An Evidential Deep Network Based on Dempster-Shafer Theory for Large Dataset

Lucas Deregnaucourt
LITIS, Normandie Univ, INSA Rouen,
UNIROUEN, UNIHAVRE
Rouen, France
lucas.deregnaucourt@insa-rouen.fr

Alexis Lechervy
GREYC, Normandie Univ, UMR CNRS 6072,
UNICAEN, ENSICAEN
Caen, France
alexis.lechervy@unicaen.fr

Hind Laghmara
LITIS, Normandie Univ, INSA Rouen,
UNIROUEN, UNIHAVRE
Rouen, France
hind.laghmara@insa-rouen.fr

Samia Ainouz
LITIS, Normandie Univ, INSA Rouen,
UNIROUEN, UNIHAVRE
Rouen, France
samia.ainouz@insa-rouen.fr

***Abstract*—We introduce a novel deep neural network architecture based on Dempster-Shafer theory capable of handling large image datasets with numerous classes, such as ImageNet. Our approach involves analyzing images through multiple experts, composed of convolutional deep neural networks that predict mass functions. These experts are then merged using the Dempster's rule, thereby returning a set of potential classes by selecting the best expected utility based on the previously computed mass functions. Our innovative algorithm can identify the best set of classes among the 2^K possible sets for K classes while maintaining a complexity of $O(K\log(K))$. To illustrate our approach, we apply it to an out-of-distribution example search problem, demonstrating its efficiency.**

***Index Terms*—Dempster-Shafer Theory, Evidence theory, belief function, Deep learning, Out-of-distribution**

I. Introduction

In recent years, image classification has made remarkable strides with the advent of deep neural networks (DNNs). However, high ambiguity in the feature vector may lead to missclassification due to the fact that multiple classes share similar expected probabilities. Moreover, a model only trained for precise classifiation may struggle to detect out-of-distribution (OOD) data.

One promising solution to this problem is set-valued classification [1], [2]. This method allows the model to assign a new data to a non-empty set of classes, particularly when uncertainty is high and precise classification is challenging.

In the context of Out-of-Distribution (OOD) detection, a prevalent approach is the utilization of a classification method with a reject option [3], [4], which can be seen as a special case of set-valued classification. Rejection is defined by assigning a data to the set of all possible classes, indicating a state of high uncertainty.

Recently, several works have sought to integrate the Dempster-Shafer theory (DST) into deep neural networks, aiming to leverage the power of evidential reasoning [5]–[7]. However, these attempts have been confined to relatively small and well-structured datasets such as MNIST [8] or CIFAR-10 [9]. The primary impediment has been the algorithmic complexity of DST, which scales exponentially with the size of the frame of discernment Ω, containing 2^K subsets where $K = |\Omega|$.

Based on [10], [11] proposed an end-to-end deep evidential neural network that allocates mass values only to singletons and Ω. This method addresses this computational bottleneck, effectively reducing the spatial complexity from $O(2^K)$ to $O(K+1)$ for the training phase. Nevertheless, the decision-making process for set-valued classification during the evaluation phase remains a computationally expensive task, requiring an exhaustive selection from all possible subsets of Ω, still operating at $O(2^K)$ complexity. Thus, they selected the possible subsets of Ω based on the distance between the classes derived from the confusion matrix.

We propose in this work an algorithmic solution to mitigate the $O(2^K)$ complexity, making set-valued decisions derived from a mass function output by a Convolutional Neural Network (CNN) feasible with linear complexity without intermediate steps to restrict the number of subsets. Additionally, we introduce mathematical optimizations to enhance numerical computations, enabling scalable implementation of set-valued classification evidential models. These contributions pave the way for the application of the DST theoretical framework to high-dimensional real-world datasets with many classes. They offer significant potential for improving the reliability of deep learning models in various applications such as OOD detection.

The remaining parts of this work are organized as follows. In section II we recall basics of Dempster-Shafer theory. In section III, we present the evidential neural network architecture we use and the algorithmic solution we propose to make set-valued decision in linear complexity. The experiments and preliminary results on large datasets are presented in section

IV. Finally, we conclude in section V.

II. Belief Theory

A. Background on belief functions

Belief function theory, called also Evidence theory or Dempster-Shafer theory [12], [13], is able to model and reason about imprecise and uncertain problems, and has more obvious advantages in the representation and combination of uncertain information.

To represent partial knowledge in the belief function theory, let consider the *frame of discernment* Ω as a finite set of variables ω which refers to K elementary events to a given problem ($\Omega = \{\omega_1, \omega_2, ..., \omega_K\}$).

The power set of Ω is the set of all the 2^K possible subsets. It is presented as follows:

$$2^\Omega = \{\emptyset, \{\omega_1\}, ..., \{\omega_k\}, \{\omega_1, \omega_2\}, \{\omega_1, \omega_3\}, ..., \Omega\}, \quad (1)$$

where the $\{w_i\}$ elements are titled as singletons and $\emptyset$ denotes the empty set.

The key point of Dempster-Shafer theory is the basic belief assignment (*bba*) which represents the partial knowledge about the value of w. A *bba* is a function from 2^Ω to $[0, 1]$ defined as follows:

$$\begin{aligned} m : 2^\Omega &\to [0,1] \\ A &\mapsto m(A) \end{aligned} \quad (2)$$

where m satisfies the following constraint:

$$\sum_{A \subseteq \Omega} m(A) = 1. \quad (3)$$

An element A of Ω is called a *focal element* when $m(A) > 0$, and the set containing all these elements is called a *body of evidence* (BOE). When each element in BOE is a singleton, m is named a *Bayesian bba*. On the other hand, when BOE contains only Ω as a focal element, we are in the *complete ignorance* situation and m is called vacuous belief function. However, when it contains only one singleton of Ω as a focal element, m is presented as a *Certain mass function*.

A *bba* function is normalized when the mass given to the empty set is constrained to be zero ($m(\emptyset) = 0$). In that case, it corresponds to the *closed-world assumption* [13]. A contrary explanation is that the frame of discernment Ω can be incomplete and the value of w can be taken outer Ω. Accordingly, the mass of belief that is not linked to Ω can allowed to be strictly positive ($m(\emptyset) > 0$). That case corresponds to the *open world assumption* [14].

B. Information fusion

The most common way to combine two *bba* m_1 and m_2 defined on the same frame of discernment Ω is the Dempster's rule [13], denoted as $\oplus$. It is defined by $m_{DS}(\emptyset) = 0$ and $\forall A \in 2^\Omega \backslash \{\emptyset\}$ by

$$m_{DS}(A) = (m_1 \oplus m_2)(A) = \frac{1}{1-\kappa} \sum_{\substack{B \cap C = A \\ B, C \in 2^\Omega}} m_1(B) m_2(C) \quad (4)$$

where κ is the degree of conflict between the two sources of evidence defined by:

$$\kappa = \sum_{\substack{B \cap C = \emptyset \\ B, C \in 2^\Omega}} m_1(B) m_2(C).$$

This fusion can be seen as the normalized version of the conjunctive rule which is defined by:

$$m_\cap(A) = \sum_{\substack{B \cap C = A \\ B, C \in 2^\Omega}} m_1(B) m_2(C). \quad (5)$$

C. Decision-making

The most common way of making decisions with belief functions is to apply the pignistic transformation [15] to obtain a probability vector of size K, then the predicted class corresponds to the argmax of this vector. However, such a strategy doesn't allow the model to predict a set of classes. To this end, [16] defines the lower and upper expected utilities of selecting $A \subseteq \Omega$ as follows:

$$\overline{\mathbb{E}}(f_A) = \sum_{B \subseteq \Omega} m(B) \max_{\omega_j \in B} u_{A,j} \quad (6)$$

$$\underline{\mathbb{E}}(f_A) = \sum_{B \subseteq \Omega} m(B) \min_{\omega_j \in B} u_{A,j} \quad (7)$$

where $u_{Aj} \in [0, 1]$ designates the utility of the act of selecting $A \subseteq \Omega$ denoted as f_A when the ground truth is ω_j. The utility matrix $U_{2^{|\Omega|} \times K}$ is computed following [17], [18] with a parameter $\gamma \in [0.5, 1]$ that represents the imprecision tolerance degree. If the true class is ω_j, the utility of assigning a sample to set A is calculated as an Ordered Weighted Average (OWA) aggregation [18] of the individual utilities associated with each precise assignment within A as follows:

$$u_{A,j} = g_{|A|} 1_{\{\omega_j \in A\}} \quad (8)$$

where $g \in \mathbb{R}^{|A|}$ is a weight vector whose elements represent the decision making strategy's tolerance to imprecision. For example if $g = (1, 0, \ldots, 0)$, then the decision making's strategy will be totally intolerant to imprecision, thus forcing the model to output only one class.

Following [17] and [19], this weight vector is obtained by maximizing the following entropy:

$$Ent(g) = \sum_{k=1}^{|A|} \log g_k \quad (9)$$

subject to constraints $\sum_{k=1}^{|A|} g_k = 1$, $\sum_{k=1}^{|A|} \frac{|A| - k}{|A| - 1} g_k = \gamma$ and $g_k \geq 0$ where γ is a parameter representing the tolerance to imprecision. An example of a utility matrix with $\gamma = 0.9$ and $\Omega = \{\omega_1, \omega_2, \omega_3\}$ is shown in Table I. As we can see, the values in the utility matrix are the same according to the cardinality of the selected set. This means that instead of computing every values of the utility matrix, we only need to compute a value U_k for each possible cardinality of the subsets of Ω. In this example, we have $U_1 = 1$, $U_2 = 0.9$ and $U_3 = 0.8263$.

Since we have:

$$\min_{\omega_j \in A} u_{A,j} = \begin{cases} U_k & if\ A = \Omega \\ 0 & else \end{cases} \tag{10}$$

and

$$\max_{\omega_j \in A} u_{A,j} = U_{|A|} \tag{11}$$

the equations (6) and (7) can be simplified as illustrated in section III-C.

	ω_1	ω_2	ω_3
$f_{\{\omega_1\}}$	1	0	0
$f_{\{\omega_2\}}$	0	1	0
$f_{\{\omega_3\}}$	0	0	1
$f_{\{\omega_1,\omega_2\}}$	0.9	0.9	0
$f_{\{\omega_1,\omega_3\}}$	0.9	0	0.9
$f_{\{\omega_2,\omega_3\}}$	0	0.9	0.9
$f_{\{\Omega\}}$	0.8263	0.8263	0.8263

TABLE I
UTILITY MATRIX WITH $\gamma = 0.9$ AND $K = 3$.

The expected utility is then obtained using the generalized Hurwicz decision criterion [20], [21] as follows:

$$\mathbb{E}(f_A) = \nu \underline{\mathbb{E}}(f_A) + (1 - \nu)\overline{\mathbb{E}}(f_A). \tag{12}$$

Where $\nu \in [0, 1]$ is the pessimism index.

When $\gamma = 0.5$, the decision-making strategy is totally intolerant to imprecision so that $u_{ij} = 1$ if $\omega_i = \omega_j$, else $u_{Aj} = 0$. In this sense, we can see the expected utility as a generalized accuracy. The other extreme strategy is the totally tolerant which is achieved when $\gamma = 1$ where $u_{Aj} = 1$ if $\omega_j \in A$, else $u_{Aj} = 0$ so that a model that always outputs Ω will get an expected utility of 1.

We chose this decision-making strategy among all those proposed in [16] since it is the most general form of decision criterion resulting from Jaffray's axioms [21]. Moreover, the expression of the expected utility leads to interesting simplifications in the restricted framework where we only consider the singletons and Ω.

III. SCALABLE EVIDENTIAL NEURAL NETWORK

In this section, we present how the DST framework can be incorporated into a deep neural network architecture. Considering some asumptions on the *bbas* the model will output, we propose an algorithmic solution to make set-valued decision in linear complexity along with mathematical optimizations for a more scalable implementation.

A. Evidential deep neural network

As depicted in Figure 1, the proposed evidential neural network architecture is very similar to a probabilistic one. Our architecture is based on the evidential deep neural network architecture introduced in [11]. The main difference resides in the construction of the mass function. The given image of size $(C \times H \times W)$ first passes through the backbone of a convolutional neural network, resulting in a feature map of size $(C' \times 1 \times 1)$. This feature map captures the data's latent representation.

In the work presented in [11], the construction of mass functions involves the use of a distance-based layer. The classifier is composed of p prototypes t_i in $\mathbb{R}^P$, where P is the dimension of the feature map. In their method, the first step is to compute the distance-based support between the feature map x of a data and each prototype t_i. For the second step, the mass function m_i associated to t_i is computed by multiplying the distance-based support s_i by a weight h_{ij} which characterizes the degree of membership of prototype t_i to the class ω_i.

Our method for constructing the mass functions is more computer vision oriented and is inspired by mixture of experts approaches [22]. Instead of considering prototypes, we consider p experts that see the feature map of a data from different points of view. For this purpose, the classical fully connected layer is replaced by a depthwise convolution [23] with a kernel of size (1×1) and p groups. For a given feature map and a given number of experts p, the depthwise convolution will output a matrix of size $(p \times (K+1))$, namely one mass function per expert. Each mass function holds $|\Omega| + 1$ values, with one value dedicated to each singleton and an another one for the entire set Ω. This vector is then reshaped into a matrix of experts of size $p \times (|\Omega| + 1)$. We apply a softmax activation to satisfy the equation (3). In this matrix, the i-th row represents the mass function associated with expert p_i. The *bbas* of this matrix are then fused with Dempster's rule to obtain a final *bba* of size $|\Omega| + 1$ which we will present in the next section.

B. Computational optimization of Dempster's rule

As seen in the previous section, since our network is only considering the masses assigned to singletons and Ω, the expression of the conjunctive rule simplifies as shown in equation (13).

$$\begin{aligned} m_\cap(A) &= \sum_{\substack{B \cap C = A \\ B,C \in 2^\Omega}} m_1(B)m_2(C) \qquad (13) \\ &= m_1(A)m_2(A) + m_1(A)m_2(\Omega) + m_1(\Omega)m_2(A) \end{aligned}$$

$\forall A \in \Omega$.

This brings us to an iterative algorithm for performing Dempster's rule as shown by the Algorithm 1. We define $\mu_1 = m_1$ and $\mu_{i+1} = m_\cap(\mu_i, m_i)$ where μ_i represents the mass function obtained by the fusion of the i first expert's mass functions by the conjunctive rule.

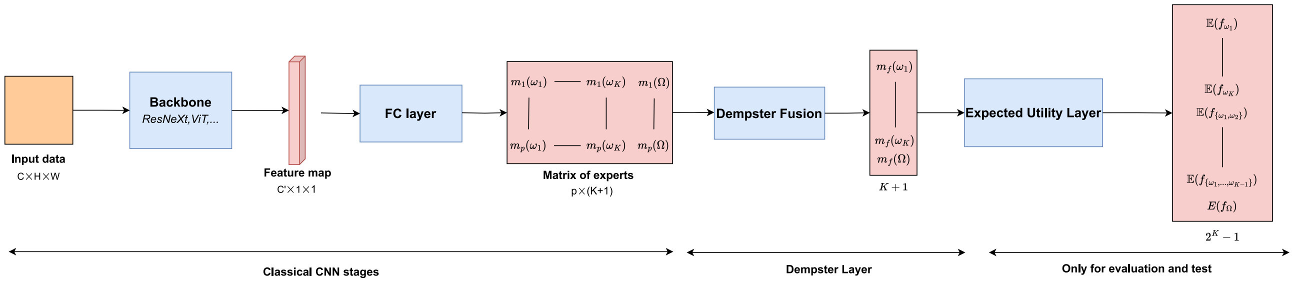

Fig. 1. Architecture of an evidential deep neural network.

Algorithm 1 Iterative Dempster's rule

```
Require: $p$ mass functions $m_1, \ldots, m_p$
  $\mu_1 \leftarrow m_1$
  for $i = 2, \ldots, p$ do
      for $j = 1, \ldots, K$ do
          $\mu_i(\{\omega_j\}) = \mu_{i-1}(\{\omega_j\})m_i(\{\omega_j\}) + \mu_{i-1}(\{\omega_j\})m_i(\Omega) + \mu_{i-1}(\Omega)m_i(\{\omega_j\})$
      end for
      $\mu_i(\Omega) = \mu_{i-1}(\Omega)m_i(\Omega)$
  end for
  return $\mu_p/Z$
  where $Z$ is a normalization term.
```

The expression of $\mu_i(\{\omega_j\})$ can be rewriten as follows:

$$\begin{aligned}\mu_i(\{\omega_j\}) &= \mu_{i-1}(\{\omega_j\})m_i(\{\omega_j\}) + \mu_{i-1}(\{\omega_j\})m_i(\Omega) \\ &\quad + \mu_{i-1}(\Omega)m_i(\{\omega_j\}) \\ &= \left(\mu_{i-1}(\{\omega_j\}) + \mu_{i-1}(\Omega)\right) \times \left(m_i(\{\omega_j\}) + m_i(\Omega)\right) \\ &\quad - \mu_{i-1}(\Omega)m_i(\Omega)\end{aligned} \tag{14}$$

which leads to an improved algorithm that only iterates on the number of classes K as presented in the Algorithm 2.

Algorithm 2 Scalable Dempster's rule

```
Require: $p$ mass functions $m_1, \ldots, m_p$
  $\mu_p(\Omega) = \prod_{i=1}^{p} m_i(\Omega)$
  for $j = 1, \ldots, K$ do
      $\mu_p(\{\omega_j\}) = \prod_{i=1}^{p} (m_i(\{\omega_j\}) + m_i(\Omega)) - \mu_p(\Omega)$
  end for
  return $\mu_p/Z$ where $Z$ is a normalization term.
```

The algorithm 2 is highly parallelizable and each element of the loop can be calculated independently of the others, unlike the algorithm 1 where each element depends on the previous iteration. In practice, this second algorithm provides a very fast implementation of Dempster's rule in the restricted framework chosen where we only consider singletons and Ω.

C. Scalable decision making

Since we only consider the singletons and Ω for the construction of the mass function, we can simplify the equations (6) and (7) as follows:

$$\overline{\mathbb{E}}(f_A) = \sum_{\omega_i \in \Omega} \left(m(\{\omega_i\})u_{A,i}\right) + m(\Omega) \max_{\omega_k \in \Omega} u_{A,k}, \tag{15}$$

$$\underline{\mathbb{E}}(f_A) = \sum_{\omega_i \in \Omega} \left(m(\{\omega_i\})u_{A,i}\right) + m(\Omega) \min_{\omega_k \in \Omega} u_{A,k}. \tag{16}$$

During the training phase, we want f_A to be a singleton. That's to say $u_{ii} = 1$ and $u_{ij} = 0 \ \forall i \neq j$ which can be seen as the classical accuracy metric. Under those hypotheses, we can simplify the equations (15) and (16) as follows:

$$\overline{\mathbb{E}}(f_{\omega_i}) = m(\{\omega_i\}) + m(\Omega) \tag{17}$$

$$\underline{\mathbb{E}}(f_{\omega_i}) = m(\{\omega_i\}) \tag{18}$$

leading to this simplified expression of the expected utility:

$$\begin{aligned}\mathbb{E}(f_{\omega_i}) &= \nu m(\{\omega_i\}) + (1-\nu)\left(m(\{\omega_i\}) + m(\Omega)\right) \\ &= m(\{\omega_i\}) + (1-\nu)m(\Omega).\end{aligned} \tag{19}$$

This expression can be considered as a rewriting of the pignistic transformation in our restricted framework. Indeed, taking $\nu = 1 - \frac{1}{|\Omega|}$ in equation (19) leads to the pignistic probability expression when $m(A) = 0 \ \forall A \subset \Omega$ such that $|A| \geq 2$.

We propose to use the cross-entropy loss on the expected utilities vector for training our network:

$$-\sum_{i=1}^{n} \sum_{k=1}^{K} y_{i,k} \log \left(\mathbb{E}(f_{\omega_k}(x_i))\right) \tag{20}$$

with n is size of training dataset, $y_{i,k}$ is 1 if the label of example x_i is ω_k and 0 otherwise.

For decision-making during the evaluation and test phase, we want our network to be able to output a subset of Ω. The main obstacle is the algorithmic complexity since it would require to compute $2^{|\Omega|}$ expected utilities to choose the subset that maximizes it. To solve this issue, [11] proposes to compute the confusion matrix from the training set generated by an evidential deep neural network as explained above. Based on the distance between the classes, they only keep the classes and groups of classes that are similar enough by thresholding. Although in practice this strategy reduces the number of expected utilities to be computed, it remains in $2^{|\Omega|}$ in the worst case (when the result is to be attributed to the Ω set). Furthermore, we are not convinced that this strategy is sufficient to scale to databases with a large number of classes such as ImageNet [24] where $|\Omega|$=1000. Moreover, it requires a costly intermediate step between the training phase and the evaluation and test phases.

To this end, we propose a very simple and computationally efficient iterative algorithm (3) to determine the argmax between all subsets of Ω without any *a priori* about the correlation between the classes nor intermediate step to restrict the number of subsets of Ω. The first step is to compute the expected utilities of singletons using the equation (19) and to sort them in a decreasing order. We then compare the higher singleton expected utility with the expected utility of the subset composed of the two best singletons using the equations (12),(15),(16) and so on until adding a new singleton to the subset decreases the expected utility. Let's consider $\Omega = \{\omega_1, \omega_2, \omega_3\, \omega_4\}$ with $\mathbb{E}(\omega_1) > \mathbb{E}(\omega_2) > \mathbb{E}(\omega_3) > \mathbb{E}(\omega_4)$. We then compute $\mathbb{E}(\{\omega_1, \omega_2\})$ and compare it with $\mathbb{E}(\omega_1)$. Let's suppose that $\mathbb{E}(\{\omega_1, \omega_2\})$ is effectively higher than $\mathbb{E}(\omega_1)$, we now have to compute $\mathbb{E}(\{\omega_1, \omega_2, \omega_3\})$. By considering that $\mathbb{E}(\{\omega_1, \omega_2\}) > \mathbb{E}(\{\omega_1, \omega_2, \omega_3\})$, we obtain $A^\star = \{\omega_1, \omega_2\}$. If $\mathbb{E}(A^\star) > \mathbb{E}(\Omega)$ then the model outputs $A^\star$, else it ouputs Ω.

Algorithm 3 Argmax of the Expected Utility

```
Require: sorted singletons expected utilities $\mathbb{E}(\{\omega_{\alpha_1}\}) \geq \mathbb{E}(\{\omega_{\alpha_2}\}) \geq \ldots \geq \mathbb{E}(\{\omega_{\alpha_K}\})$.
  $A^\star \leftarrow \omega_{\alpha_1}$
  for $i = 2, \ldots, K$ do
    $A^\star_{temp} \leftarrow \{A^\star, \omega_{\alpha_i}\}$
    if $\mathbb{E}(A^\star_{temp}) > \mathbb{E}(A^\star)$ then  $A^\star \leftarrow A^\star_{temp}$
    end if
  end for
  return $A^\star$
```

This strategy allows the model to output a set of classes among all the possible subsets of Ω while maintaining a complexity of $O(K \log(K))$ without requiring any limitations on the number of subsets of Ω to compare their expected utilities.

IV. Experiments

To demonstrate the relevance of our model, we conducted several experiments. Firstly, we carry out a study on the impact of the various parameters on our model. Secondly, we sought to demonstrate the ability of our model to process large databases containing a large number of classes and compare our model with a standard problistic model for classification problem. Finally, we demonstrated the superiority of our approach over the standard probabilistic model for an OOD detection task.

In all our experiments, we assume that the backbone used is of type ResNext50 [25]. This applies both to our model and to the probabilistic models to which the comparison is conducted.

A. Datasets

We conducted our experiments using the following 3 databases: CIFAR-100, ImageNet and SVHN dataset.

CIFAR-100 [26] is a database of low-resolution 28 × 28 images. It contains $60,000$ images divided into 100 classes with 600 images per class.

ImageNet [24] contains 1.5 million images of 224×224 resolution, manually annotated in $1,000$ categories. The annotation is based on the WordNet hierarchical object categorisation structure (augmented by 120 dog categories).

The SVHN (Street View House Numbers) database [27] is a collection of 32 × 32 digital images that includes handwritten digits from photos of house numbers taken in street scenes. The database contains 10 classes, corresponding to digits from 0 to 9.

B. Ablation study

In this section, we present some experiments designed to measure the impact of the various parameters of our approach on its performances. We measure two metrics: *expected utility* and *average cardinality*.

Given that the accuracy is obtained by fixing the imprecision tolerance degree γ to 0.5 while computing the expected utility, we propose to evaluate the *expected utilities* across a range of γ values from 0.5 to 0.95.

We compute the *average cardinality* of the predictions according to γ as follows:

$$AC(T) = \frac{1}{|T|}\sum_{i=1}^{|T|} |A(i)| \qquad (21)$$

where $T = \{x_1, \ldots, x_{|T|}\}$ is the test set and $A(i)$ is the set-valued output for the data $x_i \in T$. It is clear that for $\gamma = 1$, the model will always output f_Ω since $\mathbb{E}(\Omega) = 1$ and the *average cardinality* will be equal to the number of elements in Ω.

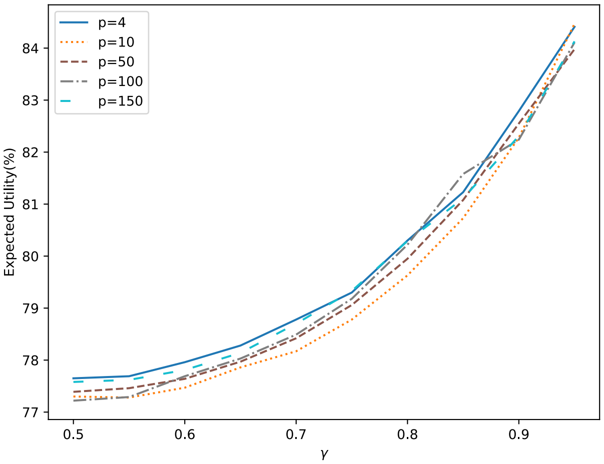

Fig. 2. Expected Utility according to the number of experts on CIFAR-100.

Firstly, we need to determine the hyperparameters of our model, namely the number of experts p and the degree of pessimism ν. Since this search process is quite time-intensive, we restrict it to the CIFAR-100 dataset. To identify the optimal number of experts, we fix ν to 0.99 so that the equation(19) corresponds to the pignistic probability. As shown on Figure 2, the impact of the number of experts does not appear to be significant. This is mainly because there is no guarantee that the experts simulated by the fully connected layer will be independent. So we choose $p = 4$ as there is no need for a lot of experts. Then we search for the optimal ν by setting the number of experts $p = 4$. As depicted in Figure 3, the model learns in a similar way, independently of ν. Indeed, the model always outputs a value very close to zero for $m(\Omega)$ for precise classification task, so the impact of ν is not significant during the training phase. Consequently, we have selected $\nu = \frac{1}{|\Omega|}$, namely $\nu = 0.99$ for CIFAR-100 and $\nu = 0.999$ for ImageNet.

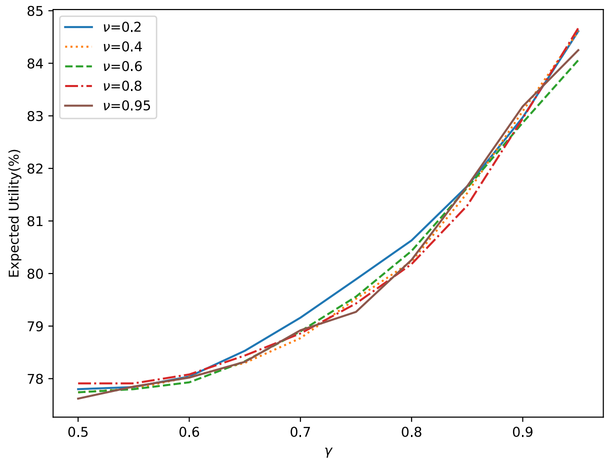

Fig. 3. Expected Utility according to ν on CIFAR-100.

C. Comparison with probabilistic approaches for image classification

Now that we have fixed the model hyperparameters, we can compare the evidential neural network with the probabilistic one on precision classification. As mentioned previously, the probabilistic model used corresponds to a ResNext50 type backbone. This is followed by a fully connected layer and a softmax.

For fair comparison between our method and the probabilistic approach, we have to allow the probabilistic network to output set-valued predictions in order to compute the expected utility. To do so, we considere the probability vector output by the model as a mass function with $m(\Omega) = 0$ and $m(\{\omega_j\}) = p(\omega_j)\ \forall j = 1, \ldots, K$.

The Expected Utility and Cardinality curves over 10 runs on CIFAR-100 are respectively presented in Figure 4 and Figure 5. The Expected Utility and Cardinality curves on ImageNet are respectively presented in Figure 6 and Figure 7. Due to the size of the database, we limited the ImageNet experiments to a single run and were therefore unable to calculate standard deviations. For both experiments, we can see that there is almost no difference between the two models from $\gamma = 0.5$ to $\gamma = 0.7$ where the decision-making strategy is quite intolerant to uncertainty, forcing the model to output one or two classes. For $\gamma = 0.75$ to $\gamma = 0.95$ the evidential model is less confident than the probabilistic one and outputs sets with a higher cardinality, which decreases the Expected Utility. On Imagenet the performance of the probabilistic model is 77.77% in accuracy against 77.65%. The difference in performance is relatively small.

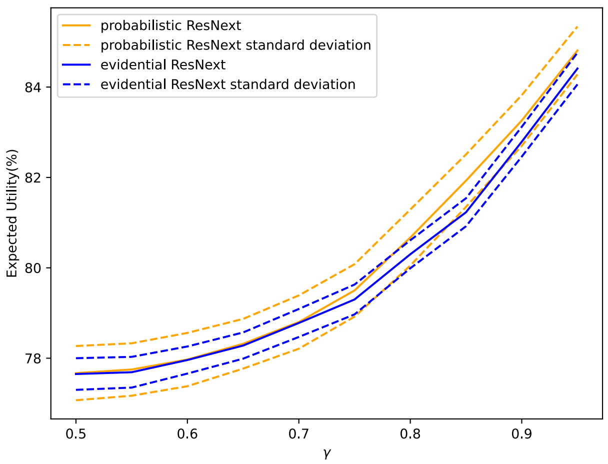

Fig. 4. Expected Utility on CIFAR-100.

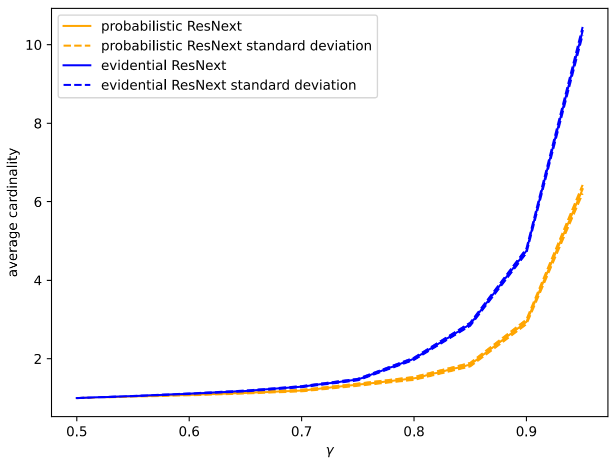

Fig. 5. Average Cardinality on CIFAR-100.

D. OOD detection

For OOD detection task, we want to evaluate the capability of the network to output Ω if, and only if, the data does not belong to the classes from the training set. For this purpose, we evaluate the rate of f_Ω by varying γ from 0.5 to 0.95. A good model has to get a high rate of f_Ω on out-of-distribution data and a low rate of f_Ω on in-distribution data. For $\gamma = 1$, the model will always predict Ω since all the non-zero values in the utility matrix will be equal to 1. So the f_Ω rate will always be equal to 100%.

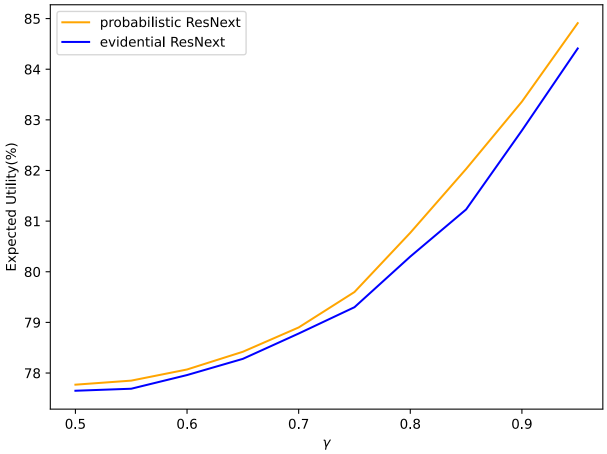

Fig. 6. Expected Utility on ImageNet.

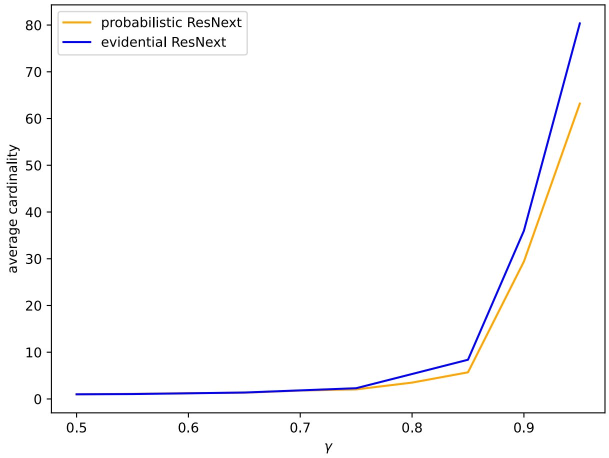

Fig. 7. Average Cardinality on ImageNet.

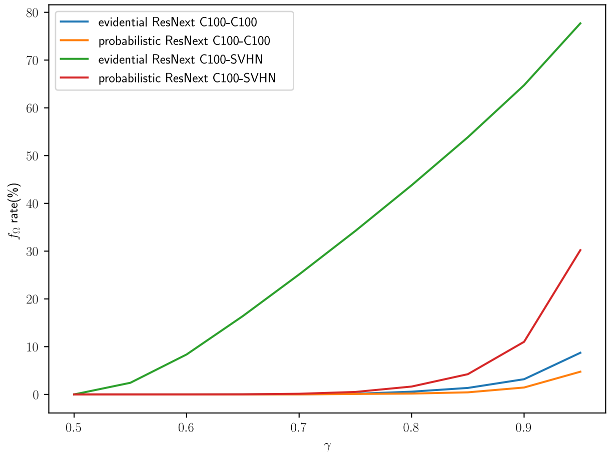

Fig. 8. f_Ω rate for OOD detection, CIFAR-100.

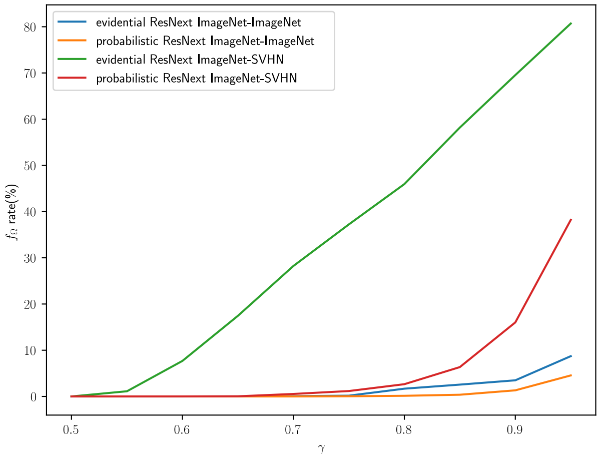

Fig. 9. f_Ω rate for OOD detection, ImageNet.

The results on the OOD detection task for the models trained on CIFAR-100 and ImageNet are respectively presented in Figure 8 and Figure 9. As expected, the f_Ω rate is very low for the evidential and the probabilistic models on in-distribution test set. However, it is clear that the evidential network outperforms the probabilistic network for OOD detection task when we evaluate them on the SVHN dataset.

V. Discussions and conclusions

In this work, we have presented a novel deep neural network based on Dempster-Shafer theory capable of handling large datasets for image classification. Furthermore, we have introduced mathematical optimizations to improve numerical computations, facilitating a scalable implementation of evidential models for set-valued classification. This approach makes it possible to obtain results on databases with a large number of classes, while avoiding the problem of traversing the 2^K subset of possible classes.

The proposed evidential neural network shows similar results to the probabilistic one for precise classification task. One way to improve it can be to ensure the independence of the experts with a Deep Ensemble approach [28], [29].

However, our network clearly outperforms the probabilistic one for OOD detection task regarding the f_Ω rate. This illustrates that the proposed method overcomes one of the main problems of neural networks, namely the overconfidence even if the data is out-of-distribution. Of course, the scope of our method does not limit itself to image classification. We can adapt it to other computer vision tasks such as semantic segmentation and instance segmentation.

Another way of improving our method would be to also take into account the partial ignorance of the experts when fusing the mass functions and making a decision. This would require to overcome computational bottlenecks but would open the doors for other decision-making strategies and more optimal fusion rules.

Acknowledgements

The first author is supported by the French National Research Agency (ANR) and Region Normandie under grant HAISCoDe. We also thank our colleagues from the Criann who provided us some computation resources with Myria and Austral and by so enabled us to get our results efficiently and fast.

References

[1] E. Chzhen, C. Denis, M. Hebiri, and T. Lorieul, "Set-valued classification – overview with a unified framework," *arXiv preprint arXiv:2102.12318*, 2021.

[2] E. Grycko, "Classification with set-valued decision functions," *Information and Classification (O. OPITZ, B. LAUSEN and R. KLAR, eds.)*, pp. 218–224, 1993.

[3] C. Chow, "On optimum recognition error and reject tradeoff," *IEEE Transactions on Information Theory*, vol. 16, no. 1, pp. 41–46, 1970.

[4] M. Pimentel, D. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014.

[5] S. Xu, Y. Chen, C. Ma, and X. Yue, "Deep evidential fusion network for medical image classification," *International Journal of Approximate Reasoning*, vol. 150, pp. 188–198, 2022.

[6] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," *Advances in Neural Information Processing Systems*, vol. 31, pp. 3179–3189, 2018.

[7] Z. Guo, Z. Wan, Q. Zhang, X. Zhao, Q. Zhang, L. Kaplan, A. Jøsang, D. Jeong, F. Chen, and J.-H. Cho, "A survey on uncertainty reasoning and quantification in belief theory and its application to deep learning," *Information Fusion*, vol. 101, 2024.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[9] A. Krizhevsky, "Learning multiple layers of features from tiny images," *Ph.D. dissertation*, 2009.

[10] T. Denoeux, "A neural network classifier based on dempster-shafer theory," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 30, no. 2, pp. 131–150, 2000.

[11] Z. Tong, P. Xu, and T. Denœux, "An evidential classifier based on dempster-shafer theory and deep learning," *Neurocomputing*, vol. 450, pp. 275–293, 2021.

[12] A. Dempster, "Upper and lower probabilities induced by a multivalued mapping," *The Annals of Mathematical Statistics*, vol. 38, no. 2, pp. 325–339, 1967.

[13] G. Shafer, *A mathematical theory of evidence*. Princeton University Press, 1976.

[14] P. Smets, "The combination of evidence in the transferable belief model," *Transactions on pattern analysis and machine intelligence*, vol. 12, no. 2, pp. 447–458, 1990.

[15] P. Smets and R. Kennes, "The transferable belief model," *Artificial Intelligence*, vol. 66, pp. 191–234, 1994.

[16] T. Denœux, "Decision-making with belief functions: a review," *International Journal of Approximate Reasoning*, vol. 109, pp. 87–110, 2019.

[17] L. Ma and T. Denœux, "Partial classification in the belief function framework," *Knowledge-Based Systems*, vol. 214, p. 106742, 2021.

[18] R. Yager, "On ordered weighted averaging aggregation operators in multicriteria decision-making," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, pp. 183–190, 1988.

[19] M. O'Hagan, "Aggregating template or rule antecedents in real-time expert systems with fuzzy set logic," in *Twenty-Second Asilomar Conference on Signals, Systems and Computers*, vol. 2, 1988, pp. 681–689.

[20] L. Hurwicz, "The generalized bayes minimax principle: a criterion for decision making under uncertainty," *cowles Commission Discussion Paper Statistics*, vol. 355, 1951.

[21] J.-Y. Jaffray, "Linear utility theory for belief functions," *Operations Research Letters*, vol. 8, no. 2, pp. 107–112, 1989.

[22] T. Baldacchino, E. J. Cross, K. Worden, and J. Rowson, "Variational bayesian mixture of experts models and sensitivity analysis for nonlinear dynamical systems," *Mechanical Systems and Signal Processing*, vol. 66-67, pp. 178–200, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327015002307

[23] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

[25] S. Xie, R. Girshick, P. Doll ́ar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5987–5995, 2017.

[26] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Technical report, University of Toronto, Toronto, Ontario, Tech. Rep. 0, 2009, backup Publisher: University of Toronto. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[27] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011. [Online]. Available: http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf

[28] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ense," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, and R. Ferg, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/9ef2ed4b7fd2

[29] O. Laurent, A. Lafage, E. Tartaglione, G. Daniel, J.-M. Martinez, A. Bursuc, and G. Franchi, "Packed-ensembles for efficient uncertainty estimation," in *ICLR*, 2023.