Lymphoma segmentation from 3D PET-CT images using a deep evidential network

Ling Huang[a,b], Su Ruan[b], Pierre Decazes[c], Thierry Denœux[a,d]

[a] *Heudiasyc, CNRS, Université de technologie de Compiègne, Compiègne, France*
[b] *Quantif, LITIS, University of Rouen Normandy, Rouen, France*
[c] *Department of Nuclear Medicine, Henri Becquerel Cancer Center, Rouen, France*
[d] *Institut universitaire de France, Paris, France*

Abstract

An automatic evidential segmentation method based on Dempster-Shafer theory and deep learning is proposed to segment lymphomas from three-dimensional Positron Emission Tomography (PET) and Computed Tomography (CT) images. The architecture is composed of a deep feature-extraction module and an evidential layer. The feature extraction module uses an encoder-decoder framework to extract semantic feature vectors from 3D inputs. The evidential layer then uses prototypes in the feature space to compute a belief function at each voxel quantifying the uncertainty about the presence or absence of a lymphoma at this location. Two evidential layers are compared, based on different ways of using distances to prototypes for computing mass functions. The whole model is trained end-to-end by minimizing the Dice loss function. The proposed combination of deep feature extraction and evidential segmentation is shown to outperform the baseline UNet model as well as three other state-of-the-art models on a dataset of 173 patients.

Keywords: medical image analysis, Dempster-Shafer theory, evidence theory, belief functions, uncertainty quantification, deep learning

1. Introduction

Positron Emission Tomography - Computed Tomography (PET-CT) scanning is an effective imaging tool for lymphoma segmentation with application to clinical diagnosis and radiotherapy planning. The standardized uptake value (SUV), defined as the measured activity normalized for body weight and injected dose to remove variability in image intensity between patients, is widely used to locate and segment lymphomas thanks to its high sensitivity and specificity to the metabolic activity of tumors [1]. However, PET images have a low resolution and suffer from the partial volume effect blurring the contours of objects [2]. For that reason, CT images are usually used jointly with PET images because of their anatomical feature-representation capability and high resolution. Figure 1 shows 3D PET-CT views of a lymphoma patient. The lymphomas are marked in black as well as the brain and the bladder. As we can see from this figure, lymphomas vary in intensity distribution, shape, type, and number.

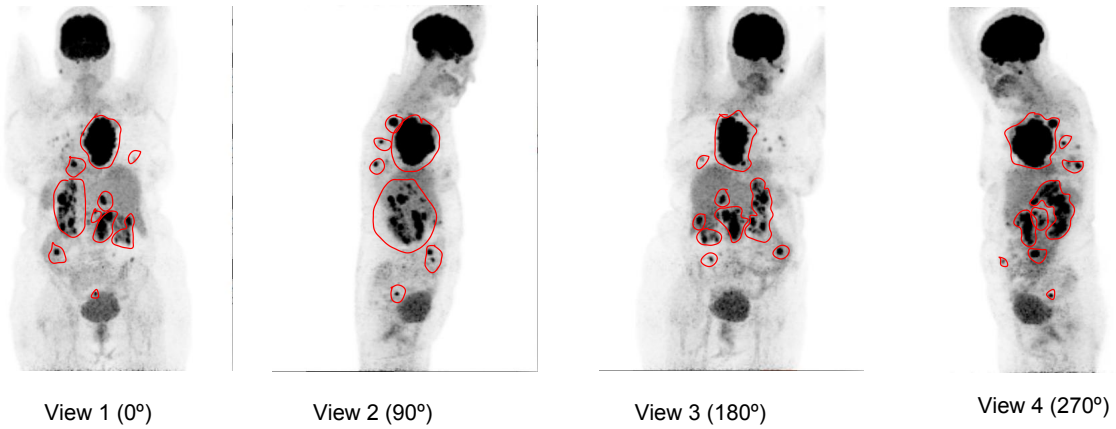


Figure 1: Example of a patient with lymphomas in 3D PET-CT views. The lymphomas are the dark areas circled in red.

Approaches to lymphoma segmentation. Techniques for lymphoma segmentation can be divided into three classes: SUV-based, region-growing-based and deep learning-based methods. For PET images, it is common to segment lymphomas with a set of fixed SUV thresholds. The so-called SUV-based methods [3][4] are fast but lack flexibility in boundary delineation and requires domain knowledge to locate the region of interest. Region-growing-based methods [5][6] have been proposed to optimize boundary delineation by taking texture and shape information into account. By setting the specific growing function and stopping condition, the tumor region grows step by step until it reaches the stopping condition. However, those methods still need clinicians to locate the seeds for region growing [5] and they are time-consuming, especially when applied to 3D images. Lymphoma segmentation with deep learning has become a popular research topic thanks to its high feature representation ability [7][8].

Deep-learning-based methods. Long et al. [9] were the first authors to show that a fully convolutional network (FCN) could be trained end-to-end for semantic segmentation, exceeding the state-of-the-art when the paper was published. UNet [10], a successful modification and extension of FCN, has become the most popular model for medical image segmentation in recent years. Driven by different tasks and datasets, several extended and optimized variants of UNet have been proposed for medical image segmentation, including VNet [11], SegResNet [12], and nnUNet [13]. VNet is a variant of UNet that introduces short residual connections at each stage. Compared with UNet, SegResNet contains an additional variational autoencoder branch. Finally, nnUNet is more flexible than UNet in three aspects: (1) residual connection in convolution blocks, (2) anisotropic kernel sizes and strides in each layer, and (3) deep supervision heads. Deep learning has been applied to lymphoma segmentation, yielding promising results. In [7], Li et al. proposed a DenseX-Net-based lymphoma segmentation model with a two-flow architecture for 3D PET-CT images: a segmentation flow (DenseU-Net) for lymphoma segmentation and a reconstruction flow (encoder-decoder) for learning semantic representation of different lymphomas. In [8], Hu et al. introduced

a multi-source fusion model for lymphoma segmentation with PET images. First, three 2D and one 3D segmentation models were trained with three orthogonal views and one 3D image, respectively. The four segmentation maps were then fused by a convolutional layer to get a final result. In [14], Blanc-Durand et al. proposed a nnUNet-based lymphoma segmentation network with additional validation of total metabolic tumor volume for 3D PET-CT images. In [15], Huang et al. proposed to fuse the outputs of two UNets trained on CT and PET data, using Dempster's rule of combination [16], a combination operator of Dempster-Shafer theory (DST) (see Section 2 below). However, the outputs of the UNets were probabilities and this approach did not harness the full power of DST.

Uncertainty. In spite of the excellent performance of deep learning methods, the issue of quantifying prediction uncertainty remains [17]. This uncertainty can be classified into three types: distribution, model, and data uncertainty. Distribution uncertainty is caused by training-test distribution mismatch (dataset shift) [18]. Model uncertainty arises from limited training set size and model misspecification [19][20][21]. Finally, sources of data uncertainty include class overlap, label noise, and homo or hetero-scedastic noise [22]. Because of the limitations of medical imaging and labeling technology, as well as the need to use a large nonlinear parametric segmentation model, PET-CT image segmentation results are particularly tainted with uncertainty, which limits the reliability of the segmentation. Figure 2 shows examples of PET and CT image slices for one patient with lymphomas. As can be seen, lymphomas in PET images usually correspond to the brightest pixels, but organs such as the brain and bladder are also located in bright pixel areas, which may result in segmentation errors. Moreover, lymphoma boundaries are blurred, which makes it hard to delineate lymphomas precisely.

Approaches to uncertainty modeling. Early approaches to uncertainty quantification in machine learning were based on Bayesian theory [23][24]. The popularity of deep learning models has revived research of model uncertainty estimation and has given rise to specific methods such as variational dropout [25][26]. In this paper, we explore a different approach based on DST [27][16] [28], a theoretical framework for reasoning with imperfect (uncertain, imprecise, partial) information. DST was first introduced by Dempster [27] and Shafer [16] and was further popularized and developed by Smets [29]. Applications in machine learning were first introduced by Denœux [30, 31, 32]. DST is based on the representation of elementary items of evidence by belief functions, and their combination by a specific operator called Dempster's rule of combination. In recent years, DST has generated considerable interest and has had great success in various fields, including information fusion [33][34][35], classification [36][37][38], clustering [28][39][40], and image segmentation [41][42][43].

In this paper[1], we propose a 3D PET-CT diffuse large B-cell lymphoma segmentation model based on DST and deep learning, which not only focuses on lymphoma segmentation accuracy but also on uncertainty quantification using belief functions. The proposed

[1]This paper is an extended version of the short paper presented at the 6th International Conference on Belief Functions (BELIEF 2021) [44].

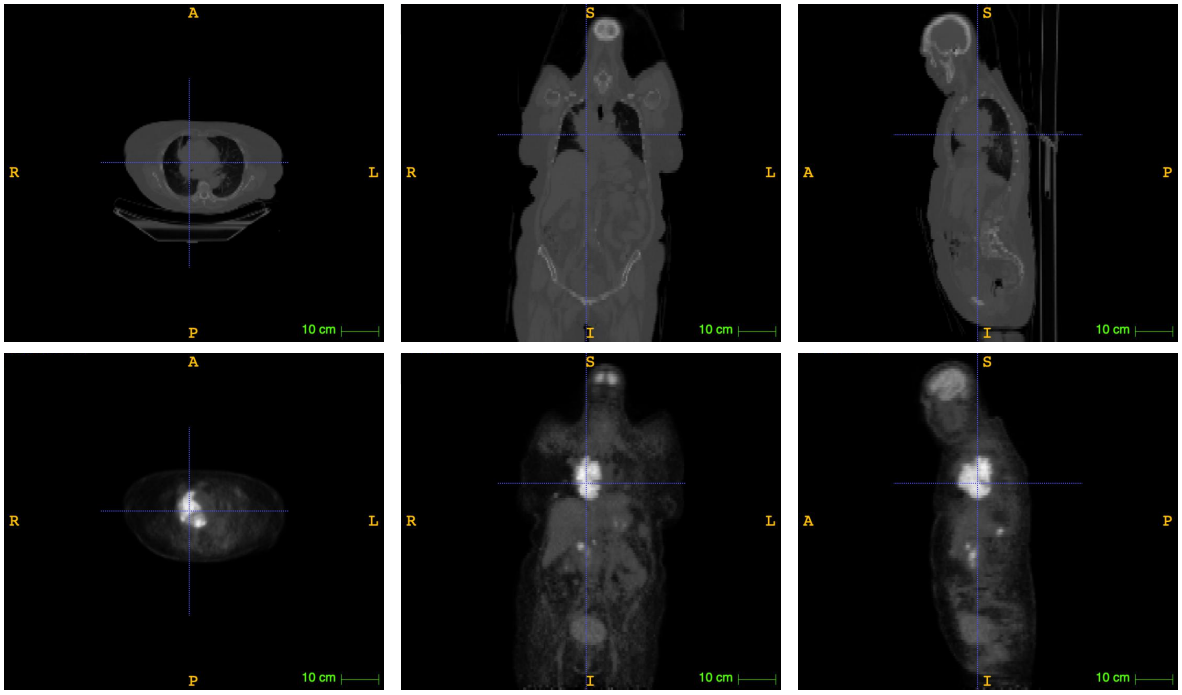


Figure 2: Example of a patient with lymphomas. The first and second rows show, respectively, PET and CT slices for one patient in axial, sagittal and coronal views.

segmentation model is composed of a UNet module for feature extraction and an evidential segmentation module for uncertainty quantification and decision-making. End-to-end learning is performed by minimizing the Dice loss function.

The rest of the paper is organized as follows. The main concepts of DST are first recalled in Section 2, and two approaches for computing belief functions in classification tasks are described in Section 3. The proposed model is then introduced in Section 4, and experimental results are reported in Section 5. Finally, Section 6 concludes the paper.

2. Dempster-Shafer theory

In this section, we first recall some necessary notations and definitions regarding DST. Let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ be a finite set of all possible answers some question, called the *frame of discernment*. Evidence about the question of interest can be represented by a *mass function* m, defined as a mapping from the power set 2^Ω to $[0, 1]$ such that

$$\sum_{A \subseteq \Omega} m(A) = 1 \tag{1}$$

and $m(\emptyset) = 0$, where $\emptyset$ denotes the empty set. Subsets $A \subseteq \Omega$ such $m(A) > 0$ are called the *focal sets* of m. Each mass $m(A)$ represents a share of a unit mass of belief allocated to focal set A, and which cannot be allocated to any strict subset of A. The mass $m(\Omega)$ allocated to the whole frame can be seen as a degree of ignorance. Full ignorance is represented by

the *vacuous* mass function $m_?$ verifying $m_?(\Omega) = 1$. A mass function is said to be *Bayesian* if its focal sets are singletons, and *logical* if it has only one focal set.

Discounting. Let m be a mass function on Ω and s a coefficient in $[0, 1]$. The *discounting* operation [16] with discount rate $1 - s$ transforms m into a weaker, less informative mass function defined as follows:

$$^s m = s\, m + (1 - s)\, m_?. \tag{2}$$

As shown in [45], coefficient s can be interpreted as a degree of belief that the source of information providing mass function m is reliable.

Simple mass functions. A mass function m is said to be *simple* if it can be obtained by discounting a logical mass function; it thus has the following form:

$$m(A) = s, \quad m(\Omega) = 1 - s, \tag{3}$$

for some $A \subset \Omega$ such that $A \neq \emptyset$ and some $s \in [0, 1]$, called the *degree of support* in A. The quantity $w = -\ln(1 - s)$ is called the *weight of evidence* associated to m [16, page 77]. In the following, a simple mass function with focal set A and weight of evidence w will be denoted as A^w.

Belief and plausibility. Given a mass function m, *belief* and *plausibility* functions are defined, respectively, as follows:

$$Bel(A) = \sum_{B \subseteq A} m(B) \tag{4}$$

and

$$Pl(A) = \sum_{B \cap A \neq \emptyset} m(B) = 1 - Bel(A^c), \tag{5}$$

for all $A \subseteq \Omega$, where A^c denotes the complement of A. The quantity $Bel(A)$ can be interpreted as a degree of support for A, while $Pl(A)$ can be interpreted as a measure of lack of support for the complement of A.

Dempster's rule. Two mass functions m_1 and m_2 derived from two independent items of evidence can be combined by considering each pair of a focal set B of m_1 and a focal set C of m_2, and assigning the product $m_1(B)m_2(C)$ to the intersection $B \cap C$. A normalization step is then necessary to ensure that the mass of the empty set is equal to zero. This operation, called *Dempster's rule of combination* [16] and denoted as $\oplus$, is formally defined by $(m_1 \oplus m_2)(\emptyset) = 0$ and

$$(m_1 \oplus m_2)(A) = \frac{1}{1 - \kappa} \sum_{B \cap C = A} m_1(B)m_2(C), \tag{6}$$

for all $A \subseteq \Omega, A \neq \emptyset$, where κ represents the *degree of conflict* between m_1 and m_2 equal to

$$\kappa = \sum_{B \cap C = \emptyset} m_1(B)m_2(C). \tag{7}$$

The combined mass $m_1 \oplus m_2$ is called the *orthogonal sum* of m_1 and m_2. It can easily be checked that the orthogonal sum of two simple mass functions A^{w_1} and A^{w_2} with the same focal set A is the simple mass function $A^{w_1+w_2}$: Dempster's rule thus adds up weights of evidence.

Decision-making. After aggregating all the available evidence in the form of a mass function, it is often necessary to make a final decision. Decision-making based on belief functions for classification tasks has been studied in [46], and, more recently, by Ma and Denœux in [47]. The reader is referred to Ref. [48] for a recent review of decision methods based on belief functions. Here, we briefly introduce the approach used in this paper. Consider a classification task with K classes in the set $\Omega = \{\omega_1, \ldots, \omega_K\}$. Assume that the utility of selecting the correct class is 1, and the utility of an error is 0. As shown in [46], the lower and upper expected utilities of selecting class ω_k are then, respectively, $Bel(\{\omega_k\})$ and $Pl(\{\omega_k\})$. A pessimistic decision-maker (DM) maximizing the lower expected utility will then select the class with the highest degree of belief, while an optimistic DM minimizing the upper expected utility will select the most plausible class. Alternatively, the Hurwicz criterion consists in maximizing a weighted sum of the lower and upper expected utility. In the decision context, we then select the class ω_k such that $(1-\xi)Bel(\{\omega_k\}) + \xi Pl(\{\omega_k\})$ is maximum, where ξ is an optimism index. Another approach, advocated by Smets in the Transferable Belief Model [45], is to base decisions on the pignistic probability distribution, defined as

$$p_m(\omega) = \sum_{\{A \subseteq \Omega : \omega \in A\}} \frac{m(A)}{|A|} \tag{8}$$

for all $\omega \in \Omega$.

3. Evidential classifiers

In this section, we review two methods for designing classifiers that output mass functions, referred to as *evidential classifiers*. The evidential neural network (ENN) classifier introduced in [31] is first recalled in Section 3.1. A new model based on the interpretation of a radial basis function (RBF) network as combining of simple mass functions by Dempster's rule, inspired by [49], is then described in Section 3.2. The two models are compared experimentally in Section 3.3.

3.1. Evidential neural network

In [31], Denœux proposed the ENN classifier, in which mass functions are computed based on distances to prototypes. The basic idea is to consider each prototype as a piece of evidence, which is discounted based on its distance to the input vector. The evidence from different prototypes is then pooled by Dempster's rule (6). We provide a brief introduction to the ENN model in this section.

The ENN classifier is composed on an input layer of H neurons (where H is the dimension of input space), two hidden layers and an output layer (Figure 3). The first input layer is

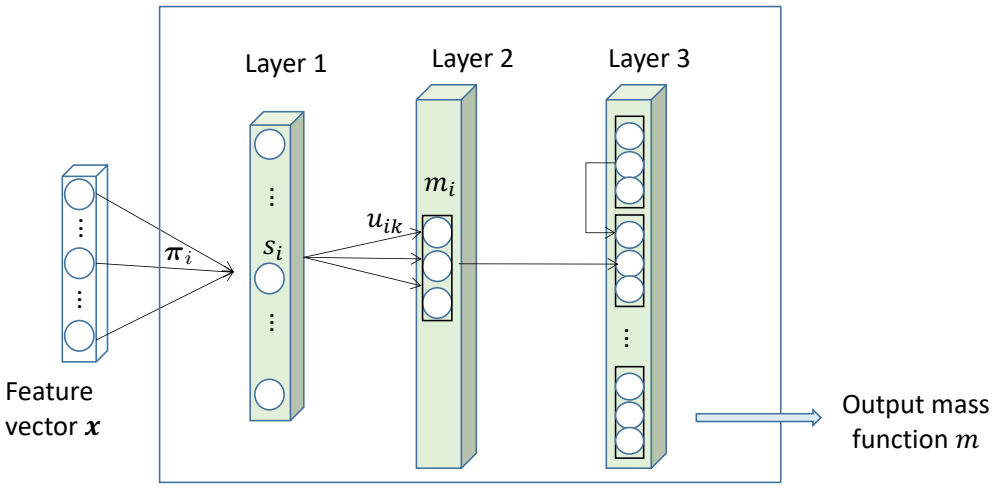


Figure 3: Evidential neural network.

composed of I units, whose weights vectors are prototypes $\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_I$ in input space. The activation of unit i in the prototype layer is

$$s_i = \alpha_i \exp(-\gamma_i d_i^2), \tag{9}$$

where $d_i = \|\boldsymbol{x} - \boldsymbol{\pi}_i\|$ is the Euclidean distance between input vector $\boldsymbol{x}$ and prototype $\boldsymbol{\pi}_i$, $\gamma_i > 0$ is a scale parameter, and $\alpha_i \in [0, 1]$ is an additional parameter.

The second hidden layer computes mass functions m_i representing the evidence of each prototype $\boldsymbol{\pi}_i$, using the following equations:

$$m_i(\{\omega_k\}) = u_{ik} s_i, \quad k = 1, ..., K \tag{10a}$$

$$m_i(\Omega) = 1 - s_i, \tag{10b}$$

where u_{ik} is the membership degree of prototype i to class ω_k, and $\sum_{k=1}^{K} u_{ik} = 1$. Mass function m_i can thus be seen as a discounted Bayesian mass function, with discount rate $1 - s_i$; its focal sets are singletons and Ω. The mass assigned to Ω increases with the distance between $\boldsymbol{x}$ and $\boldsymbol{\pi}_i$. Finally, the third layer combines the I mass functions $m_1, \ldots, m_I$ using Dempster's rule (6). The output mass function $m = \bigoplus_{i=1}^{I} m_i$ is a discounted Bayesian mass function that summarizes the evidence of the I prototypes. Because the focal sets of m are singletons and Ω, the class with the highest degree of belief also has the highest plausibility and pignistic probability: consequently, the decision rules recalled in Section 2 are equivalent in this case.

Let $\boldsymbol{\theta}$ denote the vector of all network parameters, composed of the I prototypes $\boldsymbol{\pi}_i$, their parameters γ_i and α_i, and their membership degrees u_{ik}, $k = 1, \ldots, K$. In [31], it was proposed to learn these parameters by minimizing the regularized sum-of-squares loss function

$$L_{SS}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \sum_{k=1}^{K} (p_{nk} - y_{nk})^2 + \lambda \sum_{i=1}^{I} \alpha_i, \tag{11}$$

where p_{nk} is the pignistic probability of class ω_k for instance n, N is the number of training instances, and $y_{nk} = 1$ if the true class of instance n is ω_k, and $y_{nk} = 0$ otherwise. The second term on the right-hand side of (11) is a regularization term, and λ is hyperparameter that can be tuned by cross-validation.

The idea of applying the above model to features extracted by a convolutional neural network (CNN) was first proposed by Tong et al. in [50]. In this approach, the ENN module becomes a "evidential layer", which is plugged into the output of a CNN instead of the usual softmax layer. The feature extraction and evidential modules are trained simultaneously. A similar approach was applied in [43] to semantic segmentation. In the next section, we present an alternative approach based on a radial basis function (RBF) network and weights of evidence.

3.2. Radial basis function network

As shown in [49], the calculations performed in the softmax layer of a feedforward neural network can be interpreted in terms of combination of evidence by Dempster's rule. The output class probabilities can be seen as normalized plausibilities according to an underlying belief function. Applying these ideas to a radial basis function (RBF) network, it is possible to derive an alternative evidential classifier with properties similar to those of the ENN model recalled in Section 3.1.

Consider an RBF network with I prototype (hidden) units. The activation of hidden unit i is

$$s_i = \exp(-\gamma_i d_i^2), \tag{12}$$

where, as before, $d_i = \|\boldsymbol{x} - \boldsymbol{\pi}_i\|$ is the Euclidean distance between input vector $\boldsymbol{x}$ and prototype $\boldsymbol{\pi}_i$, and $\gamma_i > 0$ is a scale parameter. For the application considered in this paper, we only need to consider the case of binary classification with $K = 2$ and $\Omega = \{\omega_1, \omega_2\}$. (The case where $K > 2$ is also analyzed in [49]). Let v_i be the weight of the connection between hidden unit i and the output unit, and let $w_i = s_i v_i$ be the product of the output of unit i and weight v_i. The quantities w_i can be interpreted as weights of evidence for class ω_1 or ω_2, depending on the sign of v_i:

- If $v_i \geq 0$, w_i a weight of evidence for class ω_1;
- If $v_i < 0$, $-w_i$ is a weight of evidence for class ω_2.

To each prototype i can, thus, be associated the following simple mass function:

$$m_i = \{\omega_1\}^{w_i^+} \oplus \{\omega_2\}^{w_i^-},$$

where $w_i^+ = \max(0, w_i)$ and $w_i^- = -\min(0, w_i)$ denote, respectively, the positive and negative parts of w_i. Combining the evidence of all prototypes in favor of ω_1 or ω_2 by Dempster's rule, we get the mass function

$$m = \bigoplus_{i=1}^{I} m_i = \{\omega_1\}^{w^+} \oplus \{\omega_2\}^{w^-}, \tag{13}$$

with $w^+ = \sum_{i=1}^{I} w_i^+$ and $w^- = \sum_{i=1}^{I} w_i^-$. In [49], the normalized plausibility of ω_1 corresponding to mass function m was shown to have the following expression:

$$p(\omega_1) = \frac{Pl(\{\omega_1\})}{Pl(\{\omega_1\}) + Pl(\{\omega_2\})} = \frac{1}{1 + \exp(-\sum_{i=1}^{I} v_i s_i)}, \tag{14}$$

i.e., it is the output of a unit with a logistic activation function. When training an RBF network with a logistic output unit, we thus actually combine evidence from each of the prototypes, but the combined mass function remains latent. In [49], mass function m defined by (13) was shown to have the following expression:

$$m(\{\omega_1\}) = \frac{[1 - \exp(-w^+)]\exp(-w^-)}{1 - \kappa} \tag{15a}$$

$$m(\{\omega_2\}) = \frac{[1 - \exp(-w^-)]\exp(-w^+)}{1 - \kappa} \tag{15b}$$

$$m(\Omega) = \frac{\exp(-w^+ - w^-)}{1 - \kappa} = \frac{\exp(-\sum_{i=1}^{I} |w_i|)}{1 - \kappa}, \tag{15c}$$

where

$$\kappa = [1 - \exp(-w^+)][1 - \exp(-w^-)] \tag{15d}$$

is the degree of conflict between mass functions $\{\omega_1\}^{w^+}$ and $\{\omega_2\}^{w^-}$.

In the approach, we thus simply need to train a standard RBF network with I prototype layers and one output unit with a logistic activation function, by minimizing a loss function such as, e.g., the regularized cross-entropy loss

$$L_{CE}(\boldsymbol{\theta}) = -\sum_{n=1}^{N} (y_n \log p_n + (1 - y_n) \log(1 - p_n)) + \lambda \sum_{i=1}^{I} w_i^2, \tag{16}$$

where p_n is the normalized plausibility of class ω_1 computed from (14) for instance n, y_n is class label of instance n ($y_n = 1$ if the true class of instance n is ω_1, and $y_n = 0$ otherwise), and λ is a hyperparameter. We note that increasing λ has the effect of decreasing the weights of evidence and, thus, obtaining less informative mass functions.

3.3. Comparison between the two models

To compare the RBF model described in Section 3.2 with the ENN model recalled in Section 3.1, we consider the two-class dataset shown in Figure 4. The two classes are randomly distributed around half circles with Gaussian noise and are separated by a nonlinear boundary. A learning set of size $N = 300$ and a test set of size 1000 were generated from the same distribution.

An ENN and a RBF network were initialized with $I = 6$ prototypes generated by the k-means algorithm and were trained on the learning data. Figures 5a and 5b show, respectively, the test error rate and the mean uncertainty (defined as the average mass assigned to the frame Ω), as functions of hyperparameter λ in (11) and (16), for 10 different runs of both

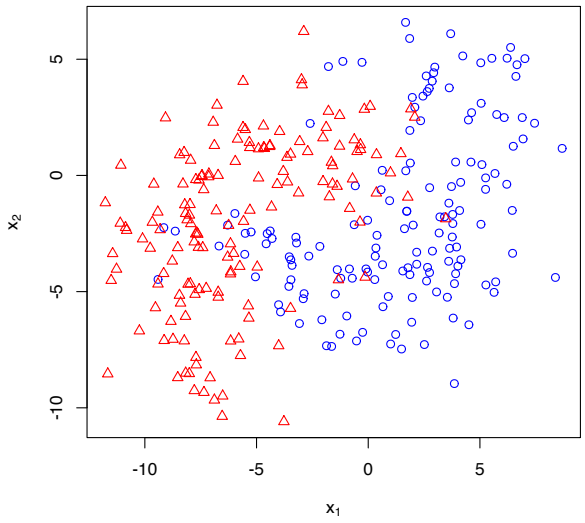


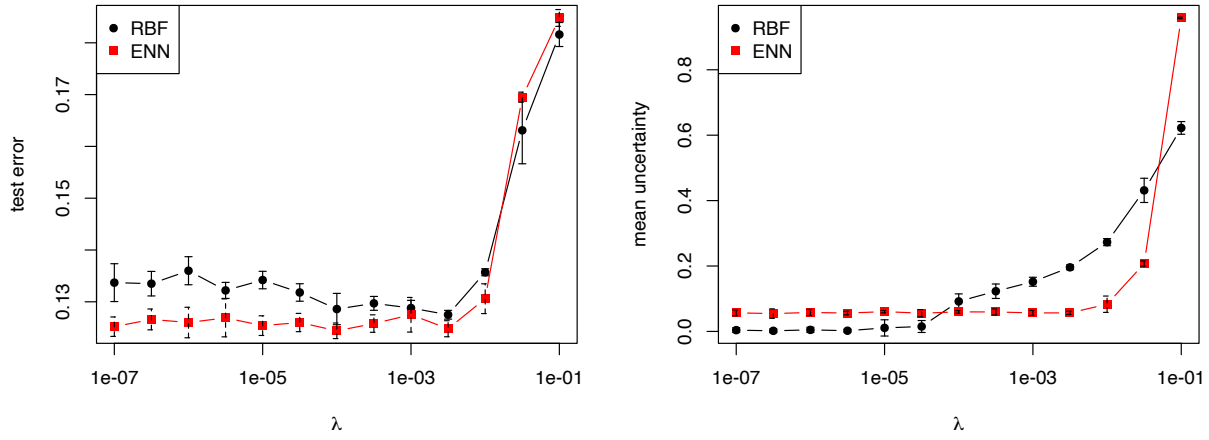
Figure 4: Simulated data.

algorithms with different initializations. As expected, uncertainty increases with λ for both models, but the ENN model appears to be less sensitive to λ as compared to the RBF model. Both models achieve similar minimum error rates for λ around 10^{-3}, and have similar mean uncertainties for $\lambda = 10^{-4}$.

As shown in [31], the robustness of the ENN model arises from the fact that, when the input $\boldsymbol{x}$ is far from all prototypes, the output mass function m is close to the vacuous mass function. This property, in particular, makes the network capable of detecting observations generated from a distribution that is not represented in the learning set. From (15c), we can expect the RBF network model to have a similar property: if $\boldsymbol{x}$ is far from all prototypes, all weights of evidence w_i will be small and the mass $m(\Omega)$ will be close to unity. To compare the mass functions computed by the two models, not only in regions of high density where training data are present, but also in regions of low density, we introduced a third class in the test set, as shown in Figure 6. Figure 7 shows scatter plots of masses on each of the focal sets computed for the two models trained with $\lambda = 10^{-3}$ and applied to an extended dataset composed of the learning data and the third class. We can see that the mass functions are quite similar. Contour plots shown in Figure 6 confirm this similarity.

4. Proposed model

The main idea of this work is to hybridize a deep medical image segmentation model with one of the evidential classifiers introduced in Section 3. Figure 8 shows the global lymphoma segmentation architecture, composed of an encoder-decoder feature extraction module (UNet), and an evidential layer based one of the two models described in Section 3.



(a) (b)

Figure 5: Test error rates (a) and mean uncertainty (b) for the ENN and RBF models, as functions of regularization parameter λ.

The input is the concatenated PET-CT image volume provided as a tensor of size $2 \times 256 \times 256 \times 128$, where 2 corresponds to the number of modality channels, and $256 \times 256 \times 128$ is the size of each input volume. The PET-CT image volumes are first fed into the feature extraction module, which outputs high-level features in the form of a tensor of size $256 \times 256 \times 128 \times H$, where H is the number of features computed at each voxel. This tensor is then fed into the evidential layer, which outputs mass functions representing evidence about the class of each voxel, resulting in a tensor of size $256 \times 256 \times 128 \times (K+1)$, where $K+1$ is the number of masses (one for each class and one for the frame of discernment Ω). The whole network is trained end-to-end by minimizing a regularized Dice loss. The different components of this model are described in greater detail below.

Feature extraction module. The feature extraction module is based on a UNet [10] with residual encoder and decoder layers [51], as shown in Figure 9. Each down-sampling layer (marked in blue) is composed of convolution, normalization, dropout and activation blocks. Each up-sampling layer (marked in green) is composed of transpose convolution, normalization, dropout and activation blocks. The last layer (marked in yellow) is the bottom connection which does not down or up-sample the data. In the experiments reported in Section 5, the channels (number of filters) were set as $(8, 16, 32, 64, 128)$ with kernel size equal to 5 and convolutional strides equal to $(2, 2, 2, 2)$. The spatial dimension, input channel and output channel of the module were set, respectively, as 3, 2, and the number H of extracted features. (Experiments with several values of H are reported in Section 5.2). The dropout rate was set as 0 and no padding operation was applied. Instance normalization [52] was used to perform intensity normalization across the width, height and depth of a single fea-

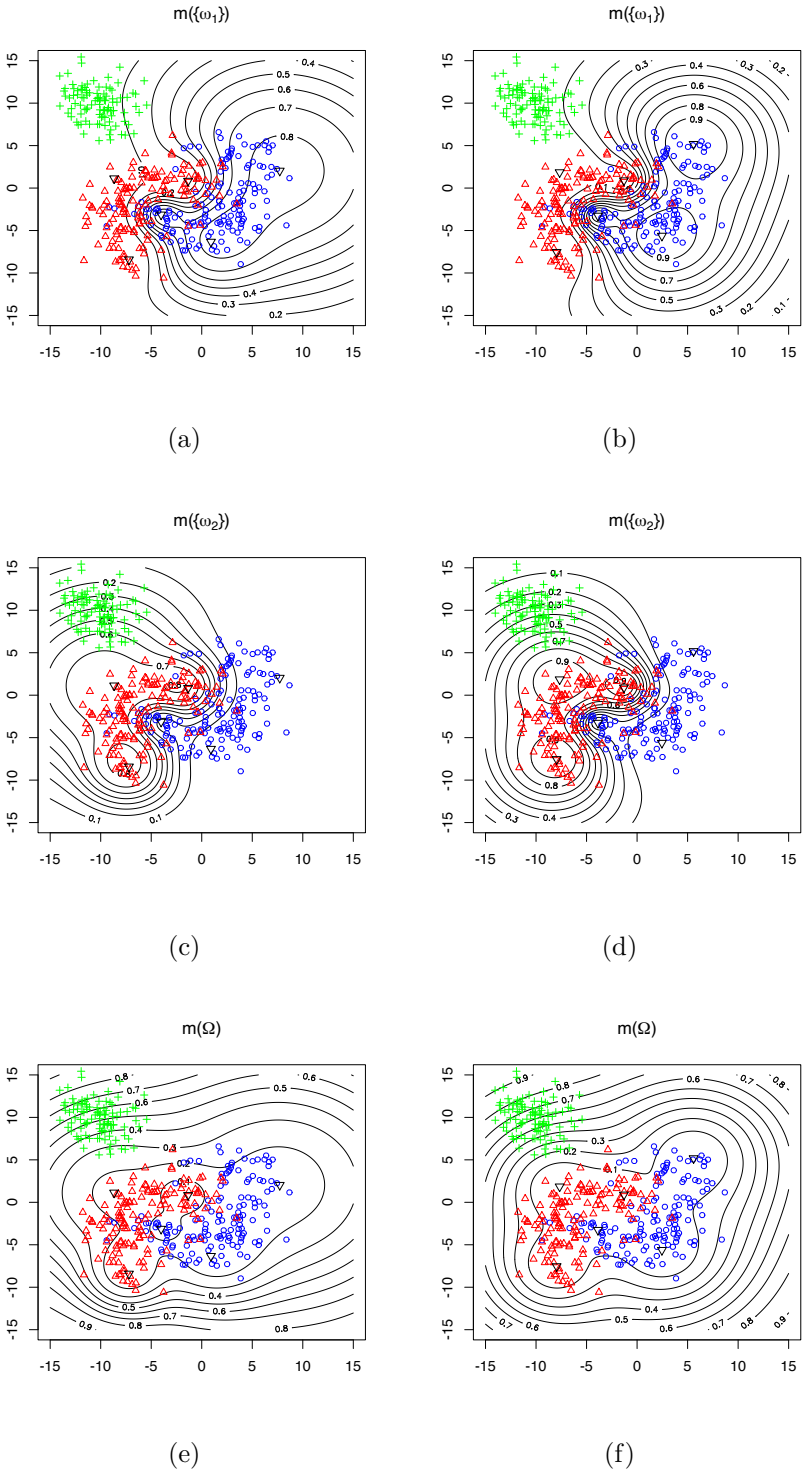
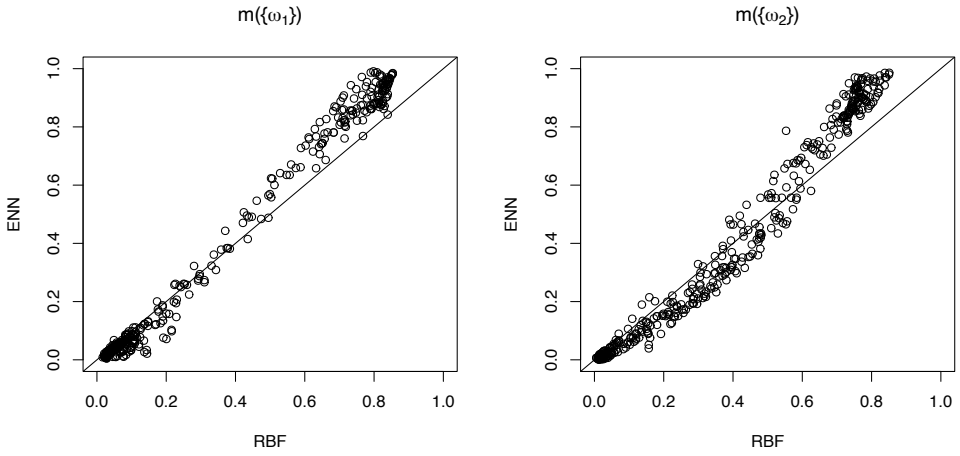
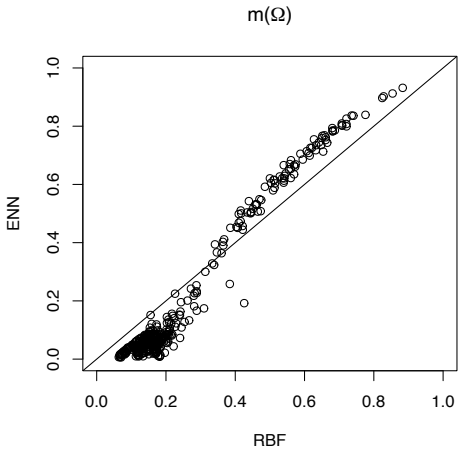


Figure 6: Contours of the mass assigned to $\{\omega_1\}$, $\{\omega_2\}$ and Ω by the RBF (left column) and ENN (right column) models. The training data are displayed in blue and red, and the third class (absent from the training data) is shown in green. Training was done with $\lambda = 0.001$ for the two models.



(a)

(b)



(c)

Figure 7: Masses computed by the RBF network (horizontal axis) versus the ENN model (vertical axis) for the extended dataset.

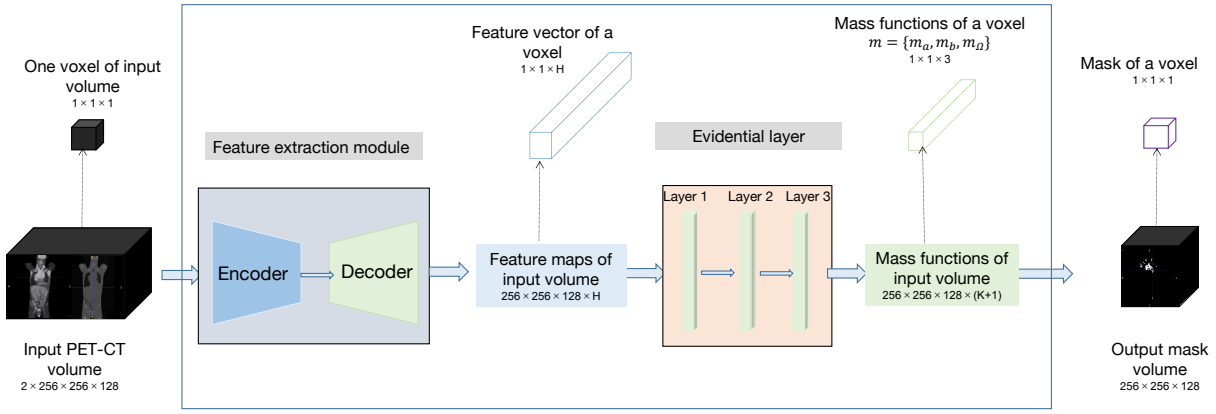


Figure 8: Global lymphoma segmentation model.

ture map of a single example. The Parametric Rectified Linear Unit (PReLU) function [53], which generalizes the traditional rectified unit with a slope for negative values, was used as the activation function. For each input voxel, the feature extraction module outputs a $1 \times H$ feature vector, which is fed into the evidential layer.

Evidential layer. A probabilistic network with a softmax output layer may assign voxels a high probability of belonging to one class while the segmentation uncertainty is actually high because, e.g., the feature vector describing that voxel is far away from feature vectors presented during training. Here, we propose to plug-in one of the evidential classifiers described in Section 3 at the output of the feature extraction module. The ENN or RBF classifier then takes as inputs the high-level feature vectors computed by the UNet and computes, for each voxel n, a mass function m_n on the frame $\Omega = \{\omega_1, \omega_2\}$, where ω_1 and ω_2 denote, respectively, the background and the lymphoma class. We will use the names "ENN-UNet" and "RBF-UNet" to designate the two variants of the architecture.

Loss function. The whole network is trained end-to-end by minimizing a regularized Dice loss. We use the Dice loss instead of the original cross-entropy loss in UNet because the quality of the segmentation is finally assessed by the Dice coefficient. The Dice loss is defined as

$$\mathsf{loss}_D = 1 - \frac{2\sum_{n=1}^{N} S_n G_n}{\sum_{n=1}^{N} S_n + \sum_{n=1}^{N} G_n}, \tag{17}$$

where N is the number of voxels in the image volume, S_n is the output pignistic probability of the tumor class (i.e., $m_n(\{\omega_2\}) + m_n(\Omega)/2$) for voxel n, and G_n is ground truth for voxel n, defined as $G_n = 1$ if voxel n corresponds to a tumor, and $G_n = 0$. The regularized loss function is

$$\mathsf{loss} = \mathsf{loss}_D + \lambda R, \tag{18}$$

where λ is the regularization coefficient and R is a regularizer defined either as $R = \sum_i \alpha_i$ if the ENN classifier is used in the ES module, or as $R = \sum_i v_i^2$ if the RBF classifier is used.

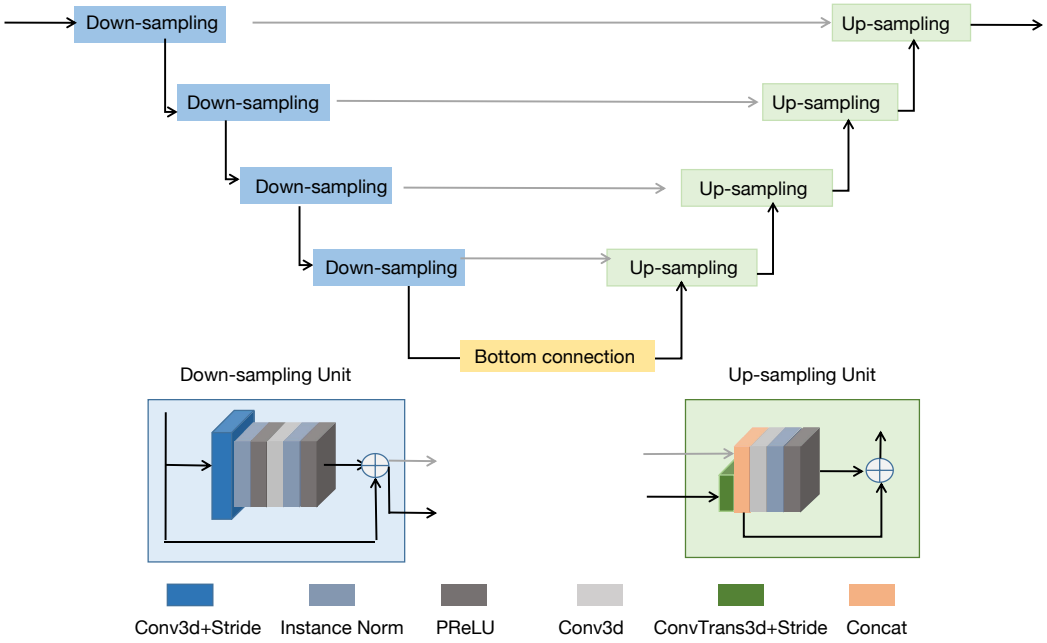


Figure 9: Feature extraction module.

The regularization term allows us to decrease the influence of unimportant prototypes and avoid overfitting.

5. Experiments

The model introduced in Section 4 was applied to a set of PET-CT data recorded on patients with lymphomas[2]. The experimental settings are first described in Section 5.1. A sensitivity analysis with respect to the main hyperparameters is first reported in Section 5.2. We then compare the segmentation accuracy and calibration of our models with those of state-of-the-art models in Sections 5.3 and 5.4, respectively.

5.1. Experimental settings

Dataset. The dataset considered in this paper contains 3D images from 173 patients who were diagnosed with large B-cell lymphomas and underwent PET-CT examination. (The study was approved as a retrospective study by the Henri Becquerel Center Institutional Review Board). The lymphomas in mask images were delineated manually by experts and considered as ground truth. All PET/CT data were stored in the DICOM (Digital Imaging and Communication in Medicine) format. The size and spatial resolution of PET and CT images and the corresponding mask images vary due to the use of different imaging machines and operations. For CT images, the size varies from $267 \times 512 \times 512$ to $478 \times 512 \times 512$. For PET images, the size varies from $276 \times 144 \times 144$ to $407 \times 256 \times 256$.

[2]Our code is available at `https://github.com/iWeisskohl`.

Pre-processing. Several pre-processing methods were used to process the PET/CT data. At first, the data in DICOM format were transferred into the NIFTI (Neuroimaging Informatics Technology Initiative) format for further processing. Second, the PET, CT and mask images were normalized: (1) for PET images, we applied a random intensity shift and scale of each channel with the shift value of 0 and scale value of 0.1; (2) for CT images, the shift and scale values were set to 1000 and 1/2000; (3) for mask images, the intensity value was normalized into the $[0, 1]$ interval by replacing the outside value by 1. Third, PET and CT images were resized to $256 \times 256 \times 128$ by linear interpolation, and mask images were resized to $256 \times 256 \times 128$ by nearest neighbor interpolation. Lastly, the registration of CT and PET images was performed by B-spline interpolation. All the prepossessing methods can be found in the SimpleITK [54][55] toolkit. During training, PET and CT images were concatenated as a two-channel input. We randomly selected 80% of the data for training, 10% for validation and 10% for testing. This partition was fixed and used in all the experiments reported below.

Parameter initialization. For the evidential layer module, we considered two variants based on the ENN classifier recalled in Section 3.1 on the one hand, and on an RBF network as described in Section 3.2 on the other hand. Both approaches are based on prototypes in the space of features extracted by the UNet module. When using ENN or RBF classifiers as stand-alone classifiers, prototypes are usually initialized by a clustering algorithm such as the k-means. Here, this approach is not so easy, because the whole network is trained in an end-to-end way, and the features are constructed during the training process. However, k-means initialization can still be performed by a four-step process:

1. A standard UNet architecture (with a softmax output layer) is trained end-to-end;
2. The k-means algorithm is run in the space of features extracted by the trained UNet;
3. The evidential layer is trained alone, starting from the initial prototypes computed by the k-means;
4. The whole model (feature extraction module and evidential layer) is fine-tuned by end-to-end learning with a small learning step.

As an alternative method, we also considered training the feature extraction module and the evidential layer simultaneously, in which case the prototypes were initialized randomly from a normal distribution with zero mean and identity covariance matrix. For the ENN module, the initial values of parameters α_i and γ_i were set, respectively, at 0.5 and 0.01, and membership degrees u_{ik} were initialized randomly by drawing uniform random numbers and normalizing. For the RBF module, the initial value of the scale parameter γ_i of RBF was set to 0.01, and the weight v_i were drawn randomly from a standard normal distribution.

Learning algorithm. Each model was trained on the learning set with 100 epochs using the Adam optimization algorithm. The initial learning rate was set to 10^{-3}. An adjusted learning rate schedule was applied by reducing the learning rate when the training loss did not decrease in 10 epochs. The model with the best performance on the validation set was saved as the final model for testing. All methods were implemented in Python with the

PyTorch-based medical image framework MONAI, and were trained and tested on a desktop with a 2.20GHz Intel(R) Xeon(R) CPU E5-2698 v4 and a Tesla V100-SXM2 graphics card with 32 GB GPU memory.

Evaluation criteria. The evaluation criteria most commonly used to assess the quality of medical image segmentation algorithms are the Dice score, Sensitivity and Precision. These criteria are defined as follows:

$$\mathsf{Dice}(P,T) = \frac{2 \times TP}{FP + 2 \times TP + FN},$$

$$\mathsf{Sensitivity}(P,T) = \frac{TP}{TP + FN},$$

$$\mathsf{Precision}(P,T) = \frac{TP}{TP + FP},$$

where TP, FP, and FN denote, respectively, the numbers of true positive, false positive, false negative voxels (See Figure 10). The reported results in the following sections were obtained by calculating these three criteria for each test 3D image and then averaging over the patients. The Dice score is a global measure of segmentation performance. It is equal to twice the volume of the intersection between the predicted and actual tumor regions, divided by the sum of the volumes of these regions. Sensitivity is the proportion, among actual tumor voxels, of voxels correctly predicted as tumor. Precision is the proportion, among predicted tumor voxels, of voxels that actually belong to the tumor region; it is, thus, an estimate of the probability that the model is correct when it predicts that a voxel is in a lymphoma region. We note that neither sensitivity, nor precision are global performance criteria. We can increase sensitivity by predicting the tumor class more often (at the expense of misclassifying a lot of background pixels), and we can increase precision by being very cautious and predicting the tumor class only when it has a high probability (at the expense of missing a lot of tumor voxels). These two criteria, thus, have to be considered jointly. Finally, we can also remark that a forth criterion can also be defined: specificity, which is the proportion, among background voxels, of voxels correctly predicted as background (i.e., $TN/(TN + FP)$). However, as there are much more background voxels than tumor ones, this criterion is not informative in tumor segmentation applications (it is always very close to 1).

In addition to quality of the segmentation, we also wish to evaluate the calibration of output probabilities or belief functions (see Section 5.4). For that purpose, we will use an additional evaluation criterion, the Expected Calibration Error (ECE) [56]. The output pignistic probabilities from the evidential layer are first discretized into R equally spaced bins B_r, $r = 1, \ldots, R$ (we used $R = 10$). The accuracy of bin B_r is defined as

$$\mathsf{acc}(B_r) = \frac{1}{\mid B_r \mid} \sum_{i \in B_r} \mathbf{1}(P_i = G_i), \tag{19}$$

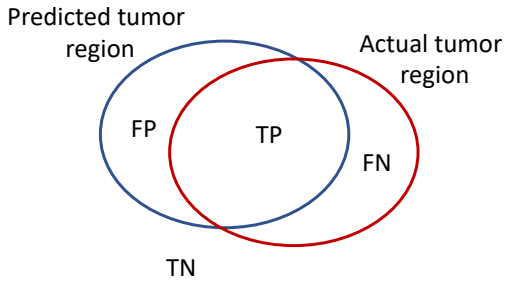


Figure 10: Geometric interpretation of the numbers of true positive (TP), false positive (FP), true negative (TN) and false negative (TN) used for the definition of evaluation criteria.

where P_i and G_i are, respectively, the predicted and true class labels for sample i. The average confidence of bin B_r is defined as

$$\mathsf{conf}(B_r) = \frac{1}{\mid B_r \mid} \sum_{i \in B_r} S_i, \tag{20}$$

where S_i is the confidence for sample i. The ECE is the weighted average of the difference in accuracy and confidence of the bins:

$$\mathsf{ECE} = \sum_{r=1}^{R} \frac{\mid B_r \mid}{N} \mid \mathsf{acc}(B_r) - \mathsf{conf}(B_r) \mid, \tag{21}$$

where N is the total number of elements in all bins, and $\mid B_r \mid$ is the number of elements in bin B_r. A model is perfectly calibrated when $\mathsf{acc}(B_r) = \mathsf{conf}(B_r)$ for all $r \in \{1, \ldots, R\}$. Through the bin-size weighting in the ECE metric, the highly confident and accurate background voxels significantly affect the results. Because our dataset has imbalanced foreground and background proportions, we only considered voxels belonging to the tumor to calculate the ECE, similar to [57][58]. For each patient in the test set, we defined a bounding box covering the lymphoma region and calculated the ECE in this bounding box. We are interested in the patient-level ECE and thus reported the mean patient ECE instead of the voxel-level ECE (i.e., considering all voxels in the test set to calculate the ECE).

5.2. Sensitivity analysis

We analyzed the sensitivity of the results to the main design hyperparameters, which are: the number H of extracted features, the number I of prototypes and the regulation coefficient λ. The influence of the initialization method was also be studied. In all the experiments reported in this section as well as in Section 5.3, learning in each of the configurations was repeated five times with different random initial conditions.

Influence of the number of features. Table 1 shows the means and standard deviations (over five runs) of the three performance indices for ENN-UNet and RBF-UNet with different numbers of features ($H \in \{2, 5, 8\}$). The number of prototypes and the regularization coefficient were set, respectively, to $I = 10$ and $\lambda = 0$. The prototypes were initialized

Table 1: Means and standard deviations (over five runs) of the performance measures for different input dimensions H, with $I = 10$ randomly initialized prototypes and $\lambda = 0$. The best values are shown in bold.

Model	H	Dice score		Sensitivity		Precision	
		Mean	SD	Mean	SD	Mean	SD
ENN-UNet	2	**0.833**	0.009	**0.819**	0.019	0.872	0.018
	5	0.831	0.012	0.817	0.016	0.870	0.011
	8	0.829	0.006	0.816	0.010	**0.877**	0.019
RBF-UNet	2	0.824	0.009	**0.832**	0.008	0.845	0.016
	5	**0.825**	0.006	0.817	0.016	**0.862**	0.010
	8	0.821	0.011	0.813	0.010	0.862	0.022

Table 2: Means and standard deviations (over five runs) of the performance measures for different values of the regularization coefficient λ, with $I = 10$ randomly initialized prototypes and $H = 2$ features. The best values are shown in bold.

Model	λ	Dice score		Sensitivity		Precision	
		Mean	SD	Mean	SD	Mean	SD
ENN-UNet	0	**0.833**	0.009	**0.819**	0.019	**0.872**	0.018
	1e-4	0.822	0.007	0.818	0.026	0.839	0.035
	1e-2	0.823	0.004	0.817	0.023	0.856	0.023
RBF-UNet	0	0.824	0.009	**0.832**	0.008	0.845	0.016
	1e-4	0.825	0.011	0.811	0.022	**0.869**	0.020
	1e-2	**0.829**	0.010	0.818	0.022	0.867	0.016

randomly. ENN-UNet achieves the highest Dice score and sensitivity with $H = 2$ features, but the highest precision with $H = 8$. However, the differences are small and concern only the third decimal point. Similarly, RBF-UNet had the best values of the Dice score and precision for $H = 5$ features, but again the differences are small. Overall, it seems that only two features are sufficient to discriminate between tumor and background voxels.

Influence of the regularization coefficient. In the previous experiment, the networks were trained without regularization. Tables 2 and 3 show the performances of ENN-UNet and RBF-UNet for different values of λ, with $I = 10$ randomly initialized prototypes and, respectively, $H = 2$ and $H = 8$ inputs. With both settings, ENN-UNet does not benefit from regularization (the best results are obtained with $\lambda = 0$). In contrast, RBF-UNet is more sensitive to regularization, and achieves the highest Dice score with $\lambda = 0.01$. This finding confirms the remark already made in Section 3.3, where it was observed that an ENN classifier seems to be less sensitive to regularization than an RBF classifier (see Figure 5a).

Influence of the number of prototypes. The number I of prototypes is another hyperparameter that may impact segmentation performance. Table 4 shows the performances of ENN-UNet and RBF-UNet with 10 and 20 randomly initialized prototypes, the other hyperparameters being fixed at $H = 2$ and $\lambda = 0$. Increasing the number of prototypes beyond 10 does not seem to improve the performance of ENN-UNet, while it does slightly improve

Table 3: Means and standard deviations (over five runs) of the performance measures for different values of the regularization coefficient λ, with $I = 10$ randomly initialized prototypes and $H = 8$ features. The best values are shown in bold.

Model	λ	Dice score		Sensitivity		Precision	
		Mean	SD	Mean	SD	Mean	SD
ENN-UNet	0	**0.829**	0.006	**0.811**	0.010	**0.877**	0.019
	1e-4	0.827	0.008	0.809	0.019	0.873	0.024
	1e-2	0.822	0.009	0.807	0.021	0.867	0.011
RBF-UNet	0	0.821	0.010	0.813	0.010	0.862	0.022
	1e-4	0.827	0.004	**0.830**	0.005	0.852	0.012
	1e-2	**0.832**	0.006	0.825	0.022	**0.867**	0.020

Table 4: Means and standard deviations (over five runs) of the performance measures for different numbers I of randomly initialized prototypes, with $H = 2$ features and $\lambda = 0$. The best values are shown in bold.

Model	I	Dice score		Sensitivity		Precision	
		Mean	SD	Mean	SD	Mean	SD
ENN-UNet	10	**0.833**	0.009	**0.819**	0.019	**0.872**	0.018
	20	0.823	0.007	0.804	0.006	0.864	0.012
RBF-UNet	10	0.824	0.009	**0.832**	0.008	0.845	0.016
	20	**0.830**	0.007	0.810	0.012	**0.867**	0.010

the performance of RBF-UNet in terms of Dice score and precision, at the expense of an increased computing time.

Influence of the prototype initialization method. Finally, we compared the two initialization methods mentioned in Section 5.1. For k-means initialization, in the first step, a UNet model was trained with the following settings: kernel size=5, channels $=(8, 16, 32, 64, 128)$ and strides$=(2, 2, 2, 2)$. The spatial dimension, input and output channel were set, respectively, 3, 2, and 2. This pre-trained UNet was used to extract $H = 2$ features, and 10 prototypes were obtained by running the k-means algorithm in the space of extracted features. These prototypes were fed into ENN or RBF layers, which were trained separately, with fixed features. For this step, the learning rate was set to 10^{-2}. Finally, the whole model was fine-tuned end-to-end, with a smaller learning rate equal to 10^{-4}. Table 5 shows the performances of ENN-UNet and RBF-UNet with random and k-means initialization. Both ENN-UNet and RBF-UNet achieve a higher Dice score when using the k-means initialization method, and the variability of the results is also reduced with this method.

Not only does the k-means initialization method slightly improve the performances of ENN-UNet and RBF-UNet quantitatively, but it also tends to position the prototypes in regions of high data density. As a result, a high output mass $m(\Omega)$ signals that the input data is atypical. In that sense, the output mass function is more interpretable. This point is illustrated by Figures 11 and 12, which show the contours, in the two-dimensional feature space, of the masses assigned to the background, the tumor class and the frame of discern-

Table 5: Means and standard deviations (over five runs) of the performance measures for different initialization methods, with $I = 10$ prototypes, $H = 2$ features and $\lambda = 0$. The best values are shown in bold.

Model	Initialization	Dice score		Sensitivity		Precision	
		Mean	SD	Mean	SD	Mean	SD
ENN-UNet	Random	0.833	0.009	0.819	0.019	0.872	0.018
	k-means	**0.846**	0.002	**0.830**	0.004	**0.879**	0.008
RBF-UNet	Random	0.824	0.009	**0.832**	0.008	0.845	0.016
	k-means	**0.839**	0.003	0.824	0.001	**0.879**	0.008

ment when using k-means initialization (with $\lambda = 10^{-2}$ and $I = 10$) with, respectively, ENN-UNet and RBF-Unet. For both models, the prototypes are well distributed over the two classes, and the mass on Ω decreases with the distance to the data, as expected. In contrast, when using random initialization (as shown in Figure 13 for the ENN-UNet model – results are similar with the RBF-UNet model), the prototypes are located in the background region, and the mass $m(\Omega)$ does not have a clear meaning (although the decision boundary still ensures a good discrimination between the two classes).

From this sensitivity analysis, we can conclude that the performances of both ENN-UNet and RBF-UNet are quite robust to the values of the hyperparameters, and that the two models achieve comparable performances. The k-means initialization method seems to yield better results, both quantitatively and qualitatively. The next section is devoted to a comparison with alternative models.

5.3. Comparative analysis: segmentation accuracy

In this section, we compare the performances of the ENN-UNet and RBF-UNet models with those of the baseline model, UNet [10], as well as three state-of-the-art models reviewed in Section 1: VNet [11], SegResNet [12] and nnUNet [13]. For all compared methods, the same learning set and pre-processing steps were used. All the compared methods were trained with the Dice loss function (17). Details about the optimization algorithm were given in Section 5.1. All methods were implemented based on the MONAI framework[3] and can be called directly. For UNet, the kernel size was set as 5 and the channels were set to $(8, 16, 32, 64, 128)$ with strides=$(2, 2, 2, 2)$. For nnUNet, the kernel size was set as $(3, (1, 1, 3), 3, 3)$ and the upsample kernel size was set as $(2, 2, 1)$ with strides $((1, 1, 1), 2, 2, 1)$. For SegResNet [12] and VNet [11], we used the pre-defined model without changing any parameter. The spatial dimension, input channel and output channel were set, respectively, 3, 2, and 2 for the four compared models. As for other hyperparameters not mentioned here, we used the pre-defined value given in MONAI. As shown by the sensitivity analysis performed in Section 5.2, the best results for ENN-UNet and RBF-UNet are achieved with $\lambda = 0$, $I = 10$, $H = 2$ and k-means initialization.

[3]More details about how to use those models can be found from MONAI core tutorials `https://monai.io/started.html#monaicore`.

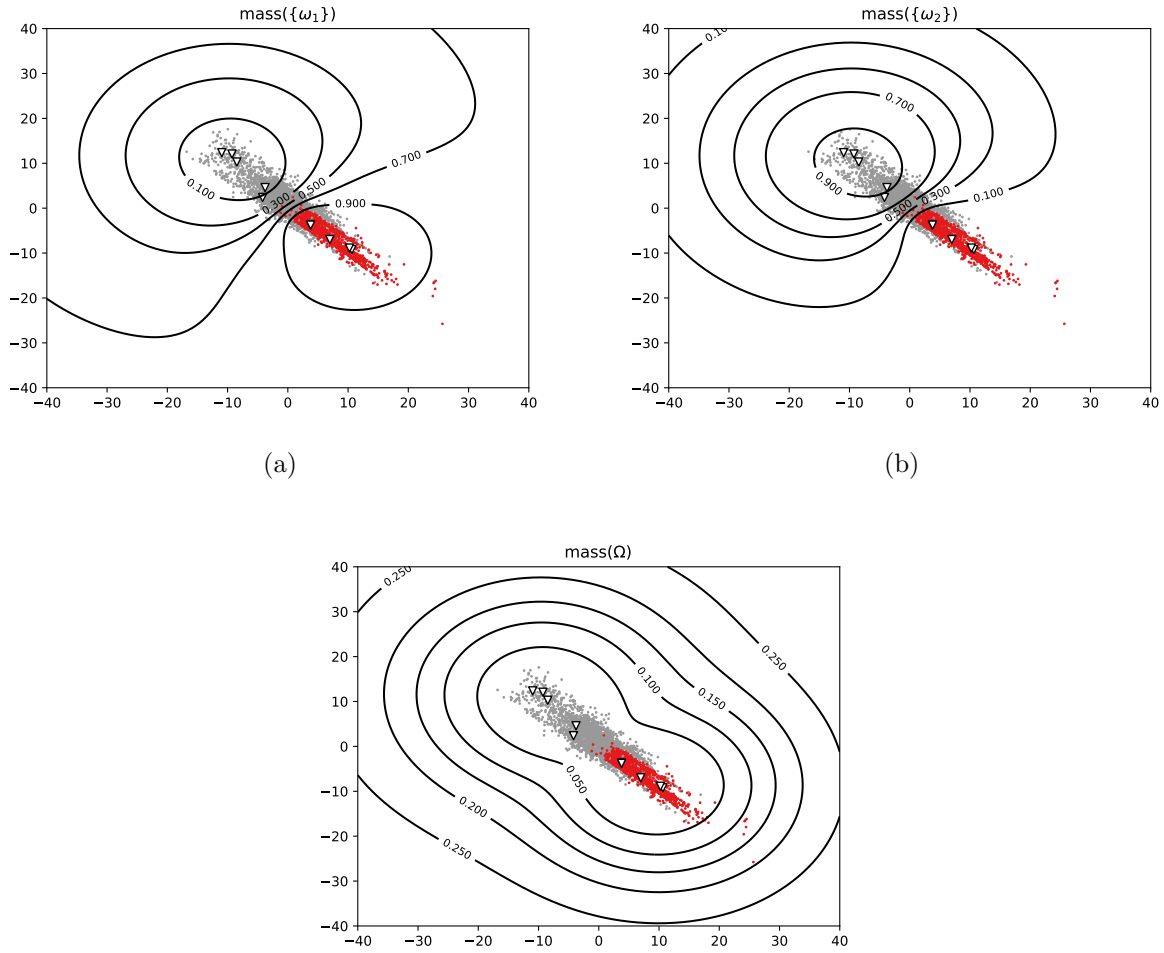


Figure 11: Contours in feature space of the masses assigned to the background (a), the tumor class (b) and the frame of discernment (c) by the ENN-UNet model initialized by k-means. Training was done with $\lambda = 10^{-2}$, $H = 2$ and $I = 10$. Sampled feature vectors from the tumor and background classes are marked in gray and red, respectively.

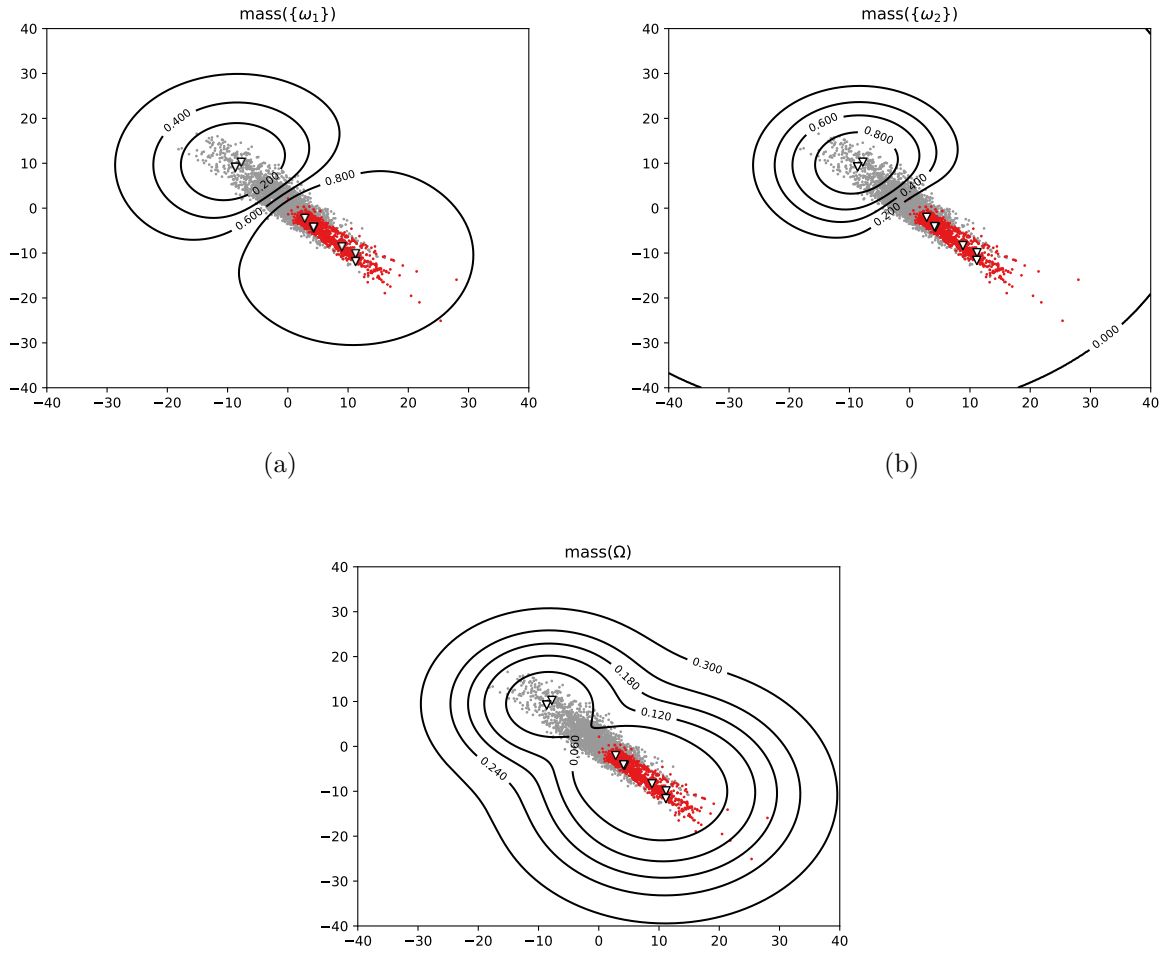
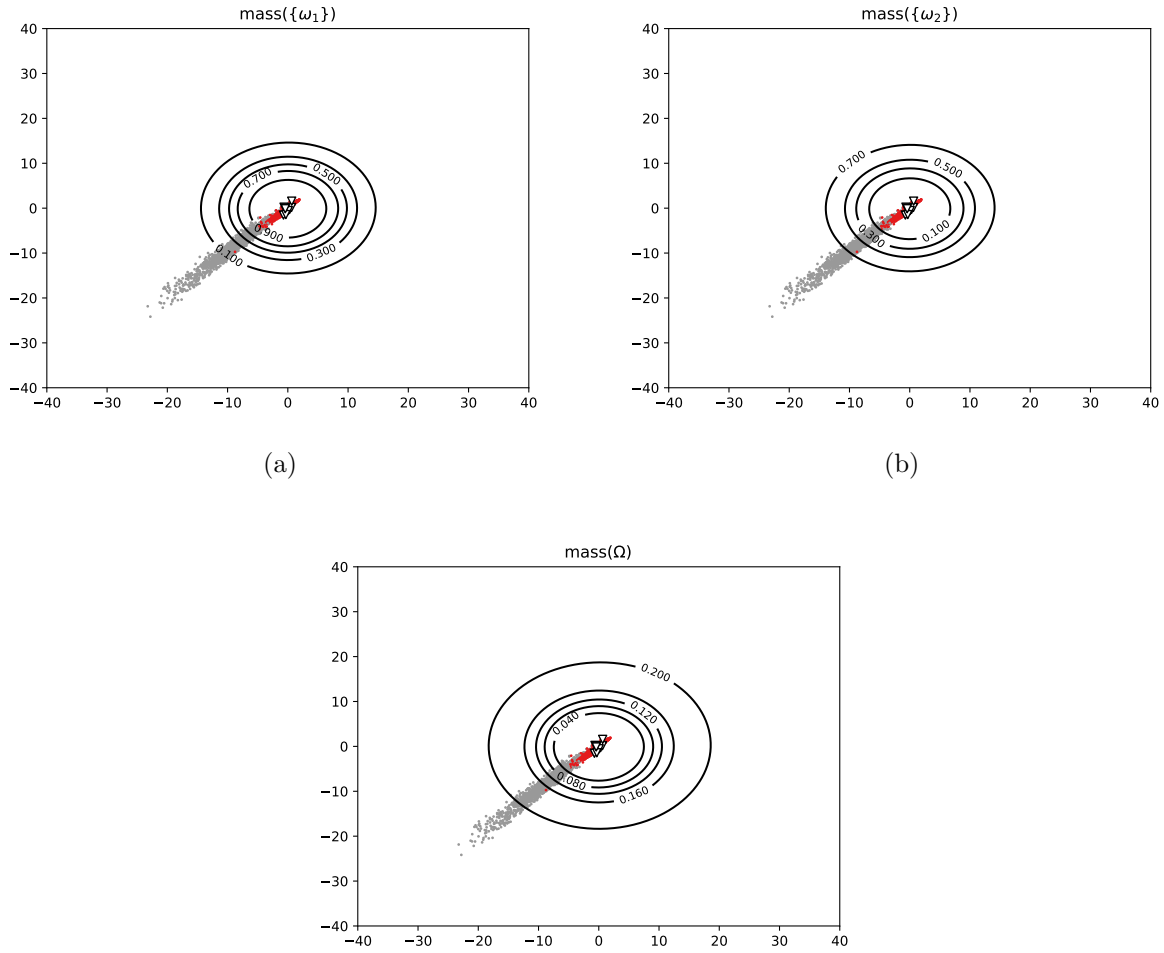


Figure 12: Contours in feature space of the masses assigned to the background (a), the tumor class (b) and the frame of discernment (c) by the RBF-UNet model initialized by k-means. Training was done with $\lambda = 10^{-2}$, $H = 2$ and $I = 10$. Sampled feature vectors from the tumor and background classes are marked in gray and red, respectively.



(a) (b) (c)

Figure 13: Contours in feature space of the masses assigned to the background (a), the tumor class (b) and the frame of discernment (c) by the ENN-UNet model initialized randomly. Training was done with $\lambda = 10^{-2}$, $H = 2$ and $I = 10$. Sampled feature vectors from the tumor and background classes are marked in gray and red, respectively.

Table 6: Means and standard deviations (over five runs) of the performance measures for ENN-UNet, RBF-UNet and four reference methods. The best result is shown in bold, and the second best is underlined.

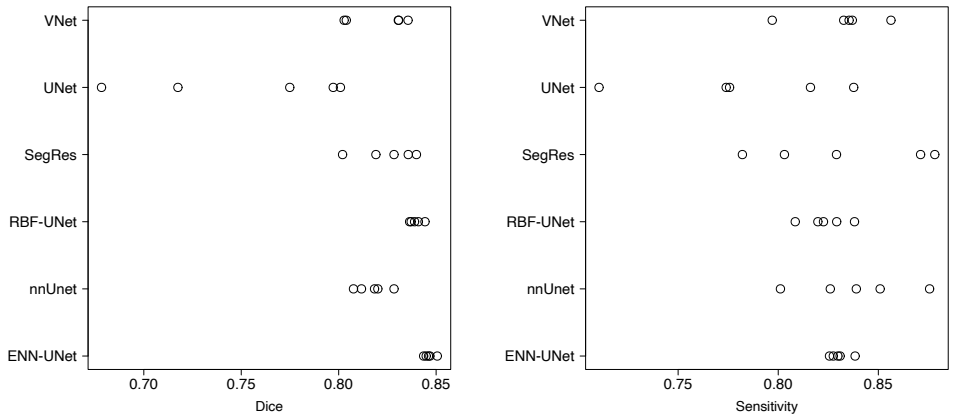
Model	Dice score		Sensitivity		Precision	
	Mean	SD	Mean	SD	Mean	SD
UNet [51]	0.753	0.054	0.782	0.048	<u>0.896</u>	0.047
nnUNet [13]	0.817	0.008	**0.838**	0.028	0.879	0.032
VNet [11]	0.820	0.016	0.831	0.021	**0.901**	0.056
SegResNet [12]	0.825	0.015	<u>0.832</u>	0.042	0.876	0.051
ENN-UNet	**0.846**	0.002	0.830	0.004	0.879	0.008
RBF-UNet	<u>0.839</u>	0.003	0.824	0.001	0.879	0.008

Table 7: Conover-Iman test of multiple comparisons between the Dice scores obtained by the six models: t-test statistics and p-values. P-values less than 0.01 are printed in bold.

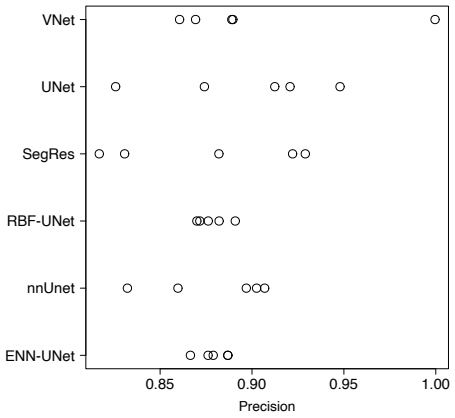
	ENN-UNet	nnUnet	RBF-UNet	SegResNet	UNet
nnUnet	6.759				
	0.0000				
RBF-UNet	2.156	-4.602			
	0.0857	**0.0004**			
SegResNet	5.349	-1.410	3.193		
	0.0001	0.3282	**0.0088**		
UNet	10.283	3.524	8.127	4.934	
	0.0000	**0.0043**	**0.0000**	**0.0002**	
VNet	6.054	-0.705	3.898	0.705	-4.229
	0.0000	0.8091	**0.0019**	0.8669	**0.0009**

The means and standard deviations of the Dice score, sensitivity and precision over five runs with random initialization for the six methods are shown in Table 6, and the raw values are plotted in Figure 14. We can see that ENN-UNet and RBF-UNet achieve, respectively, the highest and the second highest mean Dice score. A Kruskal-Wallis test performed on the whole data concludes to a significant difference between the distributions of the Dice score for the six methods (p-value = 0.0001743), while the differences are not significant for sensitivity (p-value = 0.2644) and precision (p-value = 0.9496). Table 7 shows the results of the Conover-Iman test of multiple comparisons [59][60] with Benjamini-Yekutieli adjustment [61]. We can see that the differences between the Dice scores obtained by ENN-UNet and RBF-UNet on the one hand, and the four other methods on the other hand are highly significant (p-values $< 10^{-2}$), while the difference between ENN-UNet and RBF-UNet is only weakly significant (p-value = 0.0857).

Figure 15 shows two examples of segmentation results obtained by ENN-UNet and UNet, corresponding to large and isolated lymphomas. We can see, in these two examples, that UNet is more conservative (it correctly detects only a subset of the tumor voxels), which may explain why it has a relatively high precision. However, the tumor regions predicted



(a) (b)



(c)

Figure 14: Values of the Dice score (a), sensitivity (b) and precision (c) for five runs of the six methods.

by ENN-UNet better overlap the ground-truth tumor region, which is also reflected by the higher Dice score.

5.4. Comparative analysis: calibration

Besides segmentation accuracy, another important issue concerns the quality of uncertainty quantification. Monte-Carlo dropout (MCD) [25] is a a state-of-the-art technique for improving uncertainty quantification capabilities of deep networks. In this section, we compare the ECE (21) achieved by UNet (the baseline), SegResNet (the best alternative method found in Section 5.3), and our proposals: ENN-UNet, and RBF-UNet, with and without MCD. For the four methods, the dropout rate was set to 0.5 and the sample number was set to 20; we averaged the 20 output probabilities (the pignistic probabilities for the two evidential models) at each voxel as the final output of the model.

The results are reported in Table 8. We can see that MCD enhances the segmentation performance (measured by the Dice index) of UNet et SegResNet, and improves the calibration of all methods, except SegResNet. Overall, the smallest average ECE is achieved by RBF-UNet and ENN-UNet with MCD, but the standard deviations are quite large. A Kruskal-Wallis test concludes to a significant difference between the distributions of ECE for the eight methods (p-value = 0.01). The p-values of the Conover-Iman test of multiple comparisons with Benjamini-Yekutieli adjustment reported in Table 9 show significant differences between the ECE of RBF-UNet with MCD one the one hand, and those of RBF-UNet without MCD, SegResNet with MCD, and UNet without MCD on the other hand. We also tested the pairwise differences between the ECE values obtained by RBF-UNet and ENN-UNet with MCD on the one hand, and UNet with and without MCD as well as SegResNet with and without MCD on the other hand using the Wilcoxon rank sum test. The corresponding p-values are shown in Table 10. We find significant differences between the ECE RBF-UNet with MCD and those of the other methods, but only a weakly significant difference between ENN-UNet with MCD and UNet without MCD. In summary, there is some evidence that MCD improves calibration, even for evidential models, and that the best calibration is achieved by the RBF-UNet model, but this evidence is not fully conclusive due to the limited size of the dataset; our findings will have to be confirmed by further experiments with larger datasets.

6. Conclusion

An evidential framework for segmenting lymphomas from 3D PET-CT images with uncertainty quantification has been proposed in this paper. Our architecture is based on the concatenation of a UNet, which extracts high-level features from the input images, and an evidential segmentation module, which computes output mass functions for each voxel. Two versions of this evidential module, both involving prototypes, have been studied: one is based on the ENN model initially proposed as a stand-alone classifier in [31], while the other one relies on an RBF layer and the addition of weight of evidence. The whole model is trained end-to-end by minimizing the Dice loss. The initialization of prototypes has been shown to be a crucial step in this approach. The best method found has been to pre-train

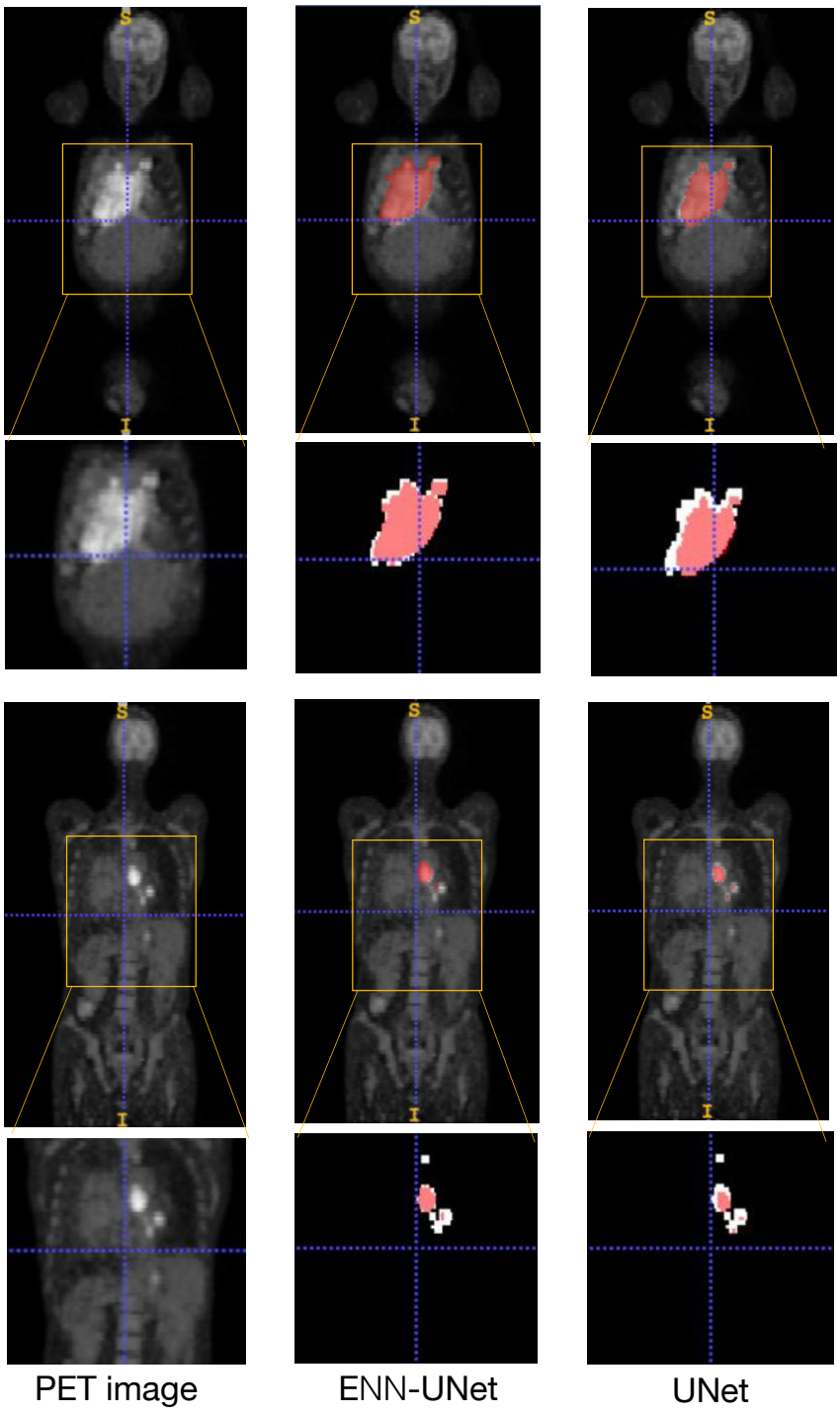


Figure 15: Two examples of segmentation results by ENN-UNet and UNet. The first and the second row are, respectively, representative of large and isolated small lymphomas. The three columns correspond, from left to right, to the PET images and the segmentation results obtained by ENN-UNet and UNet. The white and red region represent, respectively, the ground truth and the segmentation result.

Table 8: Means and standard deviations (over five runs) of the Dice score and ECE for UNet, SegResNet, ENN-UNet andRBF-UNet, with and without MCD. The best results are shown in bold, the second best are underlined.

Model	Dice score		ECE(%)	
	Mean	SD	Mean	SD
UNet	0.754	0.054	2.22	0.205
SegResNet	0.825	0.015	1.97	0.488
ENN-UNet	**0.846**	0.002	1.99	0.110
RBF-UNet	0.839	0.003	2.12	0.028
UNet with MC	0.828	0.005	1.93	0.337
SegResNet with MC	<u>0.844</u>	0.009	2.53	0.973
ENN-UNet with MC	0.841	0.003	<u>1.53</u>	0.075
RBF-UNet with MC	0.840	0.003	**1.52**	0.041

Table 9: Conover-Iman test of multiple comparisons between the ECE obtained by UNet, SegResNet, ENN and RBF, with and without MCD: t-test statistics and p-values. P-values less than 0.01 are printed in bold.

	ENN	ENN-MC	RBF	RBF-MC	SegRes	SegRes-MC	UNet
ENN-MC	0.926						
	1.0000						
RBF	-1.191	-2.118					
	0.7403	0.2892					
RBF-MC	2.812	1.886	4.004				
	0.1145	0.3419	**0.0095**				
SegRes	0.695	-0.232	1.886	-2.117			
	1.0000	1.0000	0.3761	0.3305			
SegRes-MC	-0.860	-1.787	0.331	-3.673	-1.555		
	1.0000	0.3530	1.0000	**0.0159**	0.4756		
UNet	-1.357	-2.283	-0.165	-4.169	-2.051	-0.496	
	0.6337	0.2677	1.0000	**0.0119**	0.2962	1.0000	
UNet-MC	0.430	-0.496	1.621	-2.382	-0.265	1.290	1.787
	1.0000	1.0000	0.4507	0.2564	1.0000	0.6667	0.3824

Table 10: P-values for the Wilcoxon rank sum test applied to the comparison of ECE obtained by ENN-UNet and RBF UNet with MCD on the one hand, and the four other methods on the other hand (UNet and SegResNet with and without MCD).

	UNet	UNet-MC	SegRes	SegRes-MC
ENN-MC	0.095	0.67	0.69	0.31
RBF-MC	0.0079	0.012	0.055	0.0079

a UNet with a softmax output layer, initialize the prototype with the k-means algorithm in the space of extracted features, train the evidential layer separately, and fine-tune the whole network. Our model has been shown to outperform the baseline UNet model as well as other state-of-the-art segmentation method on a dataset of 173 patients with lymphomas. Preliminary results also suggest the outputs of the evidential models (in particular, the one with an RBF layer) are better calibrated and that calibration error can be further decreased by Monte Carlo dropout. These results, however, will have to be confirmed by further experiments with larger datasets.

This work can be extended in many directions. One of them is to further evaluate the approach by applying it to other medical image segmentation problems. One of the potential problems that may arise is related to the dimensionality of the feature space. In the application considered in this paper, good results where obtained with only two extracted features. If some other learning tasks require a much larger number of features, we may need a much higher number of prototypes and learning may be slow. This issue could be addressed by adapting the loss function as proposed, e.g., in [62]. We also plan to further study the calibration properties of the belief functions computed by our approach (using calibration measures specially designed for belief functions), as well as the novelty detection capability of our model.

Acknowledgements

This work was supported by the China Scholarship Council (No. 201808331005). It was carried out in the framework of the Labex MS2T, which was funded by the French Government, through the program "Investments for the future" managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02)

References

[1] Y. S. Jhanwar, D. J. Straus, The role of PET in lymphoma, Journal of Nuclear Medicine 47 (8) (2006) 1326–1334.

[2] H. Zaidi, I. El Naqa, PET-guided delineation of radiation therapy treatment volumes: a survey of image segmentation techniques, European Journal of Nuclear Medicine and Molecular Imaging 37 (11) (2010) 2165–2187.

[3] H. Ilyas, N. G. Mikhaeel, J. T. Dunn, F. Rahman, H. Møller, D. Smith, S. F. Barrington, Defining the optimal method for measuring baseline metabolic tumour volume in diffuse large B cell lymphoma, European journal of nuclear medicine and molecular imaging 45 (7) (2018) 1142–1154.

[4] F. Eude, M. N. Toledano, P. Vera, H. Tilly, S.-D. Mihailescu, S. Becker, Reproducibility of baseline tumour metabolic volume measurements in diffuse large B-cell lymphoma: Is there a superior method?, Metabolites 11 (2) (2021) 72.

[5] D. Onoma, S. Ruan, S. Thureau, et al., Segmentation of heterogeneous or small FDG PET positive tissue based on a 3d-locally adaptive random walk algorithm, Computerized Medical Imaging and Graphics 38 (8) (2014) 753–763.

[6] H. Hu, P. Decazes, P. Vera, H. Li, S. Ruan, Detection and segmentation of lymphomas in 3D PET images via clustering with entropy-based optimization strategy, International journal of computer assisted radiology and surgery 14 (10) (2019) 1715–1724.

[7] H. Li, H. Jiang, S. Li, et al., DenseX-net: an end-to-end model for lymphoma segmentation in whole-body PET/CT images, IEEE Access 8 (2019) 8004–8018.

[8] H. Hu, L. Shen, T. Zhou, et al., Lymphoma segmentation in PET images based on multi-view and conv3d fusion strategy, in: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), IEEE, 2020, pp. 1197–1200.
[9] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, Boston, USA, 2015, pp. 3431–3440.
[10] O. Ronneberger, P. Fischer, T. Brox, U-Net: convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W. M. Wells, A. F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, Cham, 2015, pp. 234–241.
[11] F. Milletari, N. Navab, S.-A. Ahmadi, V-net: Fully convolutional neural networks for volumetric medical image segmentation, in: 2016 fourth international conference on 3D vision, IEEE, 2016, pp. 565–571.
[12] A. Myronenko, 3D MRI brain tumor segmentation using autoencoder regularization, in: International MICCAI Brain lesion Workshop, Springer, 2018, pp. 311–320.
[13] F. Isensee, J. Petersen, A. Klein, D. Zimmerer, et al., Nnu-net: Self-adapting framework for u-net-based medical image segmentation, arXiv preprint arXiv:1809.10486.
[14] P. Blanc-Durand, S. Jégou, S. Kanoun, et al., Fully automatic segmentation of diffuse large B cell lymphoma lesions on 3D FDG-PET/CT for total metabolic tumour volume prediction using a convolutional neural network., European Journal of Nuclear Medicine and Molecular Imaging (2020) 1–9.
[15] L. Huang, T. Denœux, D. Tonnelet, P. Decazes, S. Ruan, Deep pet/ct fusion with dempster-shafer theory for lymphoma segmentation, in: C. Lian, X. Cao, I. Rekik, X. Xu, P. Yan (Eds.), Machine Learning in Medical Imaging, Springer International Publishing, Cham, 2021, pp. 30–39.
[16] G. Shafer, A mathematical theory of evidence, Vol. 42, Princeton University Press, 1976.
[17] E. Hüllermeier, W. Waegeman, Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods, Machine Learning 110 (3) (2021) 457–506.
[18] J. Quiñonero-Candela, M. Sugiyama, N. D. Lawrence, A. Schwaighofer, Dataset shift in machine learning, MIT Press, 2009.
[19] R. Mehta, T. Christinck, T. Nair, P. Lemaitre, D. Arnold, T. Arbel, Propagating uncertainty across cascaded medical imaging tasks for improved deep learning inference, in: Uncertainty for Safe Utilization of Machine Learning in Medical Imaging and Clinical Image-Based Procedures, Springer, 2019, pp. 23–32.
[20] W. J. Maddox, P. Izmailov, T. Garipov, D. P. Vetrov, A. G. Wilson, A simple baseline for Bayesian uncertainty in deep learning, Advances in Neural Information Processing Systems 32 (2019) 13153–13164.
[21] L. Yu, S. Wang, X. Li, C.-W. Fu, P.-A. Heng, Uncertainty-aware self-ensembling model for semi-supervised 3d left atrium segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2019, pp. 605–613.
[22] F. C. Ghesu, B. Georgescu, A. Mansoor, Y. Yoo, E. Gibson, R. Vishwanath, A. Balachandran, J. M. Balter, Y. Cao, R. Singh, et al., Quantifying and leveraging predictive uncertainty for medical image assessment, Medical Image Analysis 68 (2021) 101855.
[23] G. E. Hinton, D. van Camp, Keeping the neural networks simple by minimizing the description length of the weights, in: Proceedings of the Sixth Annual Conference on Computational Learning Theory, COLT '93, Association for Computing Machinery, New York, NY, USA, 1993, p. 5?13.
[24] D. J. MacKay, A practical Bayesian framework for backpropagation networks, Neural computation 4 (3) (1992) 448–472.
[25] Y. Gal, Z. Ghahramani, Dropout as a Bayesian approximation: Representing model uncertainty in deep learning, in: International Conference on Machine Learning, PMLR, 2016, pp. 1050–1059.
[26] D. Tran, M. W. Dusenberry, M. van der Wilk, D. Hafner, Bayesian layers: A module for neural network uncertainty, arXiv preprint arXiv:1812.03973.
[27] A. P. Dempster, Upper and lower probability inferences based on a sample from a finite univariate population, Biometrika 54 (3-4) (1967) 515–528.
[28] T. Denœux, D. Dubois, H. Prade, Representations of uncertainty in artificial intelligence: Beyond

probability and possibility, in: P. Marquis, O. Papini, H. Prade (Eds.), A Guided Tour of Artificial Intelligence Research, Vol. 1, Springer Verlag, 2020, Ch. 4, pp. 119–150.
[29] P. Smets, The combination of evidence in the transferable belief model, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (5) (1990) 447–458.
[30] T. Denoeux, A k-nearest neighbor classification rule based on Dempster-Shafer theory, IEEE Transactions on Systems, Man, and Cybernetics 25 (5) (1995) 804–813. doi:10.1109/21.376493.
[31] T. Denœux, A neural network classifier based on Dempster-Shafer theory, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 30 (2) (2000) 131–150.
[32] T. Denœux, M.-H. Masson, Evclus: evidential clustering of proximity data, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 34 (1) (2004) 95–109.
[33] F. Pichon, D. Mercier, E. Lefèvre, F. Delmotte, Proposition and learning of some belief function contextual correction mechanisms, International Journal of Approximate Reasoning 72 (2016) 4–42.
[34] F. Pichon, D. Dubois, T. Denœux, Quality of information sources in information fusion, in: É. Bossé, G. L. Rogova (Eds.), Information Quality in Information Fusion and Decision Making, Springer International Publishing, Cham, 2019, pp. 31–49.
[35] H. Chen, S. Le Hégarat-Mascle, E. Aldea, Belief functions clustering for epipole localization, International Journal of Approximate Reasoning 137 (2021) 146–165.
[36] T. Denœux, O. Kanjanatarakul, S. Sriboonchitta, A new evidential k-nearest neighbor rule based on contextual discounting with partially supervised learning, International Journal of Approximate Reasoning 113 (2019) 287–302.
[37] C. Gong, Z. gang Su, P. hong Wang, Q. Wang, Y. You, Evidential instance selection for k-nearest neighbor classification of big data, International Journal of Approximate Reasoning 138 (2021) 123–144.
[38] A. Imoussaten, L. Jacquin, Cautious classification based on belief functions theory and imprecise relabelling, International Journal of Approximate Reasoning 142 (2022) 130–146.
[39] T. Denoeux, NN-EVCLUS: Neural network-based evidential clustering, Information Sciences 572 (2021) 297–330.
[40] V. Antoine, J. A. Guerrero, J. Xie, Fast semi-supervised evidential clustering, International Journal of Approximate Reasoning 133 (2021) 116–132.
[41] C. Lian, S. Ruan, T. Denœux, H. Li, P. Vera, Joint tumor segmentation in PET-CT images using co-clustering and fusion based on belief functions, IEEE Transactions on Image Processing 28 (2) (2018) 755–766.
[42] L. Huang, S. Ruan, T. Denœux, Belief function-based semi-supervised learning for brain tumor segmentation, arXiv preprint arXiv:2102.00097.
[43] Z. Tong, P. Xu, T. Denœux, Evidential fully convolutional network for semantic segmentation, Applied Intelligence 51 (2021) 6376–6399.
[44] L. Huang, S. Ruan, P. Decazes, T. Denœux, Evidential segmentation of 3D PET/CT images, in: T. Denœux, E. Lefèvre, Z. Liu, F. Pichon (Eds.), Belief Functions: Theory and Applications, Springer International Publishing, Cham, 2021, pp. 159–167.
[45] P. Smets, R. Kennes, The Transferable Belief Model, Artificial Intelligence 66 (1994) 191–243.
[46] T. Denœux, Analysis of evidence-theoretic decision rules for pattern classification, Pattern Recognition 30 (7) (1997) 1095–1107.
[47] L. Ma, T. Denœux, Partial classification in the belief function framework, Knowledge-Based Systems 214 (2021) 106742.
URL http://www.sciencedirect.com/science/article/pii/S0950705121000058
[48] T. Denoeux, Decision-making with belief functions: A review, International Journal of Approximate Reasoning 109 (2019) 87–110.
[49] T. Denœux, Logistic regression, neural networks and Dempster-Shafer theory: A new perspective, Knowledge-Based Systems 176 (2019) 54–67.
[50] Z. Tong, P. Xu, T. Denœux, An evidential classifier based on Dempster-Shafer theory and deep learning, Neurocomputing 450 (2021) 275–293.

[51] E. Kerfoot, J. Clough, I. Oksuz, J. Lee, A. P. King, J. A. Schnabel, Left-ventricle quantification using residual u-net, in: International Workshop on Statistical Atlases and Computational Models of the Heart, Springer, 2018, pp. 371–380.

[52] D. Ulyanov, A. Vedaldi, V. Lempitsky, Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 6924–6932.

[53] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.

[54] B. Lowekamp, D. Chen, L. Ibáñez, D. Blezek, The design of SimpleITK, Frontiers in Neuroinformatics 7 (2013) 45. doi:10.3389/fninf.2013.00045.
URL https://www.frontiersin.org/article/10.3389/fninf.2013.00045

[55] Z. Yaniv, B. C. Lowekamp, H. J. Johnson, R. Beare, SimpleITK image-analysis notebooks: a collaborative environment for education and reproducible research, Journal of Digital Imaging 31 (3) (2018) 290–303. doi:10.1007/s10278-017-0037-8.
URL https://doi.org/10.1007/s10278-017-0037-8

[56] C. Guo, G. Pleiss, Y. Sun, K. Q. Weinberger, On calibration of modern neural networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 1321–1330.

[57] A. Jungo, F. Balsiger, M. Reyes, Analyzing the quality and challenges of uncertainty estimations for brain tumor segmentation, Frontiers in Neuroscience 14 (2020) 282.

[58] A.-J. Rousseau, T. Becker, J. Bertels, M. B. Blaschko, D. Valkenborg, Post training uncertainty calibration of deep networks for medical image segmentation, in: 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI), IEEE, 2021, pp. 1052–1056.

[59] W. J. Conover, R. L. Iman, On multiple-comparisons procedures, Tech. Rep. LA-7677-MS, Los Alamos Scientific Laboratory (1979).

[60] A. Dinno, conover.test: Conover-Iman Test of Multiple Comparisons Using Rank Sums, r package version 1.1.5 (2017).
URL https://CRAN.R-project.org/package=conover.test

[61] Y. Benjamini, D. Yekutieli, The control of the false discovery rate in multiple testing under dependency, The Annals of Statistics 29 (4) (2001) 1165–1188.

[62] A. Hryniowski, A. Wong, Deeplabnet: End-to-end learning of deep radial basis networks, Journal of Computational Vision and Imaging Systems 5 (1) (2020) 1.
URL https://openjournals.uwaterloo.ca/index.php/vsl/article/view/1663