Semantic Segmentation with Incomplete Annotations

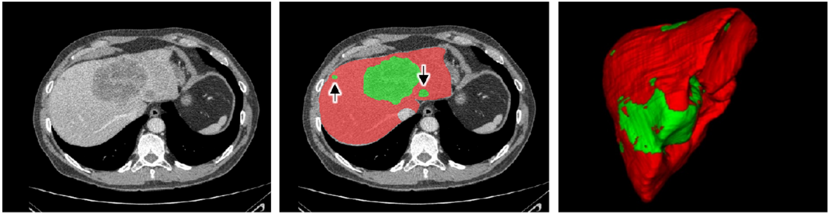
Medical Computer Vision and Health Informatics Workshop

Nicolas Thome - Joint work with O. Petit, L. Soler
Cnam Paris - nicolas.thome@cnam.fr
CEDRIC Lab - Machine Learning Team (MSDMA)
June 18, 2018

Outline

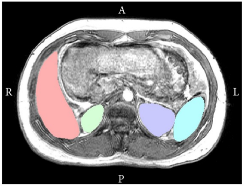
Context: Semantic Segmentation of Medical Images

- Semantic Segmentation: class label for each image pixel / voxel
- Deep ConvNets: tremendous sucess for visual recognition
- Semantic Segmentation of natural images: Fully Convolutional Networks (FCN), *e.g.* DeepLab [Chen et al., 2018]
    - Adpated FCN architectures for medical images, *e.g.* U-Net [Ronneberger et al., 2015]
    - FCN: base architecture for leading approaches in recent medical segmentation challenges, *e.g.* LITS'17 [Han, 2017, Li et al., 2017]

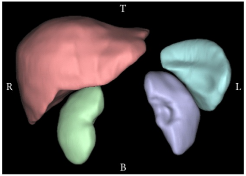


Datasets for Medical Image Semantic Segmentation

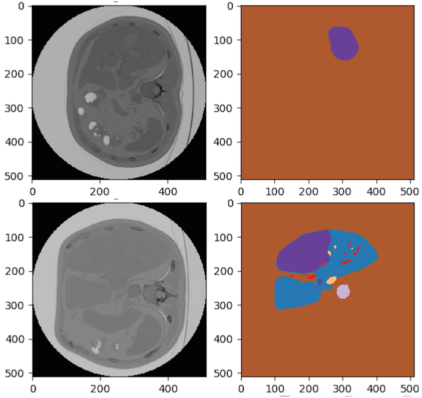
- ConvNets: large amount of data with clean annotations
- Annotation very costly for semantic segmentation: pixel-level labeling
    - Exacerbated in medical images: 3D data, highly qualified professionals needed, *e.g.* tumors (extreme appearance variations)



(a)

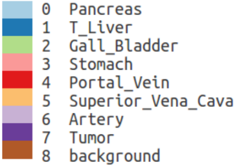


(b)



Semantic Segmentation of 3D CT-scans

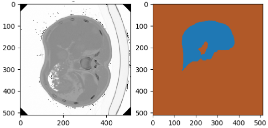
- Internal dataset[1]: ~ 1000 patients of $100 \times 512 \times 512$ images



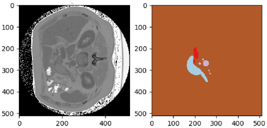
🟦	0	Pancreas
🟦	1	T_Liver
🟩	2	Gall_Bladder
🟥	3	Stomach
🟥	4	Portal_Vein
🟧	5	Superior_Vena_Cava
🟪	6	Artery
🟪	7	Tumor
🟫	8	background

- 3D segmentation: focusing on 2D slices
  $\Rightarrow$ independent training in each image

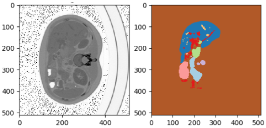
Liver

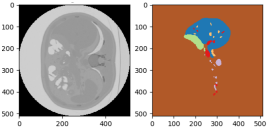


Pancreas

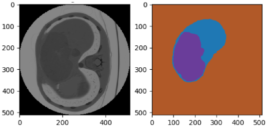


Stomach





Gall blader



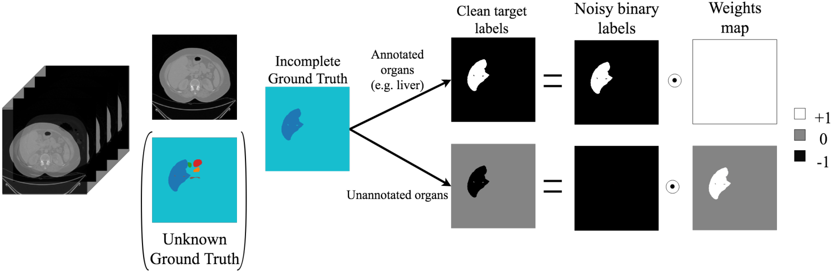
Tumor

---

[1] IRCAD: `https://www.ircad.fr/fr/`

Semantic Segmentation with Incomplete Annotations

- Large scale dataset, BUT:
  - Clinical experts: focus on a subset of organs
    ⇒ **Incomplete annotations** wrt full Ground Truth



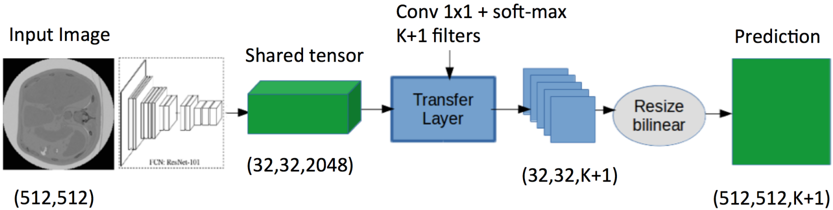
- **How to train deep ConvNets in this context ?**
  - Organ(s) missing the whole volumes, but: organ segmented in volume ⇒ complete annotation for that class
  - **Core idea:** generating clean target labels from noisy input labels
    - Binary mask w_k for each class ⇒ ambiguous *vs* non-ambiguous pixels

Outline

1. Context
2. **Semantic Segmentation with Incomplete Annotations**
3. Experiments

Semantic Segmentation with Incomplete Annotations

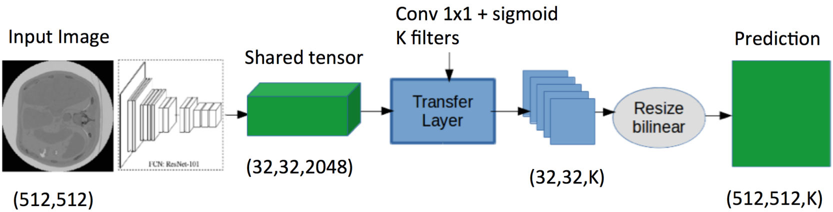
- Standard FCN not adapted to this context, *e.g.* DeepLab [Chen et al., 2018]



  - Shared Fully Convolutional Layers, ResNet [He et al., 2016]
  - Last tensor: 1×1 conv + soft-max $\Rightarrow$ single class prediction
  - **Incomplete annotation: "background" $\Leftrightarrow$ missing organ**
    $\Rightarrow$ conflict with pixels with proper organ annotations during training

Semantic Segmentation with Incomplete Annotations

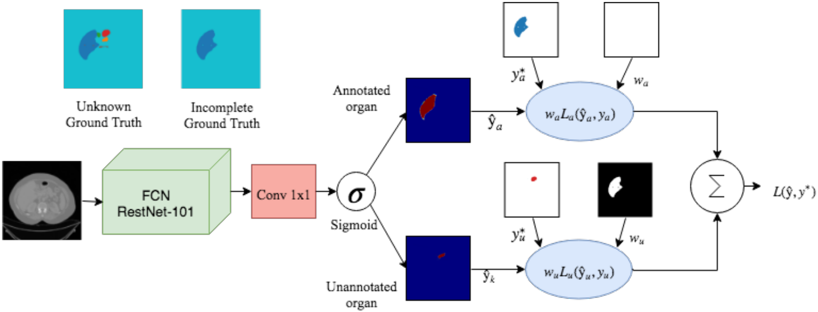
- Our approach for **S**emantic segmentation with **MI**ssing **L**abels and convn**E**ts (SMILE)
- Depart from the $(K+1)$ multi-class classification formulation, classify each organ independently using K binary classifiers



SMILE Training

- Binary CE loss at each pixel: $L_k(\hat{y}_k, y_k^*) = -\ (y_k^*\ log(\hat{y}_k) + (1 - y_k^*)\ log(1 - \hat{y}_k))$
- Final loss: weighted sum of binary losses:

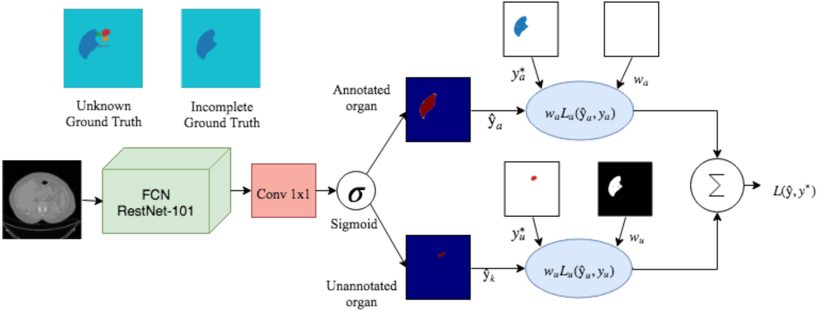
$$L(\hat{y}, y^*) = \sum_{k=1}^{K} w_k\ L_k(\hat{y}_k, y_k^*)$$



SMILE Training

- **Core SMILE component: binary weight maps $\mathbf{w_k} \in \{0; 1\}$**
  - Selecting or ignoring each pixel for class k
    - Class k present in volume: $w_k = 1 \ \forall$ pixel in volume
    - Class k absent:

$$w_k = \begin{cases} 1 \text{ if } \exists k' \neq k \text{s.t.} w_{k'} = 1 \ (\Rightarrow y_k^* = -1), \\ 0 \text{ otherwise (pixel ignored)} \end{cases}$$



SMILE Training

- **Analysis of labels used by FCN baseline and SMILE *vs* Ground Truth (GT)**
- For class k:
  - β_k ratio of voxels in a volume
  - α the ratio of missing labels for this organ in the dataset.

Baseline FCN

Used GT	Pos	Neg
Pos	$(1-\alpha)\cdot\beta_k$	$\alpha\cdot\beta_k$
Neg	0	$1-\beta_k$

SMILE

Used GT	Pos	Neg
Pos	$(1-\alpha)\cdot\beta_k$	0
Neg	0	$(1-\alpha)\cdot(1-\beta_k)+\epsilon$

$$\epsilon = \sum_{k'\neq k} \beta_{k'}$$

- **Both baseline and SMILE: only true positive**
  - **BUT only use $(1-\alpha)\cdot\beta_k$ vs β_k**

SMILE Training

Baseline FCN

Used GT	Pos	Neg
Pos	$(1-\alpha)\cdot\beta_k$	$\alpha\cdot\beta_k$
Neg	0	$1-\beta_k$

SMILE

Used GT	Pos	Neg
Pos	$(1-\alpha)\cdot\beta_k$	0
Neg	0	$(1-\alpha)\cdot(1-\beta_k)+\epsilon$

$$\epsilon = \sum_{k' \neq k} \beta_{k'}$$

- **Baseline:**
  - **False Negatives (FN)**: $\alpha \cdot \beta_k$, *i.e.* unannotated pixels indeed belonging to the organ
  - $\boxed{\frac{\textbf{TP}}{\textbf{FN}} = \frac{1-\alpha}{\alpha} : \ \alpha > \mathbf{0.5} \Rightarrow \frac{\textbf{TP}}{\textbf{FN}} < \mathbf{1}}$
- **SMILE:**
  - **Only true positives and true negatives**
  - **Less true negatives than baseline:** $(1-\alpha)\cdot(1-\beta_k)+\epsilon$ *vs* $(1-\beta_k)$
    - $\approx \alpha$ less negatives, but as $\beta << 1$, *e.g.* $\beta = 0.05$[2]
      $\Rightarrow$ **in practice, largely enough negative to train**

---

[2] organs ⇔ small volume portion

SMILEr Training

Incremental self-supervision and relabeling

- SMILE True Positives (TP) labels $\propto (1-\alpha)$
- **<u>Motovation:</u> automatically increasing number of TP labels**
  - Compensate for incomplete annotations
- Auto-supervision: create target positive labels
  $\Rightarrow$ **SMILEr** (re-labeling)
- Using a curriculum strategy [Bengio et al., 2009]
  1. Train ConvNet with SMILE: certain labels only, *i.e.* true positives and negatives $\Rightarrow$ **"easy samples"**
  2. Seek for new true positives with current model
     - **"Harder samples"**, automatic labeling
     - Use this new labels as target to train a new model with more positives
     - Iterate
- $\frac{TP}{FP}$: key indicator of SMILEr sucess

SMILEr Training

- SMILEr algorithm: applied for each binary organ classifier independently[a]

Algorithm 1 Algorithm for training SMILEr for class k

```
Require: Training set $\{(\mathbf{x}_i, \mathbf{y}_i^*)\}$, $\gamma_{max}$, $T$, SMILE model $m_0$ for class $k$.
Initialize $\mathbf{y}_{i,0}^* = \mathbf{y}_i^*$, $N_u \leftarrow$ number of unannotated images for class $k$
for t=1 to T do
    $\gamma_t = \frac{t}{T}\gamma_{max}$
    for i=1 to $N_u$ do
        $\hat{y}_i{}^{+} \leftarrow (m_t, \mathbf{x}_i)$ // Find predicted positive pixels by $m_t$ in image $\mathbf{x}_i$
        $y_{i,t}^{*,+} \leftarrow (m_t, \mathbf{x}_i, \gamma_t, \hat{y}_k{}^{+})$ // Assign new ⊕ taget labels
        $y_{i,t}^* = y_{i,t-1}^* \cup y_{i,t}^{*,+}$ // Augment training set
    end for
    $m_t = train(\{(\mathbf{x}_i, \mathbf{y}_{i,t}^*)\},)$ // Re-train model with augmented training set
end for
Ensure: SMILEr Model $m_T$
```

---

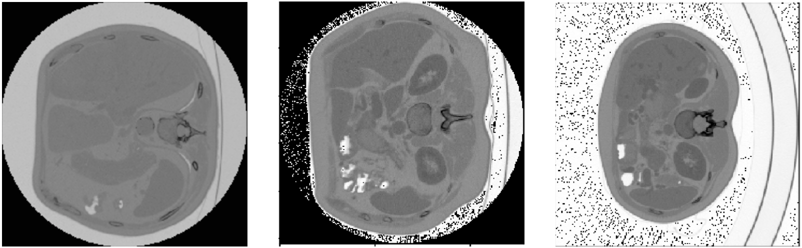
[a] Ignoring the dependence on class k for the sake of clarity.

Outline

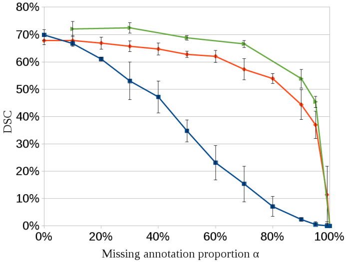
1. Context
2. Semantic Segmentation with Incomplete Annotations
3. Experiments

Dataset and setup

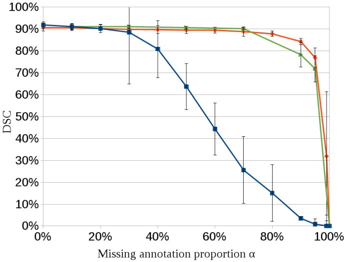
- Experiments on sub-set of our dataset with complete ground truth annotations
- 72 3D CT-scan volumes (~ 100 512×512 images) for three organs: liver, pancreas and stomach
- Partially annotated dataset generated: randomly removing $\alpha\%$ of organs in the volumes independently
- Comparison of our methods (SMILE, SMILEr) wrt DeepLab baseline
  - Train 80% / Test (20%), $K = 5$ datasplits



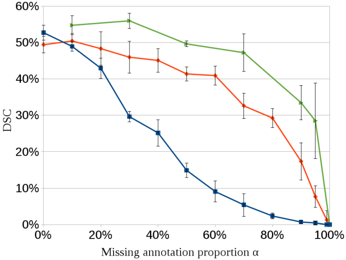
Quantitative results



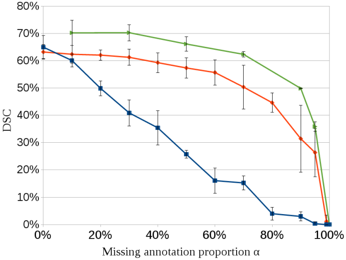
Mean



Liver



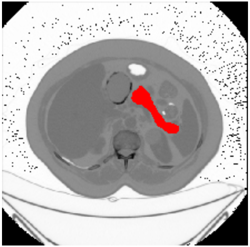
Pancreas



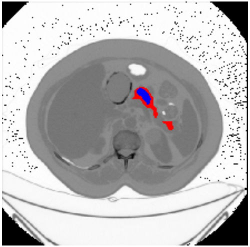
Stomach

SMILEr re-labeling, $\alpha = 50\%$

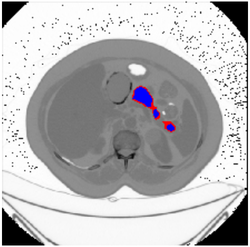
GT



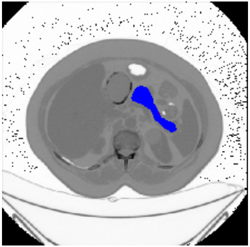
SMILEr $t = 1$



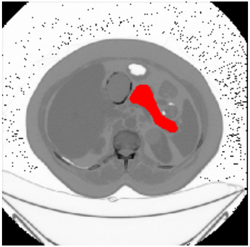
SMILEr $t = 2$



$T = 3$, $\gamma_{max} = 1.0$

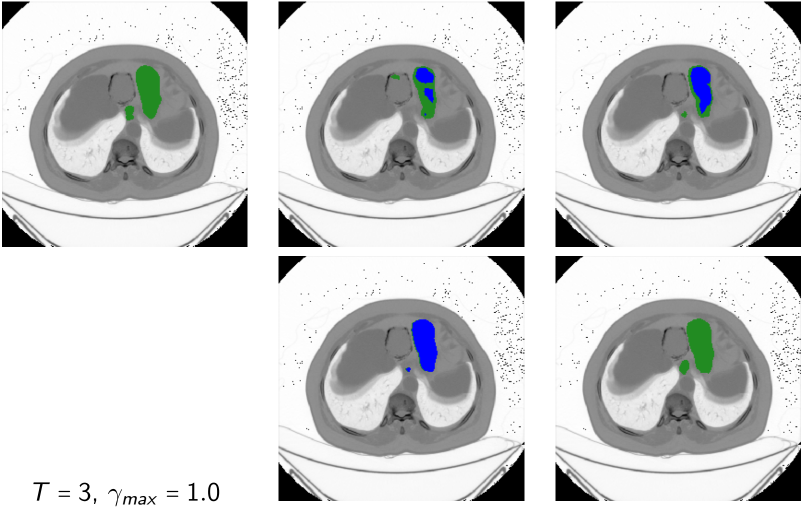


SMILEr $t = 3$



Final prediction

SMILEr re-labeling, $\alpha = 70\%$



GT

SMILEr $t = 1$

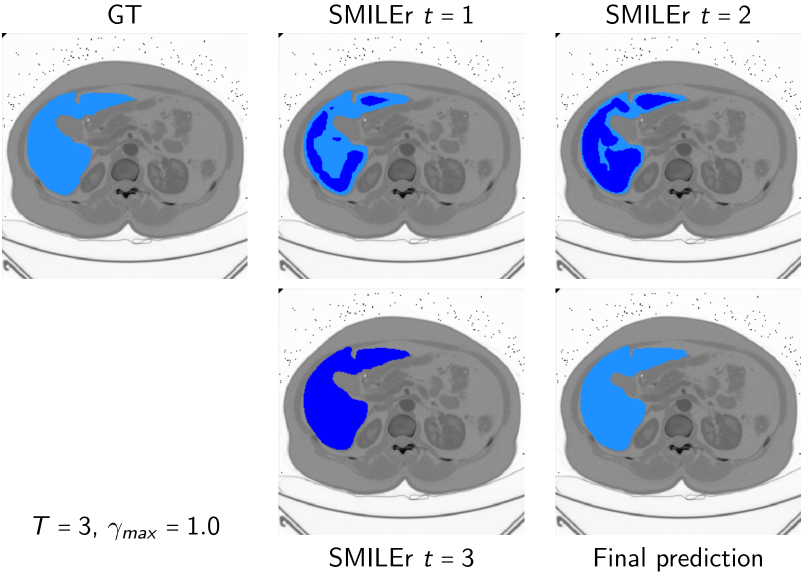
SMILEr $t = 2$

SMILEr $t = 3$

Final prediction

$T = 3$, $\gamma_{max} = 1.0$

SMILEr re-labeling, $\alpha = 70\%$



GT

SMILEr $t = 1$

SMILEr $t = 2$

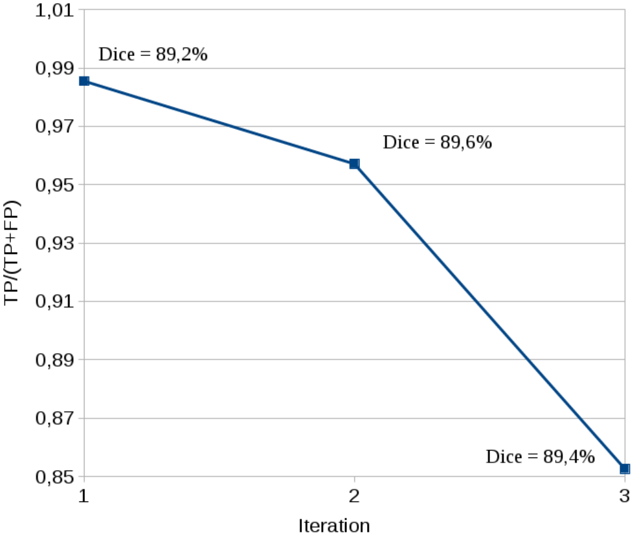
SMILEr $t = 3$

Final prediction

$T = 3$, $\gamma_{max} = 1.0$

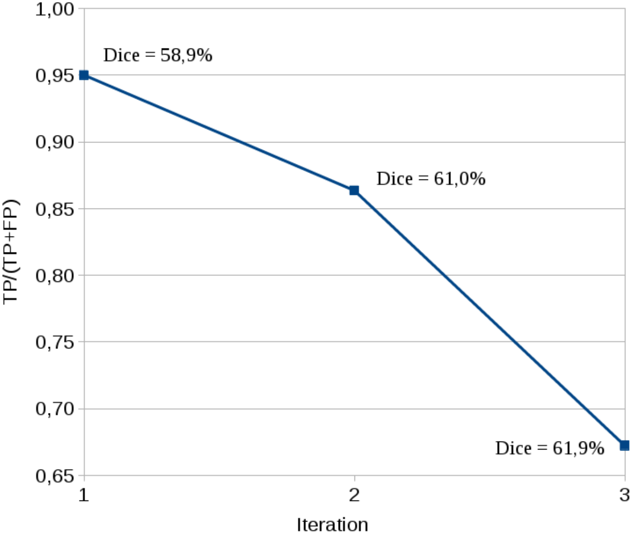
Re-labeling method

- $\frac{TP}{TP+FP}$ vs Curriculum iterations for Liver (α = 70%)



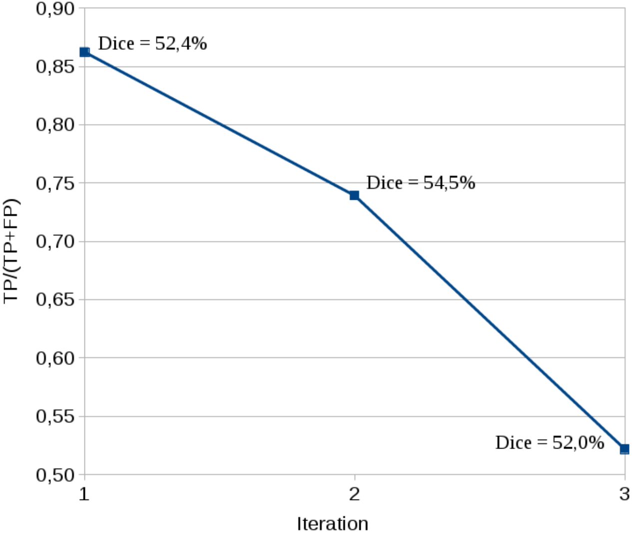
Re-labeling method

- $\frac{TP}{TP+FP}$ vs Curriculum iterations for Stomach (α = 70%)

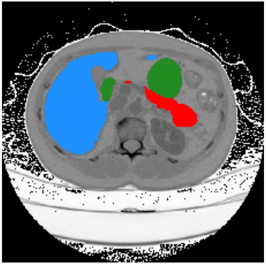


Re-labeling method

- $\frac{TP}{TP+FP}$ vs Curriculum iterations for Pancreas (α = 70%)

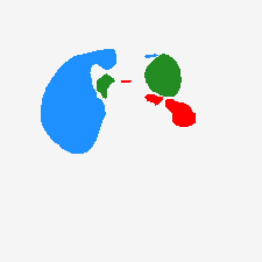


Segmentation results, $\alpha = 70\%$

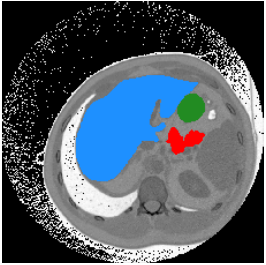


GT

baseline

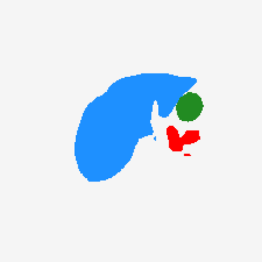


SMILEr



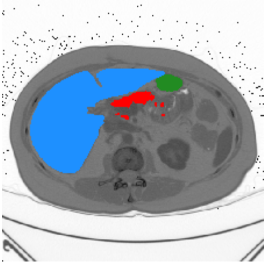
GT

baseline

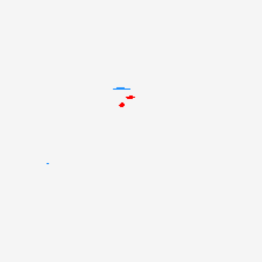


SMILEr

Segmentation results, $\alpha = 70\%$

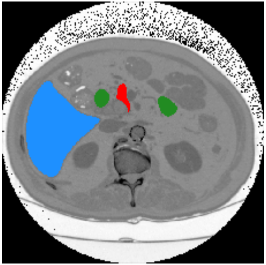


GT



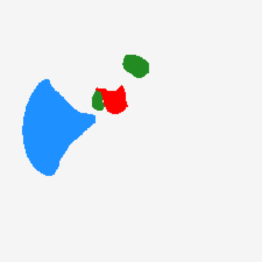
baseline

SMILEr



GT

baseline



SMILEr

Conclusion

- Method for learning with incomplete ground truth annotations
- First stage: train only with correct label
- Second stage: re-label positives
- Future works:
  - Evaluation in larger datasets with more classes
  - Using 3D conv backbones models
  - Using uncertainty estimate [Kendall and Gal, 2017] for selecting target auto-supervision labels

GT SMILEr

Thank you for your attention!

Joint work with:

- Olivier Petit, PhD Student
- Luc Soler, Prof. at IRCAD, Visible Patient CEO

Questions?

References I

[Bengio et al., 2009] Bengio, Y., Louradour, J., Collobert, R., and Weston, J. (2009).
Curriculum learning.
In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 41–48.

[Chen et al., 2018] Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2018).
Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs.
IEEE Trans. Pattern Anal. Mach. Intell., 40(4):834–848.

[Han, 2017] Han, X. (2017).
Automatic liver lesion segmentation using A deep convolutional neural network method.
CoRR, abs/1704.07239.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016).
Deep residual learning for image recognition.
In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778.

[Kendall and Gal, 2017] Kendall, A. and Gal, Y. (2017).
What uncertainties do we need in bayesian deep learning for computer vision?
In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 5574–5584. Curran Associates, Inc.

[Li et al., 2017] Li, X., Chen, H., Qi, X., Dou, Q., Fu, C., and Heng, P. (2017).
H-denseunet: Hybrid densely connected unet for liver and liver tumor segmentation from CT volumes.
CoRR, abs/1709.07330.

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015).
U-Net: Convolutional Networks for Biomedical Image Segmentation.
In *MICCAI (3)*, volume 9351 of *Lecture Notes in Computer Science*, pages 234–241. Springer.