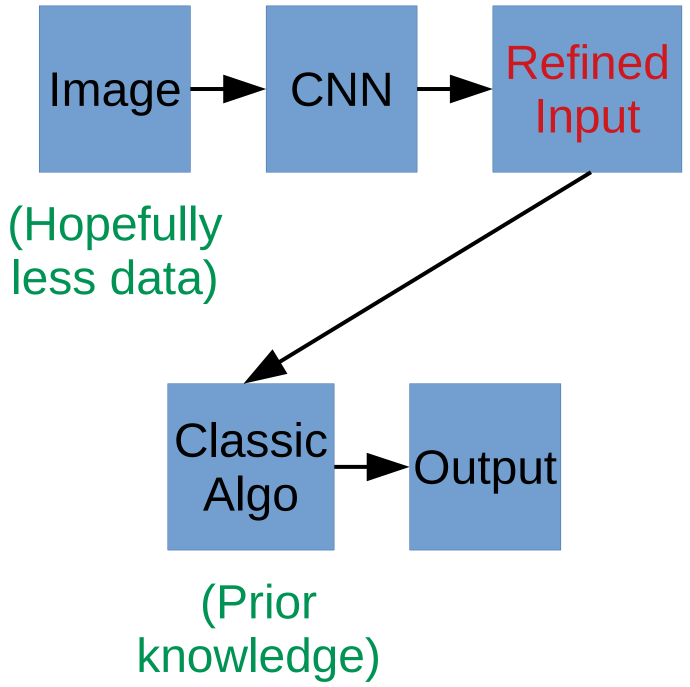
N. Ray, S. T. Acton, and H. Zhang. Seeing through clutter: Snake computation with dynamic programming for particle segmentation. In Proceedings of the 21st International Conf. on Patt. Rec. (ICPR2012), pages 801–804, Nov 2012.

Our Method: EDPCNN

Deep Learning

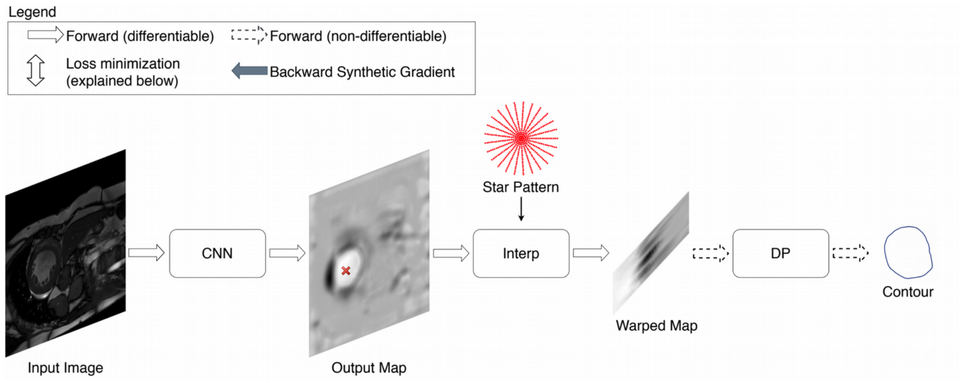


End-to-end learning of Deep Learning + Traditional method



Our Method: EDPCNN

EDPCNN: End-to-end CNN + Dynamic Programming



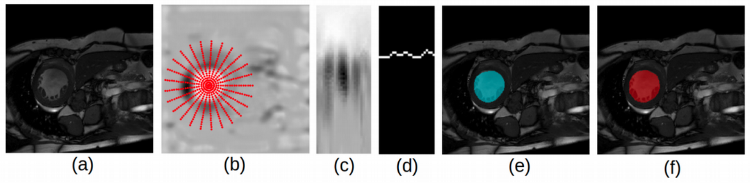


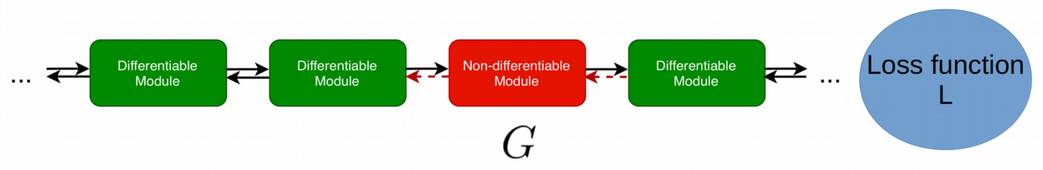
Figure 2: Illustrations of processing pipeline: (a) input image, (b) Output Map with an example star pattern, (c) Warped Map and (d) output indices indicating LV on the warped space (e) segmentation obtained with EDPCNN (f) ground truth.

Differentiable Bypass

The Dynamic Programming (Active Contour) algorithm

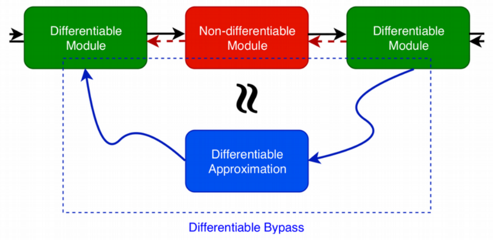
```
Algorithm 1: Dynamic programming
/* Construct value function $U$ and index function $I$ */
for $n = 1, \ldots, N-1$ do
    for $i, k = 1, \ldots, M$ do
        if $n == 1$ then
            $U(1,i,k) = \min_{1 \le j \le M}[E(1,i,j) + E(2,j,k)]$ ;
            $I(1,i,k) = \text{argmin}_{1 \le j \le M}[E(1,i,j) + E(2,j,k)]$ ;
        else
            $U(n,i,k) = \min_{1 \le j \le M}[U(n-1,i,j) + E(n+1,j,k)]$ ;
            $I(n,i,k) = \text{argmin}_{1 \le j \le M}[U(n-1,i,j) + E(n+1,j,k)]$ ;
        end
    end
end
/* Backtrack and output $v(1), \ldots, v(N)$ */
$v(1) = \text{argmin}_{1 \le j \le M}[U(N-1,j,j)]$;
$v(N) = I(N-1, v(1), v(1))$;
for $n = N-1, \ldots, 2$ do
    $v(n) = I(n-1, v(1), v(n+1))$;
end
```

Non-differentiable!



$\nabla_x G$ doesn't exist

Backpropagation does not work



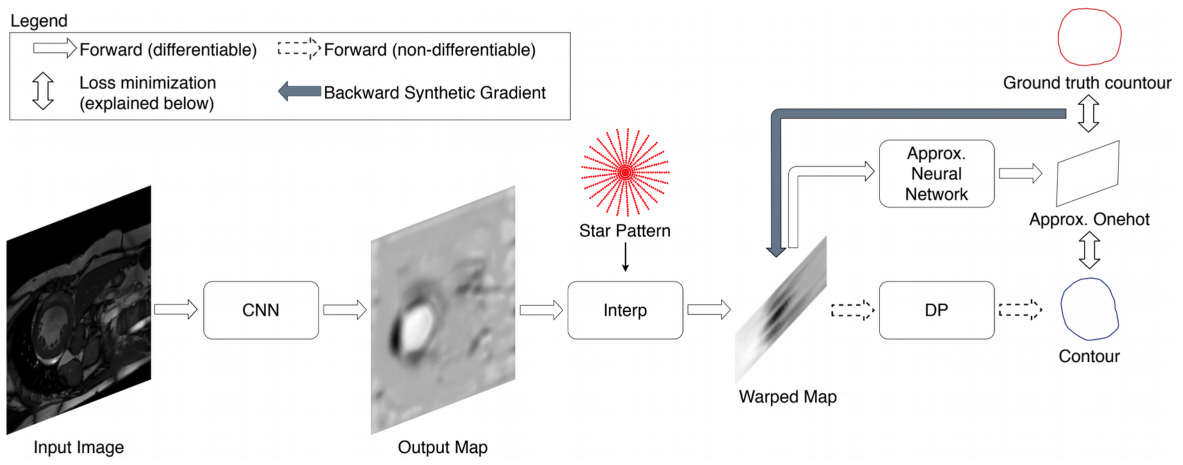
Train a **differentiable** function F(x) such that: $F(x) \approx G(x)$
(a good candidate for F(x) is a NN)

Substitute gradient of F for gradient of G in the backpropagation algorithm: $\nabla_x F \approx \nabla_x G$

Need to make sure F fit G well by introducing random exploration in the input space of G during training!

Differentiable bypass is applicable for any generic module, not just this dynamic programming algorithm

Our Method: EDPCNN

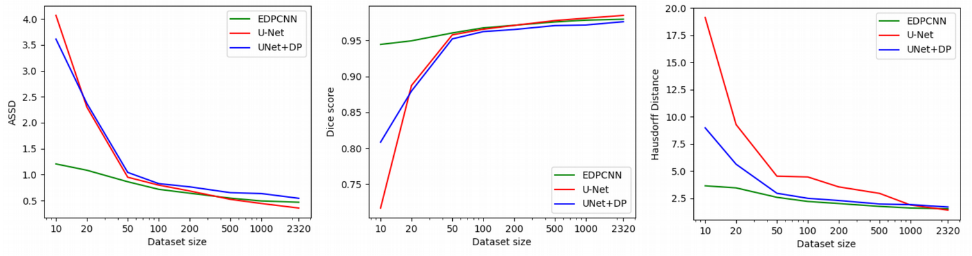


```
Algorithm 2: Training EDPCNN using synthetic gradients
for $J, p_{gt} \in$ Training {Image, Ground truth} batch do
    /* Compute Warped Map                                   */
    $g = Interp(Unet(J))$;
    /* Train approximating neural network                   */
    Initialize $s$ to 0;
    for $S$ steps do
        Sample $\varepsilon$ from $\mathcal{N}(0;1)$;
        $\min_\phi L(F(g+\sigma\varepsilon), DP(g+\sigma\varepsilon))$;        Exploration!
    end
    /* Train U-Net                                          */
    $\min_\psi \ L(F(g), p_{gt})$;
end
```

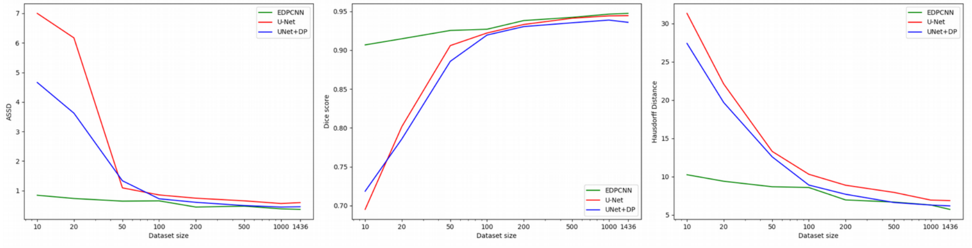
Experiment

- Dataset: ACDC and LVQuan2018
- Comparison between:
  - EDPCNN (End-to-end U+Net + active contour)
  - U-Net
  - U-Net + DP (EDPCNN not trained end-to-end)
    - U-Net is trained to predict the segmentation
    - Active contour is applied to refine the output.
- Training datasets with increasing size: from 10 images, 20 images... to full dataset.
- Results on the full validation set are reported, irrespective of training set size
- Report: Dice score, Average symmetric surface distance (ASSD) and Hausdorff distance (HD).

Experiments



ACDC dataset



LVQuan2018 dataset

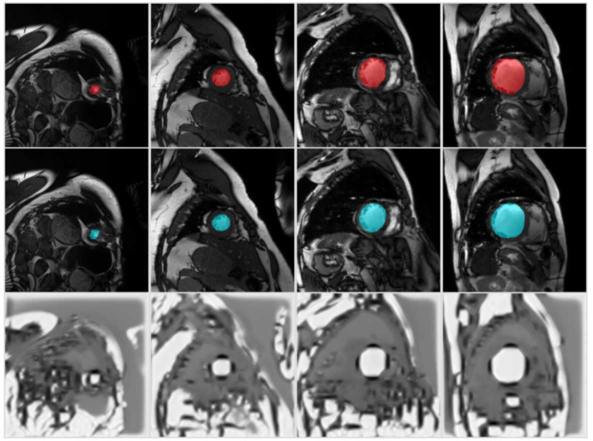


Figure 5: Example segmentation by EDPCNN on ACDC dataset. Top to bottom: ground truth mask (red), mask predicted by EDPCNN (blue), output map of EDPCNN. Left to right: object size from large to small.

Table 2: Detailed results for different methods at 10 training samples and full dataset size for ACDC dataset.

	10 training samples			Full dataset		
	Dice ↑	ASSD ↓	HD ↓	Dice ↑	ASSD ↓	HD ↓
UNet	0.695	7.00	31.34	0.944	0.59	6.88
UNet+DP	0.719	4.66	27.42	0.936	0.45	6.19
EDPCNN	**0.907**	**0.84**	**10.26**	**0.947**	**0.36**	**5.73**

Table 3: Detailed results for different methods at 10 training samples and full dataset size for LVQuan2018 dataset.

	10 training samples			Full dataset		
	Dice ↑	ASSD ↓	HD ↓	Dice ↑	ASSD ↓	HD ↓
UNet	0.717	4.07	19.13	**0.985**	**0.36**	**1.39**
UNet+DP	0.808	3.61	8.97	0.976	0.54	1.68
EDPCNN	**0.944**	**1.21**	**3.63**	0.980	0.47	1.51

Conclusion and summary

- Classical computer vision algorithms can be used to complement deep learning.
- Training neural networks end-to-end to adapt to classical algorithms is better than just applying CNN + classical algorithms directly.
- Differentiable bypass can facilitates learning of differentiable and non-differentiable modules together end-to-end.
- Pontential to use differentiable bypass for combining any differentiable and non-differentiable modules together, not just CNN and algorithm.

Thank you!

Nhat M. Nguyen, MSc

Nilanjan Ray, PhD