Beyond the Final Layer: Reflections on **Deeply-Supervised Nets**


Chen-Yu Lee*


Saining Xie*


Patrick Gallagher


Zhengyou Zhang


Zhuowen Tu

*equal contribution

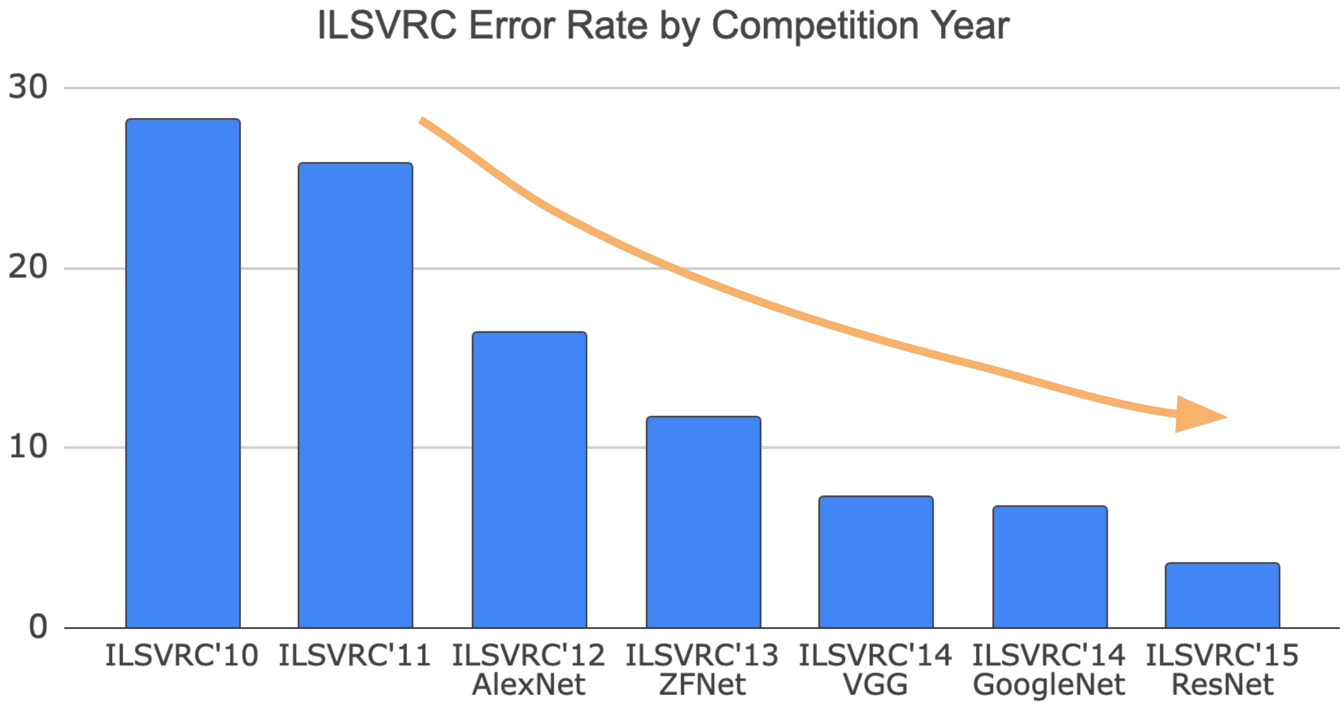
Test of Time, AISTATS 2025

AISTATS

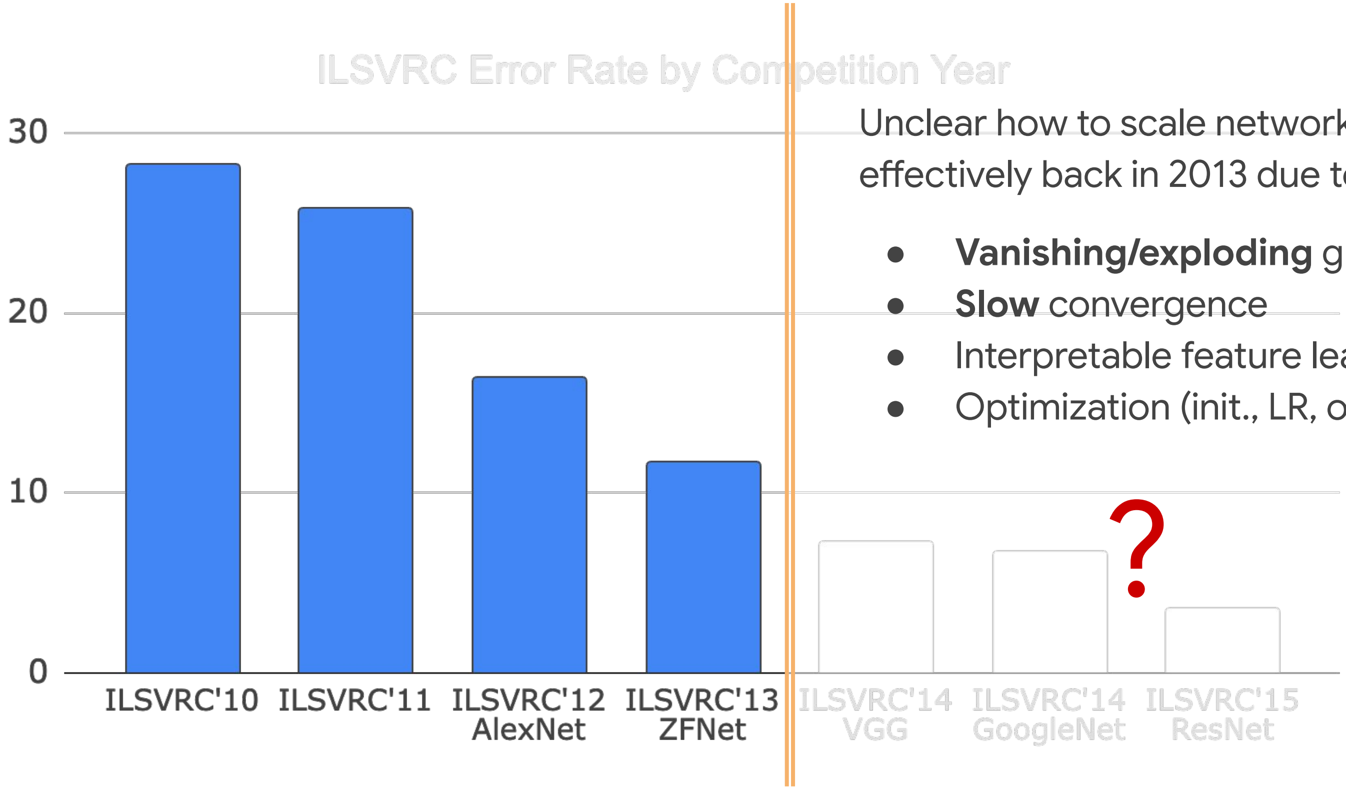
UC San Diego

Context & Challenges

Context: rapid development of much **deeper** networks since 2012



Challenges of training deeper networks back in 2013



Unclear how to scale network **depth** effectively back in 2013 due to:

- **Vanishing/exploding** gradients
- **Slow** convergence
- Interpretable feature learning
- Optimization (init., LR, optimizer, etc)

Proposed Solution

Question: can we force **intermediate layers** to learn classifiable features?

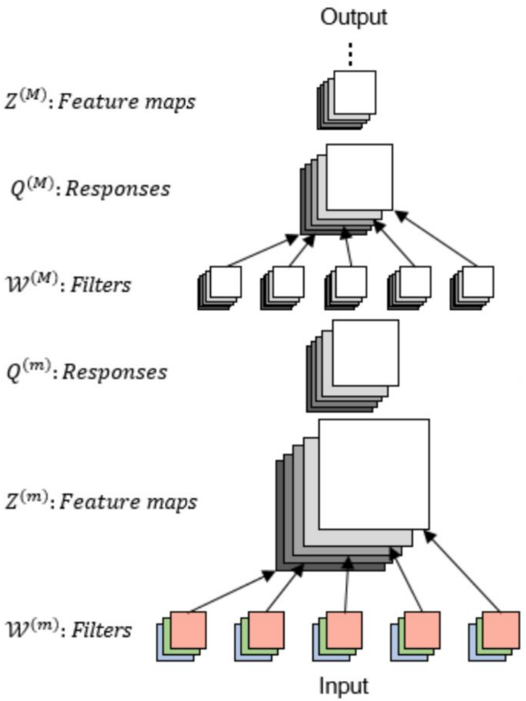


Illustration of a ConvNet (LeCun et al.)

Question: can we force **intermediate layers** to learn classifiable features?

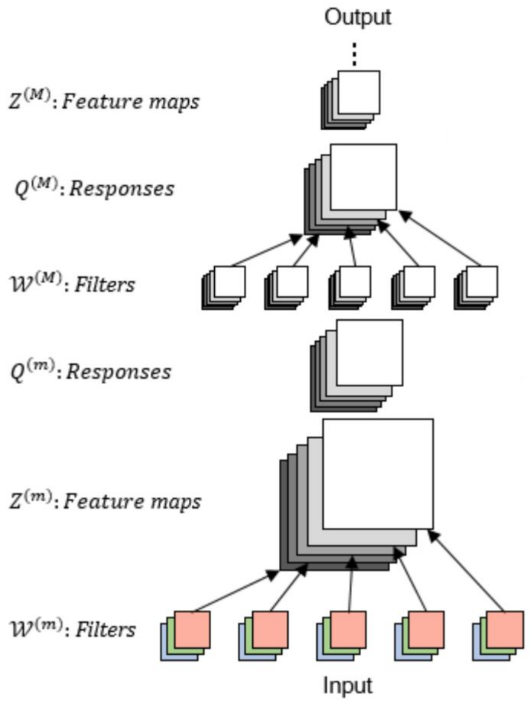
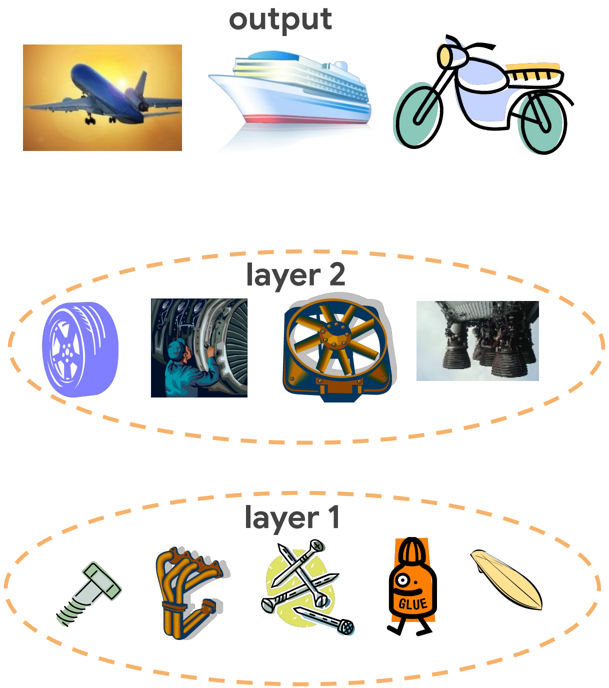


Illustration of a ConvNet (LeCun et al.)



Our proposal: introduce **auxiliary classifiers (deep supervision)** at intermediate layers

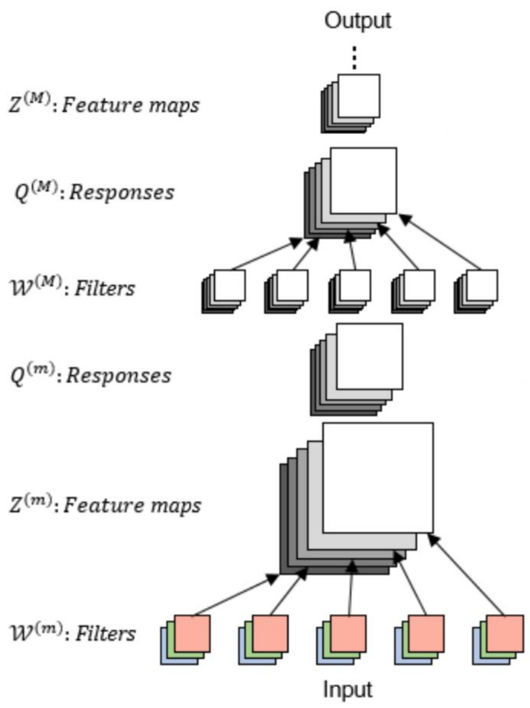
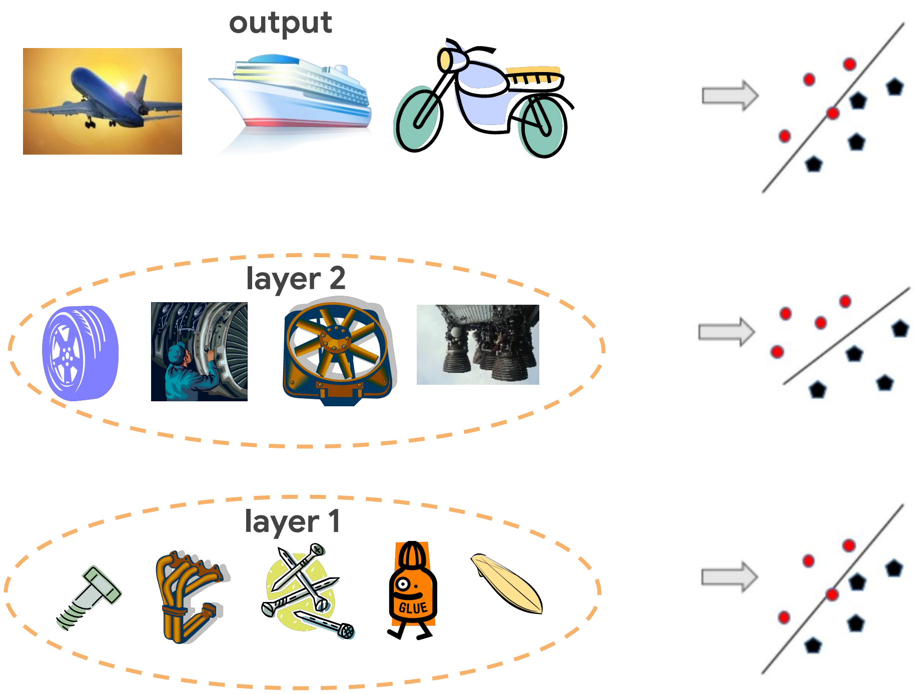


Illustration of a ConvNet (LeCun et al.)



Our proposal: introduce **auxiliary classifiers (deep supervision)** at intermediate layers

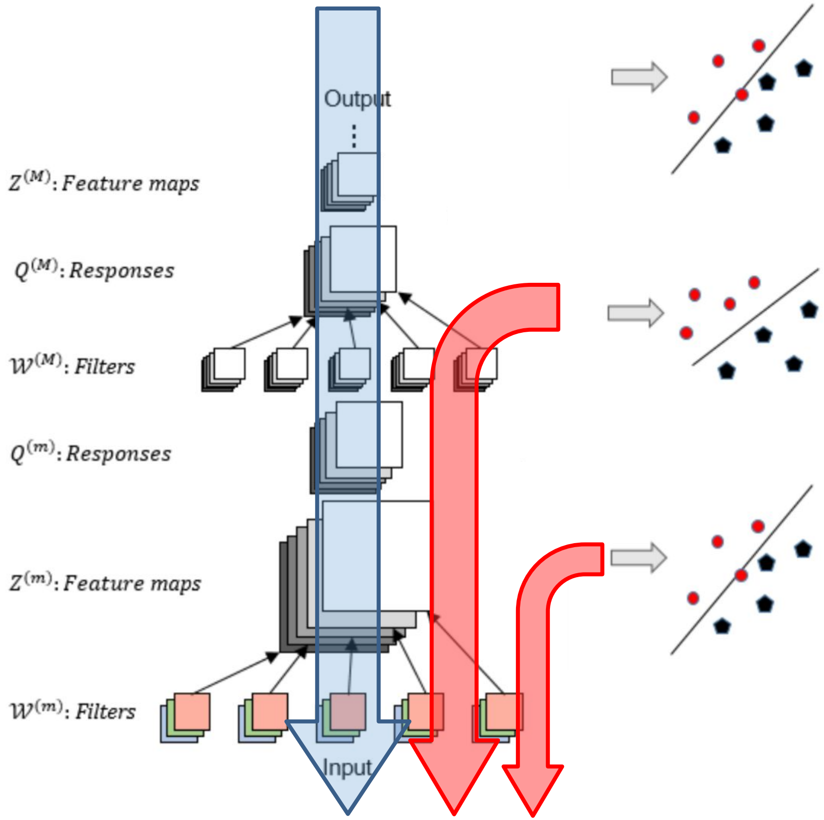
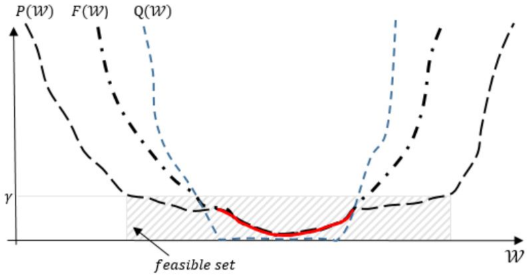


Illustration of a ConvNet (LeCun et al.)

Goal of the deep supervision

- Directly combat the **vanishing** gradient problem
- Encourage more **discriminative features** in the intermediate layers

A loose assumption

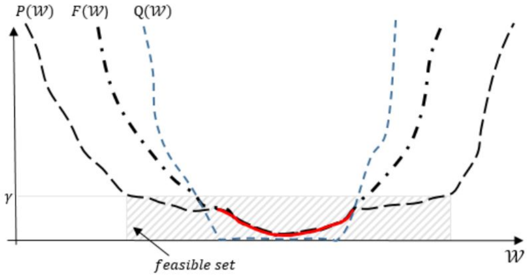


$$\|\mathbf{w}^{(out)}\|^2 + \mathcal{L}(\mathsf{W}, \mathbf{w}^{(out)}) + \sum_{m=1}^{M-1} \alpha_m [\|\mathbf{w}^{(m)}\|^2 + \ell(\mathsf{W}, \mathbf{w}^{(m)}) - \gamma]_+$$

$$F(\mathsf{W}) \equiv \mathcal{P}(\mathsf{W}) + \mathcal{Q}(\mathsf{W})$$

Theorem 1 *Let $\mathcal{P}(\mathsf{W})$ be λ_1-strongly convex and $\mathcal{Q}(\mathsf{W})$ be λ_2-strongly convex near optimal $\mathsf{W}^\star$ and denote by $\mathsf{W}_T^{(F)}$ and $\mathsf{W}_T^{(\mathcal{P})}$ the solution after T iterations when following SGD on $F(\mathsf{W})$ and $\mathcal{P}(\mathsf{W})$, respectively. Then DSN framework improves the relative convergence speed $\frac{\mathbb{E}[\|\mathsf{W}_T^{(\mathcal{P})} - \mathsf{W}^\star\|^2]}{\mathbb{E}[\|\mathsf{W}_T^{(F)} - \mathsf{W}^\star\|^2]}$, viewed from the ratio of their upper bounds as $\Theta(\frac{(\lambda_1+\lambda_2)^2}{\lambda_1^2})$, when $\eta_t = 1/\lambda t$.*

A loose assumption



$$\|\mathbf{w}^{(out)}\|^2 + \mathcal{L}(\mathsf{W}, \mathbf{w}^{(out)}) + \sum_{m=1}^{M-1} \alpha_m [\|\mathbf{w}^{(m)}\|^2 + \ell(\mathsf{W}, \mathbf{w}^{(m)}) - \gamma]_+$$

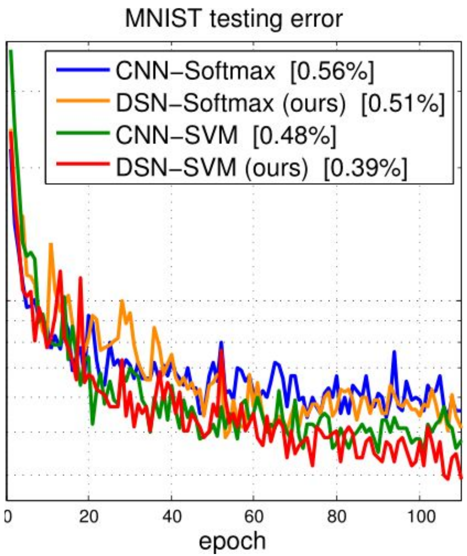
$$F(\mathsf{W}) \equiv \mathcal{P}(\mathsf{W}) + \mathcal{Q}(\mathsf{W})$$

Lemma 1 *$\forall m, m' = 1..M-1$, and $m' > m$ if $\|\mathbf{w}^{(m)}\|^2 + \ell((\hat{\mathsf{W}}^{(1)}, .., \hat{\mathsf{W}}^{(m)}), \mathbf{w}^{(m)}) \leq \gamma$ then there exists $(\hat{\mathsf{W}}^{(1)}, .., \hat{\mathsf{W}}^{(m)}, .., \hat{\mathsf{W}}^{(m')})$ such that $\|\mathbf{w}^{(m')}\|^2 + \ell((\hat{\mathsf{W}}^{(1)}, .., \hat{\mathsf{W}}^{(m)} .., \hat{\mathsf{W}}^{(m')}), \mathbf{w}^{(m')}) \leq \gamma$.* [2]

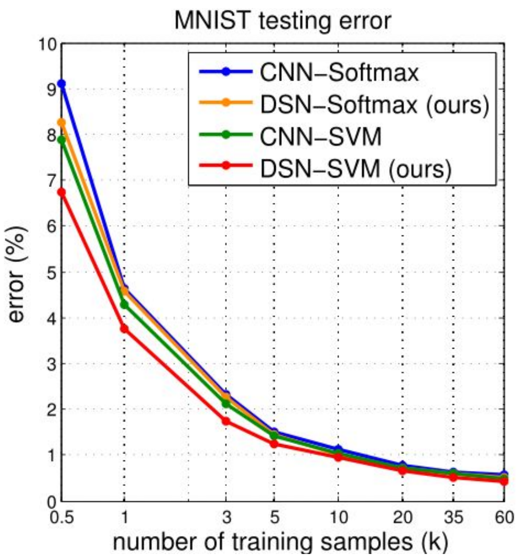
Lemma 2 *Suppose $\mathbb{E}[\|\hat{\mathbf{gp}}_t\|^2] \leq G^2$ and $\mathbb{E}[\|\hat{\mathbf{gq}}_t\|^2] \leq G^2$, and we use the update rule of $\mathsf{W}_{t+1} = \Pi_{\mathcal{W}}(\mathsf{W}_t - \eta_t(\hat{\mathbf{gp}}_t + \hat{\mathbf{gq}}_t))$ where $\mathbb{E}[\hat{\mathbf{gp}}_t] = \mathbf{gp}_t$ and $\mathbb{E}[\hat{\mathbf{gq}}_t] = \mathbf{gq}_t$. If we use $\eta_t = 1/(\lambda_1 + \lambda_2)t$, then at time stamp T*

$$\mathbb{E}[\|\mathsf{W}_T - \mathsf{W}^\star\|^2] \leq \frac{12G^2}{(\lambda_1 + \lambda_2)^2 T} \tag{9}$$

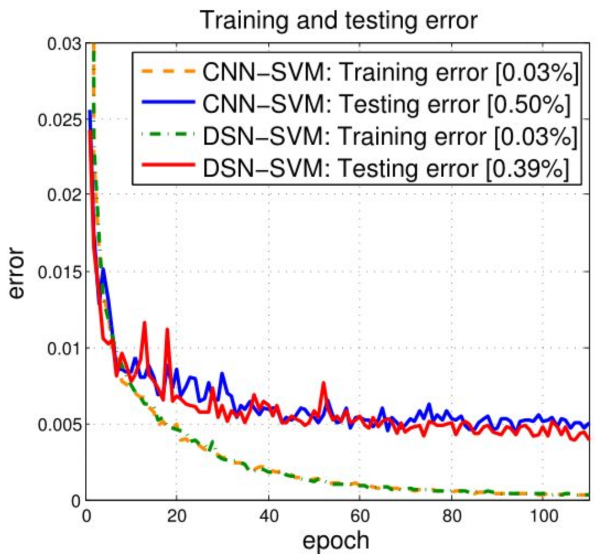
Deeply-Supervised Nets (DSN) on MNIST



DSN works well with various **loss functions**

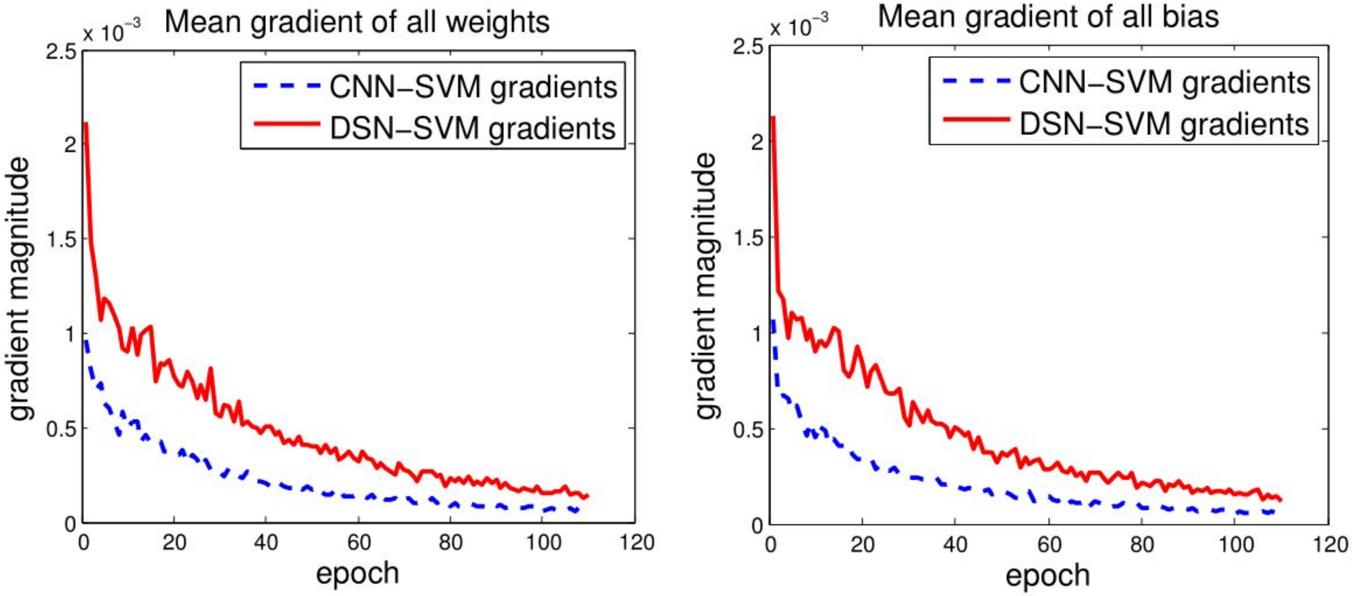


DSN **generalizes better** in low data regime (26% gain at 500 samples)



DSN **converges faster** without overfitting

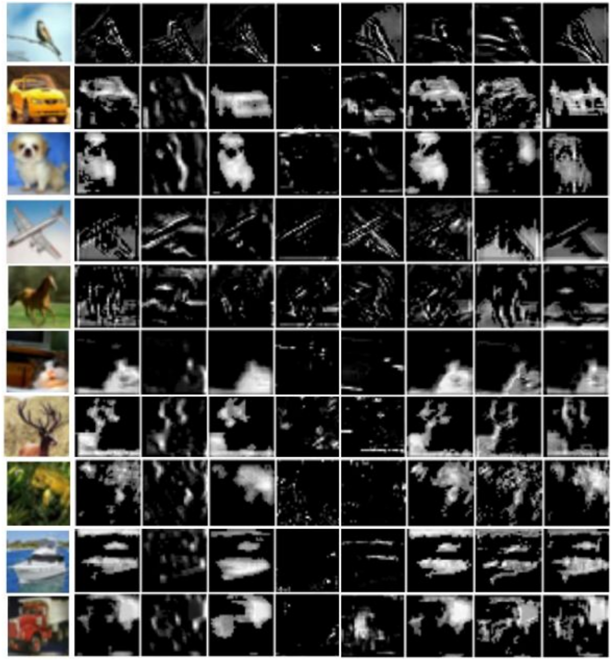
Deeply-Supervised Nets (DSN) on MNIST



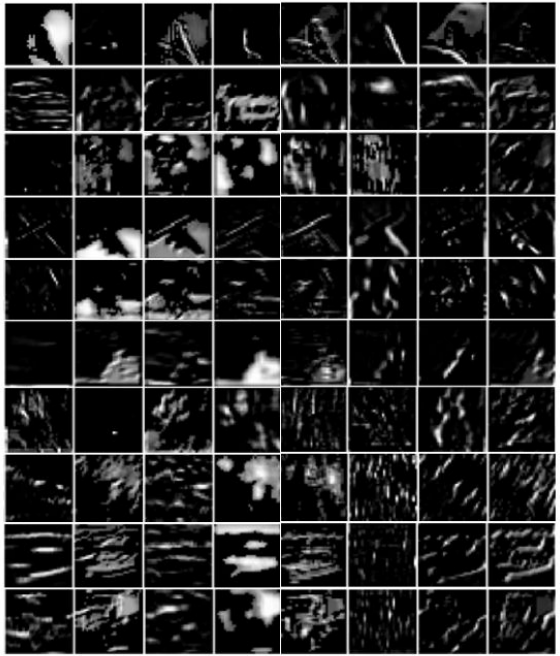
DSN provides **naturally higher gradient magnitude**
without artificially tuning up the learning rate

Deeply-Supervised Nets (DSN) generates more **intuitive** intermediate feature maps

w/ deep supervision



w/o deep supervision



Inspired by: M. Zeiler and R. Fergus. “Visualizing and understanding convolutional networks”, ECCV 2014.

Results

MNIST

Method	Error(%)
CNN [13]	0.53
Stochastic Pooling [32]	0.47
Network in Network [20]	0.47
Maxout Networks[9]	0.45
DSN (ours)	**0.39**

CIFAR 100

Method	Error(%)
Stochastic Pooling [32]	42.51
Maxout Networks [9]	38.57
Tree based Priors [27]	36.85
Network in Network [20]	35.68
DSN (ours)	**34.57**

CIFAR 10

Method	Error(%)
No Data Augmentation	
Stochastic Pooling [32]	15.13
Maxout Networks [9]	11.68
Network in Network [20]	10.41
DSN (ours)	**9.78**
With Data Augmentation	
Maxout Networks [9]	9.38
DropConnect [19]	9.32
Network in Network [20]	8.81
DSN (ours)	**8.22**

SVHN

Method	Error(%)
Stochastic Pooling [32]	2.80
Maxout Networks [9]	2.47
Network in Network [20]	2.35
Dropconnect [19]	1.94
DSN (ours)	**1.92**

Related work

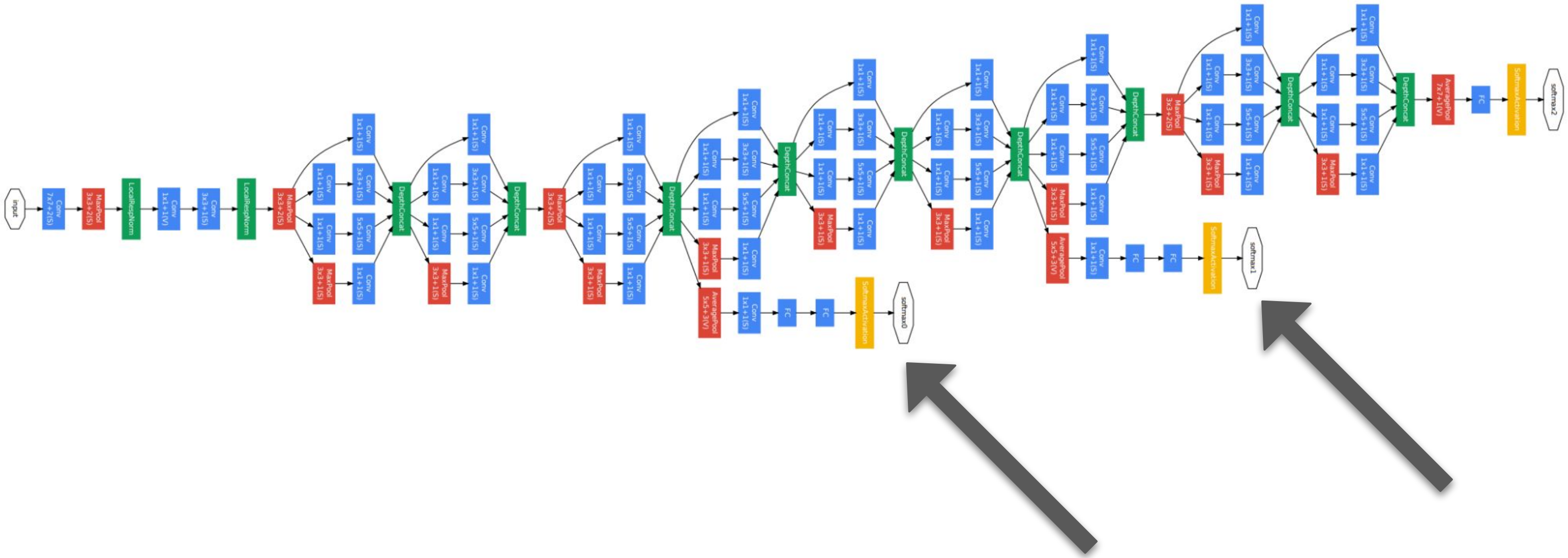
- **M. A. Carreira-Perpinan and W. Wang, "Distributed optimization of deeply nested systems.", AISTATS 2014.**
    - *Penalty-based methods using alternating optimization*
- **P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks", IJCNN, 2011.**
    - *The output of the 1st stage, together with the final stage output, is also fed to the classifier*
- **Z Tu, "Auto-context and its application to high-level vision tasks", CVPR 2008.**
    - *Trains classifiers by using iteratively refined probability maps from previous steps as context alongside image features*
- **Y. Bengio et al. "Greedy layer-wise training of deep networks". NIPS, 2007.**
    - *Solve optimization problems through layer-wise training*

[non-exhaustive]

Reflections & Impact

Reflections: GoogLeNet employed 2 auxiliary classifiers to aid gradient flow

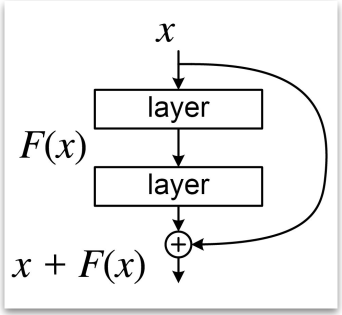
GoogLeNet



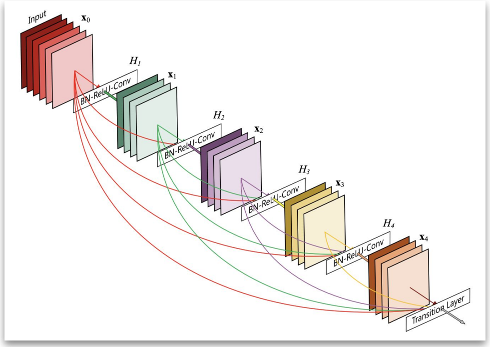
Szegedy et al. Going Deeper with Convolutions. CVPR 2015 [66k citations]

Reflections: many more approaches have then been proposed for better training

ResNets [266k citations]



DenseNets [53k citations]

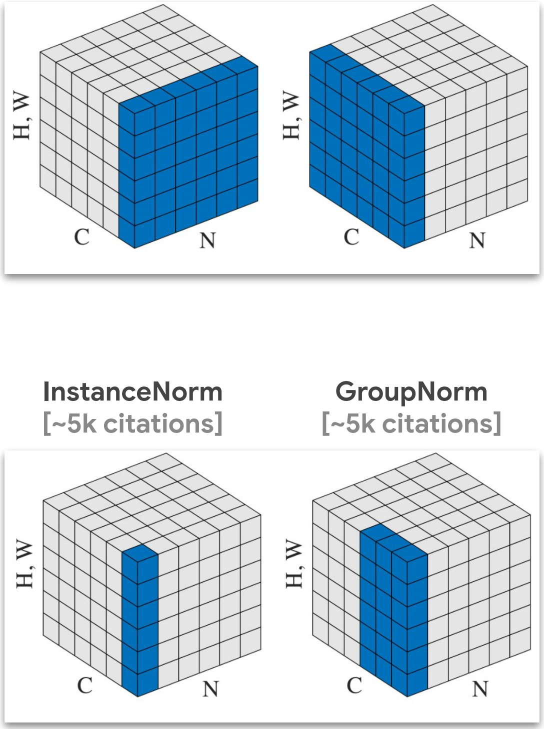


BatchNorm [61k citations]

LayerNorm [15k citations]

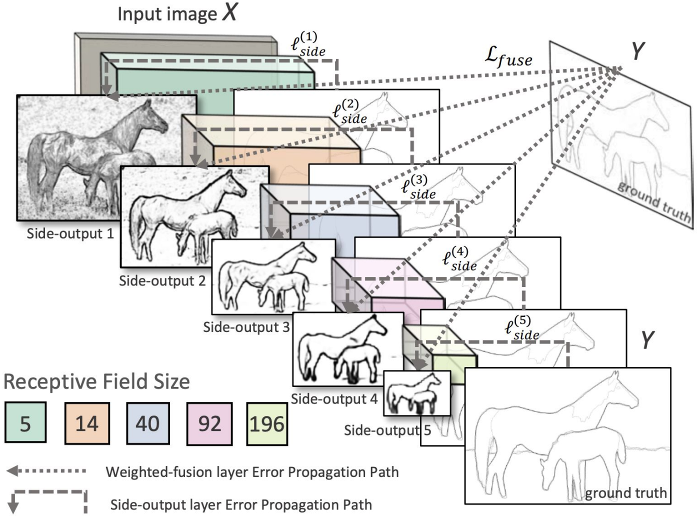
InstanceNorm [~5k citations]

GroupNorm [~5k citations]



Impact: edge detection

Holistically-Nested Edge Detection (HED)



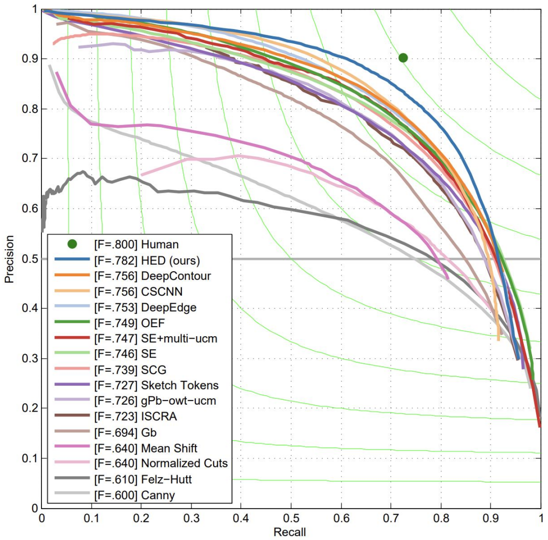


Figure 5. Results on the BSDS500 dataset. Our proposed HED framework achieves the best result (ODS=.782). Compared to several recent CNN-based edge detectors, our approach is also orders of magnitude faster.

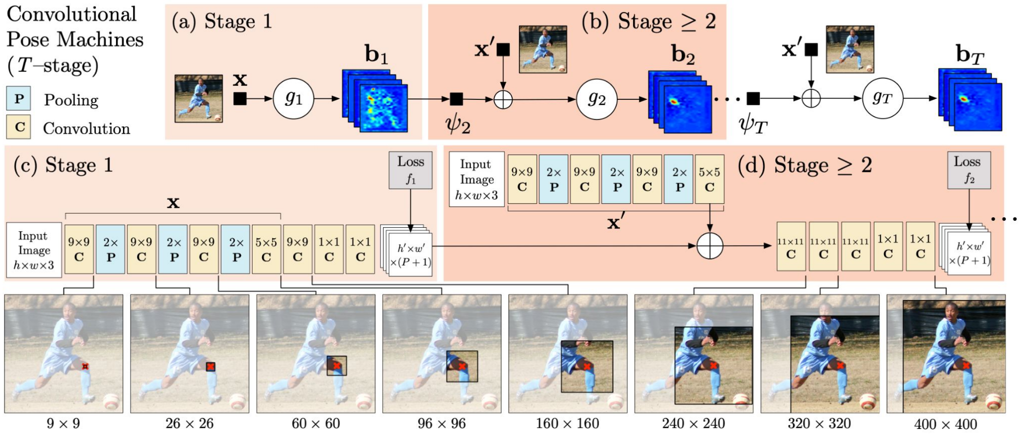
The application of deep supervision to a fully convolutional net (FCN) shows a great performance boost and produces more intuitive **multi-scale** feature maps.

S Xie and Z Tu. Holistically-Nested Edge Detection. ICCV 2015 [4k citations]

Impact: human pose estimation

Convolutional Pose Machines (CPMs)

Input Image (a) Stage 1 (b) Stage 2 (c) Stage 3



Intermediate supervision addresses vanishing gradients for **sequential** structured prediction

SH Wei et al. Convolutional Pose Machines. CVPR 2016 [~4k citations]

Impact: scene parsing

Pyramid Scene Parsing Network (PSPNet)

(a) Image (b) Ground Truth

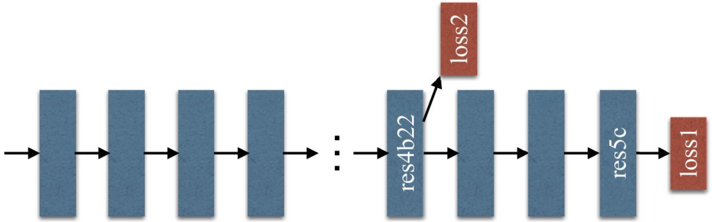


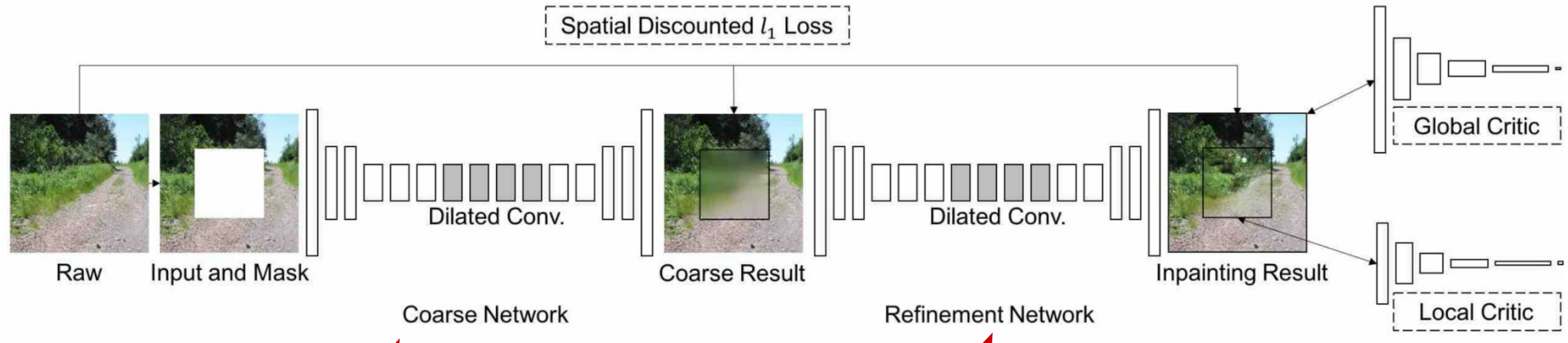
Figure 4. Illustration of auxiliary loss in ResNet101. Each blue box denotes a residue block. The auxiliary loss is added after the res4b22 residue block.

The auxiliary loss helps optimize the learning process

H Zhao et al. Pyramid Scene Parsing Network. CVPR 2017 [17k citations]

Impact: image inpainting

Generative Image Inpainting with Contextual Attention



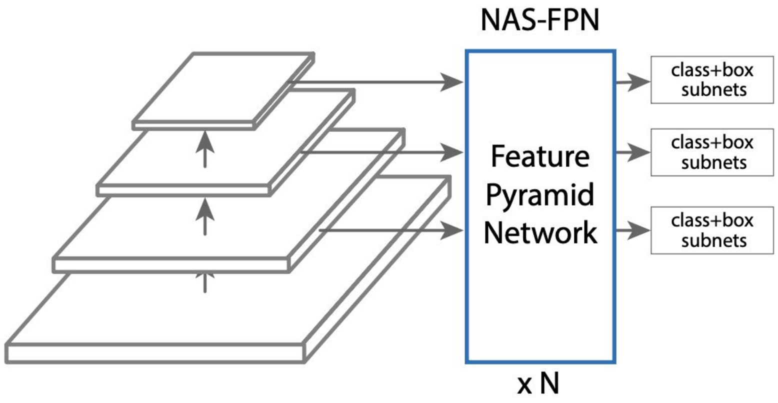
(a) reconstruction loss

(a) reconstruction loss +
(b) global WGAN-GP + (c) local WGAN-GP

J Yu et al. Generative Image Inpainting with Contextual Attention. CVPR 2018 [3k citations]

Impact: **early exit** for object detection

NAS-FPN

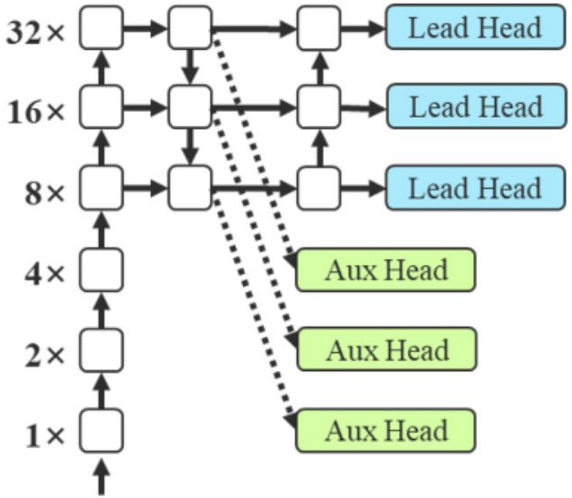


- Attach classifier and box regression heads after **all intermediate** pyramid networks.
- This enables **anytime detection** which can generate detection results with early exit

G Ghiasi et al. NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. CVPR 2019 [2k citations]

Impact: object detection with **coarse-to-fine** deep supervisions

YOLOv7



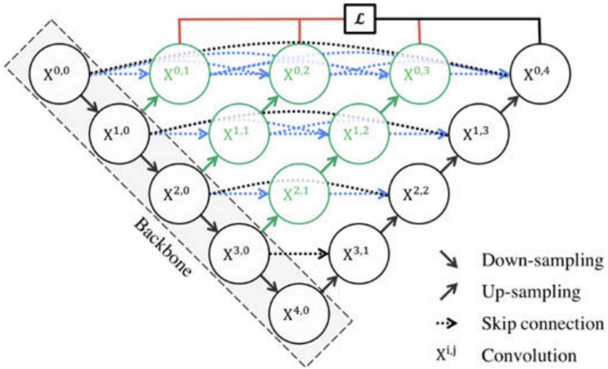
- Even for architectures that converge well, deep supervision can still **significantly improve** the performance

- Use lead head prediction as guidance to generate **coarse-to-fine hierarchical labels**, which are used for auxiliary head learning

CY Wang et al. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. CVPR 2023 [11k citations]

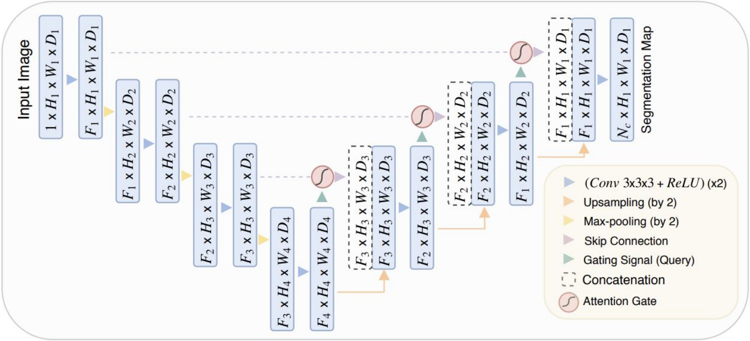
Impact: medical image analysis

UNet++



Deep supervision leads to **marked improvement** for liver and lung nodule segmentation

Attention U-Net



Use deep supervision to force the **intermediate** feature maps to be semantically **discriminative** at each image scale

Z Zhou et al. UNet++: A Nested U-Net Architecture for Medical Image Segmentation. DLMIA 2018 [9k citations]
O Oktay et al. Attention U-Net: Learning Where to Look for the Pancreas. MIDL 2018 [~8k citations]

Conclusion

Deep supervision offers several benefits for neural nets

- For relatively shallow networks, it provides strong regularization, helping to reduce test error.
- For deeper networks, deep supervision greatly relieves the vanishing gradient problem, which otherwise makes the learning process very challenging.
- Deep supervision allows combination with various loss types (e.g., multi-scale, coarse-to-fine, different modalities) at different layers for complex tasks
- Deep supervision enables early exit for real-time applications.

Deeply-Supervised Nets

Q&A

Chen-Yu Lee*
Saining Xie*
Patrick Gallagher
Zhengyou Zhang
Zhuowen Tu

*equal contribution