## Detailed analysis of the DL semantic segmentation reviewed articles

### First FCN papers

#### Li et al. (2015)

The aim of the study was to perform tumor segmentation using a classification model with patches as input and convolutional layers introduced early in the network to recognize low-level features. A patch was considered as positive if at least 50% of its pixels are tumoral. Five different patch sizes were tested to train a CNN dedicated to perform a binary classification between tumoral and non-tumoral labels. Using cross-validation on a DB containing 26 CECT volumes, it was noticed that 17x17 patches were the one that reached the best accuracy. The CNN achieved slightly better performance than classical ML techniques built on hand-crafted features (AdaBoost, RF and SVM).

#### Ben-Cohen et al. (2016)

**Ben-Cohen et al. (2016)**, used the famous VGG-16 architecture to construct his own network. The dense layers were converted into convolutional layers, especially the output classification layer that was removed and replaced by a 1x1 convolutional layer with 2 channels for a pixel-wise classification. The remaining layers were distributed in a FCN-8s-like architecture allowing to combine low-layer information with high-layer information. They also decided to provide not only one slice as input, but 3 successive slices (namely *2.5D input*) to help the network retrieve the 3D information. The output of the network corresponding to the segmentation of the middle slice. In order for the 3D information to be consistent from one patient to another, they interpolated the slices in the z-axis to have a fixed 1mm-spacing.

Data augmentation was used to generate natural-like images in order to artificially increase the size of the database. In their case, 4 different synthetic images were generated from one raw image by applying a scaling in the range [0.8, 1.2].

Two different databases were used to train their model, the SLIVER07 dataset was used to train the liver segmentation part, and an internal database of 20 patients, with only 68 liver-segmented slices, and 43 lesions-segmented slices, was used to train the second part of the architecture.

To evaluate their model, it was decided to consider the entire segmentation for the first stage by computing the Dice, whereas an object-based detection metric was implemented for the second stage, by analyzing the false positive rate per case (FPC).

The liver segmentation performance was evaluated on the 68 liver-annotated slices of their internal dataset, and they obtained a dice of 0.89 (note that only a slight improvement was obtained thanks to the 2.5D input: 0.88 vs 0.89).  
The accuracy of the lesion detection model was compared to both a patch-based approach where 17x17 patches were sampled within the liver, and a sparsity-based model. The FCN achieved better results than the other two methods reaching a FPC of 0.74.

When using the entire cascaded architecture, they realized that the automatic liver segmentation often doesn't include darker borders, and that those pixels could further be classified as FP, so the fully automatic segmentation leads to an improvement in FPC.

They also conducted some synthetic experiments and realized that the network mostly relies on the gray level differences.

#### Christ et al. (2016)

**Christ et al**, provided a cascaded architecture for the segmentation of both the liver and the potential lesions it contains. Their motivations were similar since they believed that the network in the first step will learn filters specific to the detection and the segmentation of the liver in an abdominal axial CT image, while considering that the liver ROI will help reduce the number of FP for the lesion segmentation task. The U-Net architecture introduced by **Ronneberger et al** which was used as a basis, was chosen for its ability to combine low and high-level features through so-called *skip-connections*.

In order to prove their hypothesis, they worked with the 3DIRCADb dataset containing 20 patients, where only the 15 patients presenting hepatic tumors were retained.

The images were pre-processed by first clipping the HU values in the range [-100, 400], followed by a histogram equalisation. Data augmentation was also used to increase the size of the database via rotations, flips and addition of gaussian noise, so they were able to train their models with more than 20k 2-D slices. In order not to train their networks from scratch, they used the pre-trained UNet from Ronneberger et al. trained on cell image data. A weighted loss function was implemented to combat the class imbalance during training, and they applied a 3D conditional random field (CRF) as a post-processing step on the obtained probability maps, to refine the results.

They however noticed that finding the appropriate weights for the CRF remains an open problem.

They obtained a Dice of 0.93, increased at 0.94 after applying the 3D CRF, for the liver segmentation, and a Dice of 0.56 for the lesions. They applied the same cross-validation learning on an internal database of 100 CT scans (Train: 60, Val: 20, Test: 20) and they reached a dice of 0.91 for the liver and 0.61 for the lesions.

#### Dou et al. (2016)

In 2016, the first fully 3D neural network dedicated to the segmentation of the liver was developed by **Dou et al**. Their 3D CNN network was built with large kernels, and had intermediate supervised layers to combat the gradient vanishing problem. In order to implement deeply-supervised mechanism, 3D Deconvolutional layers were built, to bridge coarse feature volumes, to dense probability ones. The SLIVER07 database containing 30 patients was evaluated, where 20 patients were used to train the networks, and 10 to test it. The global loss of the network incorporated an intermediate loss at each of those stages, and it was noticed that this mechanism allowed a better convergence than a classical 3D-CNN. A fully connected CRF model was also applied in the transverse plane to refine the obtained volume. The results obtained in a cross-validation manner confirm the gain brought by the deeply-supervised mechanism. On the test set, the results were on par with state-of-the-art “classical” methods (level-set, shape-based model…), but with a faster processing time.

#### Rafiei et al. (2018)

3D convolutional layers were also used by **Rafiei et al (2018)** to perform the liver segmentation task. Their architecture incorporated an encoding part where 3D layers only were used, whereas the decoding part consisted only of 2D layers. At each iteration, a block of 512x512x38 slices was fed to the network, and the decoding part was mainly intended to capture the 3D shape of the liver. The connection between the encoding part and the decoding part is done by custom skip connections, which first select the center slice in the 3D volume, and then crop it to meet the required size for the concatenation. To combat the classical problem of segmentation errors at the edges of the liver, they used a spatial-weighted-cross-entropy loss function, where pixels close to the liver border will be assigned a higher weight than those in the center. To help the generalization abilities of the network, a dropout layer was added just after the encoding part. Once the segmentation performed, they applied a fully connected CRF model only on the border pixels especially to reduce computation time.

Training and evaluation was completed on the MICCAI 2015 database containing 30 patients. Data augmentation was utilized to combat the lack of data, and 7 images were generated from one original image by rotation in the range [-30, 30]°.

They slightly improved the results obtained by **Heinrich et al [4]**, who used a multi-altas patch-based method (92.95 vs 93.52 on the average Dice), but with a high reduction of inference time (~1000s vs ~60s)

#### Sun et al. (2017)

**Sun et al.**, were the first in 2017 to use the multiphase information in a deep CNN for the lesion segmentation task by developing a so-called multi-channel fully convolutional networks(MC-FCN). To implement the architecture, they transformed the AlexNet to a FCN by converting fully connected layers into convolutional layers. They started by training their architecture with images from the different phases separately, then they realized that the weights were very different from one phase to another. So they built the MC-FCN so it can take 512x512x3 volumes as input, where each channel corresponds to a different phase. The 3 channels were separated early in the network by a so-called *slice layer*, before feature extraction was performed for each phase using the FCN. Finally, a feature fusion layer is employed to merge the information from the different phases, and the output is cropped to 512x512x2.

They tried to perform the segmentation of the lesions within the liver, but the network failed to converge, certainly due to the lack of sufficient training data, even when using a large number of iterations. To solve this issue, they first pre-trained the network to segment the liver in the image before fine-tuning it for the lesion segmentation.

They finally proved by the obtained results, an improvement of the segmentation accuracy when using the multiphase information.

#### Frid-Adar et al. (2017)

**Frid-Adar et al.** implemented a patch-based liver lesions detection method. Contrary to **Li et al. (2015)**, their main hypothesis was that non-lesion patches are different if they are located in the liver interior or in the liver boundary. They proposed to classify the patches into lesion and non-lesion categories, where lesion patches include lesion-boundary patches, and non-lesion patches contain both normal-interior patches and normal-boundary patches. An additional step was added to reduce the number of FP, with a binary classification CNN trained only on lesion patches and normal-boundary patches.

They also decided to use two different scales to capture fine (20x20) and global spatial information (50x50), before concatenating the obtained features from both sides at the end of the network. Patches were randomly sampled to respect class-balancing, and data augmentation was applied via random flips and rotations [5°, 130°, 300°]. Intensity was shifted so they all share the same mean intensity.

140 000 patches per class were used for the training, and 2 initialisation schemes were tested: a training from scratch, and a fine-tuning after training the network on the Cifar-10 dataset. The network was tested on a database containing 132 CT scans, with a total of 498 metastases. Images were first resampled to get a 0.71-fixed spacing, which corresponded to the finest available spacing in the dataset. They confirmed the improvement brought by the separation of the non-lesion patches into two different categories (TPR of 85.9 vs 80 when using a strict binary classification). They also found that fine-tuning from Cifar-10 was not as successful as training from scratch (82.8 vs 85.9 on the TPR) .

### LITS papers

In 2017, the Liver Tumor Segmentation (LITS) benchmark was created, where the goal was to have a standard database with a sufficient amount of humanly annotated data to assess the quality of a given model to perform one or multiple tasks, among which, the liver segmentation, the tumor segmentation and detection, and the tumor burden estimation.

The dataset contains a total of 201 CT scans, acquired in 7 different hospitals, and reviewed by 3 independent radiologists. The visible (131 volumes) and the hidden part (70 volumes) were separated so that each institution has the same proportion on both sides.

A high heterogeneity is present in the database, where from 0 up to 75 tumors are present per volume, and an average tumor-liver HU difference of 31.94 HU.

Two challenges have been launched by ISBI and MICCAI, aiming to segment this database. Some of the top-ranked studies that participated have been reviewed since they shared some common specifications.

#### Li et al. (2017)

**Li et al. (2017)** combined both 2D and 3D information to get the final classification. They first trained a Residual Network to get a coarse liver segmentation. Then, a 2.5D DenseUNet (densely connected path with UNet-like connections) architecture was developed to get a 64-channels features map corresponding to the 2D features. A pixel-wise classification step was added to get a pixel-wise 2D classification. The binary maps were stacked to a 12-slices 3D block, and combined as input with the original 3D volume to obtain a 64-channels features map encoding the 3D inter-slice information.

The 64-channels feature maps encoding both the 2D and the 3D information were merged so that the last layer of their architecture could extract the so-called *hybrid features*, before a pixel-wise probability is generated.

For the coarse segmentation step, they decided to resample the images so they have a fixed spacing of 0.69x0.69x1.0 mm³, but for lesion segmentation, the network was trained with images at their original resolution.  
From their experiments, they realized that the DenseUNet performed better than simple DenseNet, confirming the importance of long skip-connections. Interestingly, the 2D version of this DenseUNet allowed a better learning behaviour, when compared to the 3D version, and this hypothesis was confirmed by the prediction results (improvement of 0.89 points for the Dice per case, and 0.3 points for the Dice global). They also proved that the hybrid features help the network improve the obtained results.

Using the final architecture, they got a **Dice per case of 0.72 and a Dice global of 0.82**.

#### Han et al. (2017)

**Han et al. (2017)** decided to combine the residual network and the UNet, by building a resnet-like network with UNet skip connections between encoding and decoding parts.

They also provided 5 stacked slices as input in a 2.5D fashion, so the network can rely on the information provided by adjacent slices.

The first step of their model consisted of getting a coarse liver segmentation, before being refined in the second step which also segmented the lesions it may contain.

A weighted cross-entropy loss function with empirically chosen weights was implemented to train the network. The entire CT slices resampled to a fixed resolution of 1x1x2.5 mm³ were used to train the liver segmentation, whereas only liver slices with their original resolution were utilized for the lesions segmentation step.

During inference, the input volume is resampled to 1x1x2.5 mm³, then the liversegmentationis performed before getting the largest 3D connected component. The original resolution is then used to perform the lesion segmentation in order to get finer details from the liver and its lesions. A post-processing step was employed so that connected components considered as lesions will be removed if the maximal probability value is less than a given threshold.

The entire process allowed them to reach a **Dice per case of 0.67.**

#### Yuan et al. (2017)

**Yuan et al. (2017)** proposed a hierarchical convolution-deconvolution neural network to carry out the segmentation of both the liver and the lesions.

The first step of their architecture consisted of getting a coarse liver segmentation on the entire 3D volume, before the second step refined the probability map, and the final one executed the lesions segmentation.

The same architecture was used for the three different steps, where 2.5D input was chosen, and training was performed with the Jaccard distance as loss function. This function takes advantage from the fact that it maximizes directly the performance that is computed from the output of the network. The architecture is composed of a so-called *convolutional path* where the information is compressed, and a *deconvolutional path*, where the full image resolution is recovered. Dropout layers were also used at the end of the convolution path, and the other just before the last deconvolution layer.

During the first step of their cascade, images were down-scaled to 128x128 and the volumes were resampled with a slice thickness of 3mm. The training process of this specific step was made only from liver slices plus 5 slices above and below it. For the inference, the segmentation was performed on each slice, and pixels were considered as being part of the liver after applying a thresholding of 0.5 on the probability map, and after extracting the largest connected component.

To train the second step, which consisted of the refinement of the previously obtained liver area, the original volume was firstly resampled to get a fixed slice-thickness of 2 mm. The liver 3D bounding box was extracted from the annotated liver, with 10 more voxels in the 3 directions, and the axial dimension was adjusted to 256x256 pixels.

The inference was achieved by extracting the slices from the predicted liver area, and applying the prediction with the second network. The same post-processing step was implemented to conserve the 3D consistency.

In order to train the last network, the liver VOI was extracted as previously, but by keeping the original resolution to avoid losing small details. The training was completed using only lesion slices, and an additional image was added beside the original image, where intensity difference between lesion and liver voxels was enhanced via 3D regional histogram equalization.  
During the inference, all activations outside the predicted liver were removed.

Their method achieved average DSCs of **0.96** for liver segmentation (Dice global of 0.967), **0.66** for tumor segmentation (Dice global of 0.82), and a root mean square error (RMSE) of **0.02** for tumor burden estimation.

#### Bellver et al. (2017)

The cascade paradigm was also the key element in the study of **Bellver et al. (2017)**.

The main architecture they used was the DRIU (deep retinal image understanding), consisting of VGG16 backbone with last Dense Layers replaced by a convolutional layer.

The network was pre-trained on the ImageNet dataset, before fine-tuned with CT images having intensity clipped to the range [-150, 250] and further normalized.

To train the network, the weighted binary cross entropy (BCE) was implemented in order to deal with class balancing. When training the network responsible for the segmentation of the lesion, they decided to suppress the loss outside the liver area, and considered only pixels inside the liver region when computing the weights of their loss function. Each one of the networks were trained with 2.5D input to exploit the 3D information.

After executing both the liver and lesions segmentation, they developed a lesion detector to reduce the number of FP. A patch-based approach was used for the training of the detector, with patches of 80x80 pixels sampled through the liver (having at least 25% overlap with the liver), and classified as positive if at least 50 pixels of the region belong to a lesion.

The detector was trained using a ResNet-50-like architecture, pre-trained again on ImageNet, and that was customized by replacing the last classification layer by a binary-classification layer (healthy/non-healthy).

A post-processing invoking a fully-connected 3D CRF model was further called to refine the obtained regions.

Their results proved that the detector helps the network to reduce the number of FP, and also confirmed that learning to segment the lesions only from the liver area helps the network.

With their method, they obtained a Dice per case of **0.59** on the test set of LITS.

#### Chlebus et al. (2018)

**Chlebus et al. (2018)** submitted their results to both ISBI and MICCAI challenges while targeting only the lesion segmentation. In their study they not only described the methods used to generate their segmentation, but they challenged the LITS dataset by comparing it with manual annotations from another expert.

3 different U-Net-like architectures (axial, sagittal, coronal) on 4 resolution levels with additional short skip connections dedicated to the segmentation of tumors only were trained on the LITS database. The loss function considered only the voxels belonging to the liver. In order to reduce the number of FP, an additional step was added where hand-crafted features were computed on the 3D objects obtained after the segmentation step. In order to assess the performance of the neural network, a human (MTRA: medical-technical radiology assistant) was asked to segment the LITS dataset. The dice per case between MTRA and LITS was 0.70. The neural network reached a DSC of 0.51 on the LITS (4.6 FP/case). After using the classifier, the Dice was improved to 0.58, and the number of FP was reduced to 0.7/case. They finally reached a dice per case of **0.68** for the lesion and **0.96** for the liver.  
Among the several hand-crafted features, the most relevant one for detecting the FP were shape-based features as long the distance from the object to the boundary.

#### Kaluva et al. (2018)

**Kaluva et al. (2018)** proposed to build their model with the DenseNet architecture as a basis. The DenseNet is constructed such that a given layer *l* is connected to each of its predecessors *l-1, l-2, … 0* in a concatenated manner. The number of feature maps at a given stage is known as *growth rate*. The number of feature maps in the classical DenseNet grow linearly with the depth. After each dense block, a Transition Down (TD) layer is introduced to reduce the spatial dimensionality of the feature maps.

Their cascaded model consisted of 2 steps, one specified on the liver segmentation, and the second one on the lesions segmentation.

To train the liver segmentation network, the original CT images were downsampled to 256x256 pixels before normalizing HU intensities clamped to the range [-100, 300].

No downsampling was performed when training the lesions segmentation network, and comparing to the first network, 3 differents HU windows ([0, 100], [-100, 200] and [-100, 400]) were applied on the original image and combined to get a 3-channels input.

To deal with class imbalancing, the first network was trained only on liver slices (with 10 slices above and below it), whereas the second one was trained only on lesion slices (plus 5 slices above and below).

They also decided to connect the input and the output of each Dense block in the downsampling part, in a ResNet-like fashion.

Their first network was trained thanks to a spatial-weighted-cross-entropy as loss function with a higher loss at the edges of the liver, relative to the interior. The same loss function was used for the second network, with a higher weight given to the tumor region, and was combined with the dice loss.

As post-processing, they simply decided to mask all tumor activations that were located outside the predicted liver.

On the 70 volumes of the hidden set (LIST) the average Dice for the liver was **0.91** and **0.72** for the lesions.

#### Bi et al. (2017)

**Bi et al. (2017)** also developed a cascaded architecture where a first ResNet segment both the liver and the lesion, and the second one combining the original image with the two obtained probability maps to produce the final segmentation map.

Images were preprocessed by first clipping the intensities to the range [-160, 240], before applying a min-max normalization. To train their networks, they used a total of more than 8,000 randomized slices, where half of them presented both liver and lesions, and the other half where the liver was not present at all.

The original network was pre-trained on the ImageNet dataset for 100 epochs, before being fine-tuned with augmented data from LITS (random scaling, crops and flips).

During inference, a multiscale-strategy was used by resizing the original image to different scales from 512x512 to 640x640 with a step of 32 at each stage, and averaging the obtained outputs.

The use of the multiscale-strategy slightly improves the results obtained using only their cascaded ResNet. A slight improvement was also notable between their cascaded version and the classical ResNet. They finally obtained a Dice of **0.64** for the segmentation of the lesions.

#### Vorontsov et al. (2018)

**Voronstov et al. (2018)** decided also to use a cascaded architecture where two U-Net-like networks were combined, but they decided to build a model allowing an end-to-end training of the entire cascade.

The first network took as input an axial slice and predicted the probability of each pixel to belong to the liver, and the second one took as input both the original slice and the output of the first network. The latter implemented short skip connections in a resnet-fashion to propagate the liver localisation information to all the layers.

Images were pre-processed by dividing the input intensities to 255, and clipping the remaining values to the range [-2, 2], and only liver slices were used to train the network with the dice loss as loss function.

Data augmentation was applied via flips, rotations, zoom and elastic deformations.

The network was pre-trained on the down-sampled images (256x256) then fine-tuned on the full resolution slices.

To improve the segmentation, a classifier was added at the end of each network, aiming to introduce a so-called *cross-slice context*. The segmentation was performed on 3 adjacent slices, and the output features were combined by a classifier to predict the middle-slice segmentation.

During inference, 3 different networks were associated via a voting strategy, where each one provided a prediction corresponding to the average of 4 oriented images after flipping the original one.

The final liver segmentation corresponded to the largest connected component, and lesions were kept when belonging to a dilated version of the predicted liver. This prevents missing lesions when the liver is slightly under-segmented.

They finally obtained a Dice per case of **0.66** for the lesions and of **0.95** for the liver.

#### Bilic et al. (2019) LITS Review

The majority of the papers that were submitted for these 2 challenges share some common properties, and they were reviewed by the organizers themselves in **Bilic et al. (2019)**.

They started with a reminder about the composition of the database, where the 201 total CT scans were acquired in 7 different hospitals, and reviewed by 3 independent radiologists. The visible (131 volumes) and the hidden part (70 volumes) were separated so that each institution has the same proportion on both sides.

A high heterogeneity is present in the database, where from 0 up to 75 tumors are present per volume, and an average tumor-liver HU difference of 31.94 HU.

Regarding the submitted methods, a U-Net-like network was often chosen as a basis, with modifications by incorporating residual connections, or adjusting the input resolution. No direct 3D methods on the original images, but some methods used 3D convolutions for the lesions segmentation task, with smaller input resolution images. Some other techniques used 2.5 input.

Images were pre-processed by clipping the HU values, but without a clear consensus regarding the range to apply. The images are then often normalized and resampled to a specific resolution. The size of the dataset is often increased after the application of data augmentation, which corresponds most of the time to standard geometrical transformations.

The most commonly used loss functions were the weighted cross entropy loss function, having the advantage of solving the class imbalance problem. The Dice loss and the Jaccard loss were also used and are useful because they allow the network updates its weights to reach the best scores regarding the final used metric.  
After performing the segmentations, the vast majority of the teams used post-processing techniques to refine their results. Among these techniques, they reported the extraction of connected components, the removal of lesions activation outside of the liver, the fill of erroneous holes in the lesions shape via morphological operations, the application of connected CRF, or the use of a classifier (such as Random Forest) to detect the FP.

Ensemble learning was also applied by the top-ranked methods, either via axis-related networks, the combination of different rescaled input data, or the use of different networks trained in a CV-manner.

Regarding the results, they realized that the best tumor-segmentation results were obtained for large lesions, and within a specific lesion-liver HU difference interval (a difference between -10 and -60).

The top-ranked methods used some 3D approaches in their architecture **(Chlebus et al, Li et al),** showing perspectives to capture the whole volume context in the future.

The introduction of such a standard database led to a rise in the interest towards automatic liver and lesions segmentation. The organizers are planning to relaunch the same type of competition in the future, by particularly providing multiple ground truth (instead of only 1 for LITS) and potentially splitting the lesion segmentation task into large and small lesions, since current methods still struggle in segmenting the small lesions.

### New hybrid techniques

#### Jin et al. (2018)

**Jin et al. (2018)** proposed a network that integrated both U-Net and attention residual mechanism to proceed the segmentation of both the liver and the lesions.

The residual attention mechanism has been introduced in 2017 by [**Wang et al.**](https://arxiv.org/pdf/1704.06904.pdf) to perform image classification, with the idea that attention mechanism can help the network focusing on specific parts of the image.

The study from Jin et al. was the first to use the attention mechanism for semantic segmentation purpose.

Their model contained 3 stages, the first one to get a coarse liver segmentation from 2D images, the second one to refine the obtained volume of interest using a 3D architecture, and the last one to segment tumors from the precise liver VOI.

Each network was built according to the U-Net shape but using residual blocks as key components. The combination of both residual blocks and attention mechanism allows the network to learn on deep architectures, but with a focus on localizations that are relevant for the requested task.

The images were pre-processed by first clipping the HU intensities to the range [-100, 200], before applying a zero-mean and a min-max normalization. The network was then trained using the dice loss as loss function.

In the first stage, a set of ⅔ liver images and ⅓ randomly picked slices were down-sampled to 256x256 pixels to train the network. The coarse liver segmentation was obtained through the extraction of the largest connected component.

Large 3D patches of 224x224x32 slices were extracted from both the lesions and non-lesions area and were provided to train the second network.

Finally, for the lesions segmentation task, they tested 3 different patch-sizes (32x32x32, 64x64x32 and 128x128x32) before realizing that the larger the patch was, the richer the extracted information will be. The patches were randomly sampled within the liver.

Since the model was trained on 5 CV-folds, they implemented a voting strategy from the five resulting networks to get the final segmentation.

They obtained a Dice per case of **0.59** for the lesion segmentation task on the hidden set of LITS, outperforming a lot of 2D-based methods, but still far from the top-ranked teams.