Evidential deep neural network in the framework of Dempster-Shafer theory

Zheng Tong

Université de technologie de Compiègne
HEUDIASYC (UMR CNRS 7253)

Supervisors: Thierry Denœux and Philippe Xu

Problems in DNNs

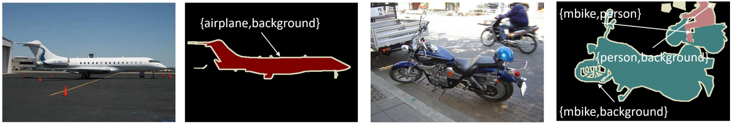
- Deep neural networks (DNNs) achieve state-of-the-art results in many applications:
    - Object classification
    - Semantic segmentation
    - ...
- Such achievements are due to their reliable feature representations with multiple layers, which progressively extract high-level features from raw data.
- However, they still face the problems of data uncertainty.

Data uncertainty

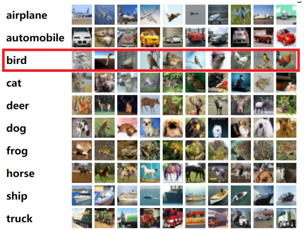
1. Ambiguous raw data and their representations → incorrect decision

 /cat or dog?  /cat species?

2. Imprecise and unreliable data → effects on learning systems



3. Incomplete data → difficulty in novelty detection and model fusion



Objectives

- Many theories have been combined with DNNs to solve these uncertainty problems:
  - Bayesian probability
  - Imprecise probability
  - Fuzzy sets
  - Dempster-Shafer (DS) theory
  - ...
- The DS theory of belief functions, also referred to as evidence theory, is applied to a wide range problems involving uncertainty in machine learning.

Key features of DS theory in machine learning

Generality: DS theory is based on the idea of combining sets and probabilities. It extends both

- Probabilistic reasoning
- Propositional logic, computing with sets (interval analysis)

DS theory can do much more than sets or probabilities.

Operationality: DS theory is easily put in practice by breaking down the available evidence into elementary pieces of evidence, and combining them by a suitable operator called Dempster's rule of combination.

- We aim to develop new DNNs based on DS theory with the capacity to deal with data uncertainty.

Outline

Mass, belief, and plausibility functions

- Let $\Omega = \{\omega_1, \ldots, \omega_M\}$ be a class set called the frame of discerment.
- A mass function on Ω is a mapping $m : 2^\Omega \to [0, 1]$ such that

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

  If $m(\emptyset) = 0$, m is said to be normalized.
- Every subset $A \subseteq \Omega$ such that $m(A) > 0$ is called a focal set of m.
- Belief and plausibility functions are defined as

$$Bel(A) = \sum_{B \subseteq A} m(B), \quad Pl(A) = \sum_{B \cap A \neq \emptyset} m(B).$$

Dempster's rule of combination

- Two independent mass functions m_1 and m_2 on Ω is combined as their orthogonal sum

$$(m_1 \oplus m_2)(A) := \frac{\sum_{B\cap C=A} m_1(B)m_2(C)}{1-\sum_{B\cap C=\emptyset} m_1(B)m_2(C)}$$

  for all $A \neq \emptyset$ and $(m_1 \oplus m_2)(\emptyset) = 0$.

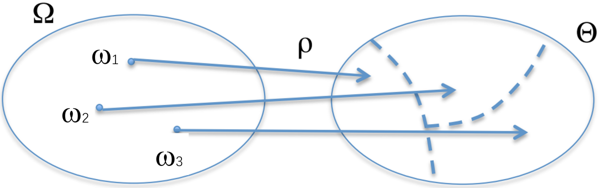
- Property w.r.t normalized contour function pl:

$$pl(\omega) = Pl(\omega),\ \forall \omega \in \Omega.$$

$$p_m(\omega) := \frac{pl(\omega)}{\sum_{j=1}^{M} pl(\omega_j)},$$

$$p_{m_1\oplus m_2}(\omega) \propto p_{m_1}(\omega)p_{m_2}(\omega), \quad \omega \in \Omega,$$

Refinement



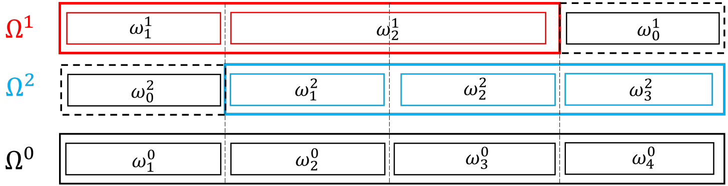
Definition

A frame Θ is a refinement of Ω iff there is a mapping $\rho : 2^{\Omega} \rightarrow 2^{\Theta}$ such that:

- $\{\rho(\{\omega\}), \omega \in \Omega\} \subseteq 2^{\Theta}$ is a partition of Θ,
- $\forall A \subseteq \Omega, \rho(A) = \bigcup_{\omega \in A} \rho(\{\omega\})$.

Compatible frames and vacuous extension

- Two frames of discernment are said to be compatible if they have a common refinement.
- In machine learning, add an "anything else" element in different frames to make them compatible.



- $m^{\Omega^1 \uparrow \Omega^0}$ is called the vacuous extension of m^{Ω^1} on Ω^0, such that

$$m^{\Omega^1}(\{\omega_1^1\}) = m^{\Omega^0}(\{\omega_1^0\}), \quad m^{\Omega^1}(\{\omega_2^1\}) = m^{\Omega^0}(\{\omega_2^0, \omega_3^0\}),$$

$$m^{\Omega^1}(\{\omega_0^1\}) = m^{\Omega^0}(\{\omega_4^0\}).$$

Definitions and notations

- A decision problem with a set of states of the nature Ω is formalized:
  - A set of acts $\mathcal{F}$
  - A utility function $u : \mathcal{F} \times \Omega \to \mathbb{R}$, such that $u_{f,\omega}$ is the utility of selecting act $f \in \mathcal{F}$ when the true state is ω.

	$u_{f_i,1}$	$u_{f_i,2}$	$u_{f_i,3}$	$\min u_{f_i,j}$	$\max u_{f_i,j}$
f_1	0.37	0.25	0.23	0.23	0.37
f_2	0.49	0.70	0.20	0.20	0.70

- With a mass function of DS theory m describing the uncertainty on Ω, the lower and upper expected utilities of act f is defined as

$$\underline{\mathbb{E}}_m(f) = \sum_{B \subseteq \Omega} m(B) \min_{\omega_j \in B} u_{f,j}, \quad \overline{\mathbb{E}}_m(f) = \sum_{B \subseteq \Omega} m(B) \max_{\omega_j \in B} u_{f,j}.$$

- The generalized Hurwicz expected utility is a weighted average of lower and upper expected utilities

$$\mathbb{E}_{m,\nu}(f) = \nu \underline{\mathbb{E}}_m(f) + (1-\nu)\overline{\mathbb{E}}_m(f).$$

Precise and imprecise classification with belief functions

- A problem of precise classification can be formalized as
  - A set of acts $\mathcal{F} = \{f_{\omega_1}, \ldots, f_{\omega_M}\}$
  - A utility function u described by a utility matrix $\mathbb{U}_{M\times M}$ with general term u_{ij}

	Class		
	ω_1	ω_2	ω_3
f_{ω_1}	1	0	0
f_{ω_2}	0	1	0
f_{ω_3}	0	0	1

- A problem of imprecise classification can be formalized as
  - A set of acts is $\mathcal{F} = \{f_A, A \in 2^{\Omega} \backslash \emptyset\}$
  - A utility function u described by a utility matrix $\mathbb{U}_{(2^{\Omega}-1)\times M}$ with general term $\widehat{u}_{A,j}$
- How to extend $\mathbb{U}_{M\times M}$ to $\mathbb{U}_{(2^{\Omega}-1)\times M}$?

Ordered weighted average aggregation

- Term $\widehat{u}_{A,j}$ is an ordered weighted average aggregation of the utilities of each precise assignment in A as

$$\widehat{u}_{A,j} = \sum_{k=1}^{|A|} g_k \cdot u_{(k)j}^{A}.$$

- Parameters g_k are determined to maximize the entropy subject to

$$\sum_{k=1}^{|A|} \frac{|A|-k}{|A|-1} g_k = \gamma.$$

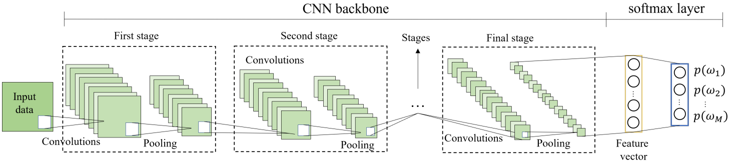
- γ measures the tolerance to imprecision; it controls the imprecision of the decisions:
  - $\gamma = 0.5$ gives the average (minimum tolerance degree)
  - $\gamma = 1$ gives the maximum (maximum tolerance degree)

Example of $\mathbb{U}_{(2^{\Omega}-1)\times M}$ with $\gamma = 0.8$

	Classes		
	ω_1	ω_2	ω_3
$f_{\{\omega_1\}}$	1	0	0
$f_{\{\omega_2\}}$	0	1	0
$f_{\{\omega_3\}}$	0	0	1
$f_{\{\omega_1,\omega_2\}}$	0.8	0.8	0
$f_{\{\omega_1,\omega_3\}}$	0.8	0	0.8
$f_{\{\omega_2,\omega_3\}}$	0	0.8	0.8
f_{Ω}	0.682	0.682	0.682

Outline

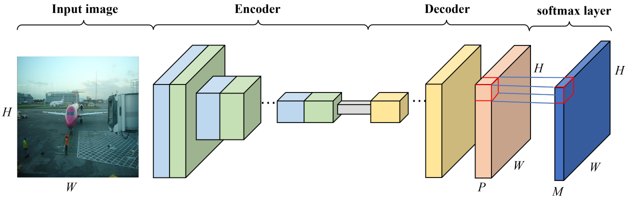
Probabilistic CNN classifier for object classification



- A CNN stage is a combination of convolutional and pooling layers.
- A CNN backbone is composed of at least one stage for feature extraction.
- A probabilistic CNN classifier converts the feature vector from a backbone into a probability distribution using a softmax layer for decision-making.

Probabilistic FCN model for semantic segmentation

- A FCN backbone (encoder-decoder architecture) extracts pixel-wise feature maps from an input image.

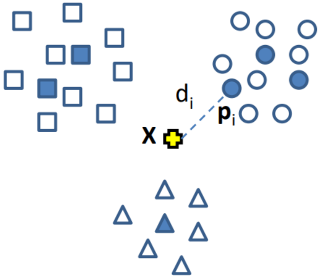


- An encoder-decoder architecture consists of
  - CNN stages to extract features from the input image
  - Upsampling layers to upsample features into pixel-wise feature maps
- How to transform the features from a backbone into mass functions?

Outline

1. Background
   - Dempster-Shafer theory
   - Deep neural network
   - Evidential neural network
2. Evidential deep neural networks
   - Object classification
   - Semantic segmentation
3. Evidential multi-model fusion

Principle



- A learning set is summarized by n prototypes in the form of feature vectors.
- Each prototype $\boldsymbol{p}^i$ has membership degree h_j^i to each class ω_j with $\sum_{j=1}^{M} h_j^i = 1$.
- Each prototype $\boldsymbol{p}^i$ is a piece of evidence about the class of $\boldsymbol{x}$; its reliability decreases with the distance d^i between $\boldsymbol{p}^i$ and $\boldsymbol{x}$.

Propagation equations

- Mass functions associated to $\boldsymbol{p}^i$:

$$m^i(\{\omega_j\}) = h^i_j \tau^i \exp(-(\eta^i d^i)^2)$$
$$m^i(\Omega) = 1 - \tau^i \exp(-(\eta^i d^i)^2)$$

- Combination:

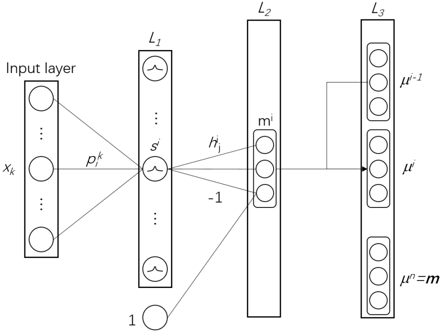
$$m = \bigoplus_{i=1}^{n} m^i$$

- The combined mass function:

$$\boldsymbol{m} = (m(\omega_1), \ldots, m(\omega_M), m(\Omega))^T$$

Evidential neural network (DS layer)

- Evidential classifier can be implemented as neural network layer, called a DS layer.



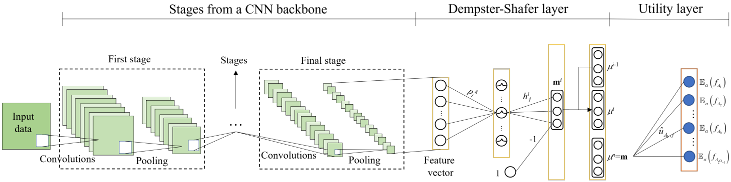
- The performance of an evidential classifier heavily depends on its input feature vector.

Outline

1. Background
   - Dempster-Shafer theory
   - Deep neural network
   - Evidential neural network
2. Evidential deep neural networks
   - Object classification
   - Semantic segmentation
3. Evidential multi-model fusion

Network architecture

- Aim to solve the uncertainty problems in object classification.
- Basic idea: plug a "DS layer" at the output of a CNN backbone, called an "evidential CNN classifier (E-CNN)".



- The decision-making process with mass functions and utility theory is implemented as a neural network layer, called a utility layer.
- The connection weights in the utility layer is $\widehat{u}_{A,j}$ and do not need to be updated during training because $\widehat{u}_{A,j}$ depends on the tolerance to imprecision γ.

Learning

- Given a sample $\boldsymbol{x}$ with class label ω_*, using the generalized Hurwicz criterion, the prediction loss is defined as

$$\mathcal{L}_\nu(m,\omega_*) = -\sum_{k=1}^{M} y_k \log \mathbb{E}_{m,\nu}(f_{\omega_k}) + (1-y_k)\log(1-\mathbb{E}_{m,\nu}(f_{\omega_k}))$$

where y_k equals 1 if $\omega_k = \omega_*$, otherwise 0.

- The loss $\mathcal{L}_\nu(m,\omega_*)$ is minimized when $\mathbb{E}_{m,\nu}(f_{\omega_k}) = 1$ for $\omega_k = \omega_*$ and $\mathbb{E}_{m,\nu}(f_{\omega_l}) = 0$ if $\omega_l \neq \omega_*$.

Examples	Outputs of a DS layer			
	$m(\{\omega_1\})$	$m(\{\omega_2\})$	$m(\{\omega_3\})$	$m(\Omega)$
#1	0.70	0.10	0.10	0.10
#2	0.97	0.01	0.01	0.01
#3	0.50	0.50	0	0
#4	0.40	0.40	0	0.2

Examples	Expected utility			Loss ($\omega_* = \omega_1$)
	$\mathbb{E}_{m,1}(\{\omega_1\})$	$\mathbb{E}_{m,1}(\{\omega_2\})$	$\mathbb{E}_{m,1}(\{\omega_3\})$	
#1	0.70	0.10	0.10	0.303
#2	0.97	0.01	0.01	0.026
#3	0.50	0.50	0	0.602
#4	0.40	0.40	0	0.796

- In practice, the error propagation can be performed automatically in TensorFlow.

Evaluation metrics for classification performance

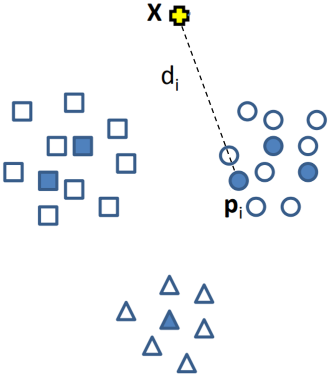
- Averaged utility measures the utilities of all assignments in testing set T:

$$AU(T) = \frac{1}{|T|} \sum_{i=1}^{|T|} \widehat{u}_{A(i),y_i}$$

- When only considering precise acts, AU is equal to classification accuracy.
- Averaged cardinality measures the imprecision of the decisions in T:

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

Evaluation metrics for novelty detection

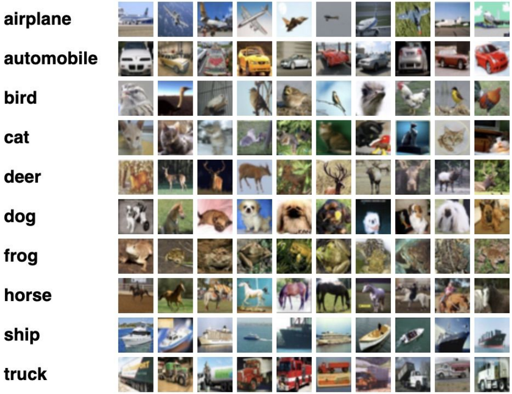


- An outlier $\boldsymbol{x}$ has large d^i to each prototypes.
- The DS layer outputs $m(\Omega) \approx 1$ for $\boldsymbol{x}$.
- The final decision is act f_Ω for $\boldsymbol{x}$ and set Ω means "everything".
- A good classifier should have a high rate of assignment f_Ω in an outlier testing set and a low rate of assignment f_Ω in an inlier testing set.

Dataset in the image-classification experiment

CIFAR-10 to train and evaluate classification performance:

- 10 classes
- 5000 tiny images of each class for training and validation
- 1000 tiny images of each class for testing

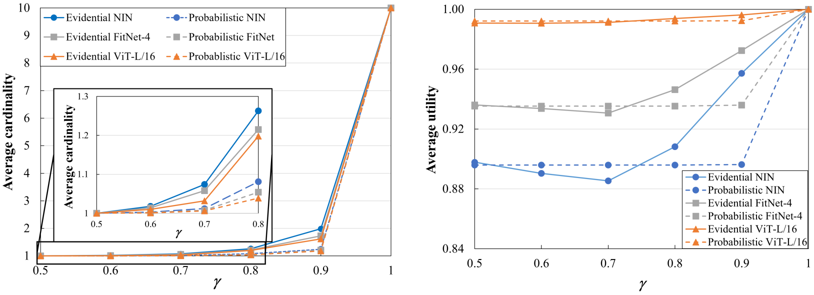


Results of precise classification

NIN	FitNet-4	ViT-L/16
Input: 32 × 32 × 3		
		16 × 16 × 3 × 4 patches
5 × 5 Conv. NIN 64 *ReLU*	3 × 3 Conv. 32 *ReLU* 3 × 3 Conv. 32 *ReLU* 3 × 3 Conv. 32 *ReLU* 3 × 3 Conv. 48 *ReLU* 3 × 3 Conv. 48 *ReLU*	3 × 3 Conv. 32 *ReLU* 3 × 3 Conv. 32 *ReLU* 3 × 3 Conv. 32 *ReLU* 3 × 3 Conv. 48 *ReLU* 3 × 3 Conv. 48 *ReLU*
2 × 2 max-pooling with 2 strides		
5 × 5 Conv. NIN 64 *ReLU* 2 × 2 mean-pooling with 2 strides	3 × 3 Conv. 80 *ReLU* 3 × 3 Conv. 80 *ReLU* 3 × 3 Conv. 80 *ReLU* 3 × 3 Conv. 80 *ReLU* 3 × 3 Conv. 80 *ReLU*	3 × 3 Conv. 80 *ReLU* 3 × 3 Conv. 80 *ReLU* 3 × 3 Conv. 80 *ReLU* 3 × 3 Conv. 80 *ReLU* 3 × 3 Conv. 80 *ReLU*
2 × 2 max-pooling with 2 strides		
5 × 5 Conv. NIN 128 *ReLU* 2 × 2 mean-pooling with 2 strides	3 × 3 Conv. 128 *ReLU* 3 × 3 Conv. 128 *ReLU* 3 × 3 Conv. 128 *ReLU* 3 × 3 Conv. 128 *ReLU* 3 × 3 Conv. 128 *ReLU* 8 × 8 max-pooling with 2 strides	3 × 3 Conv. 128 *ReLU* 3 × 3 Conv. 128 *ReLU* 3 × 3 Conv. 128 *ReLU* 3 × 3 Conv. 128 *ReLU* 3 × 3 Conv. 128 *ReLU* 4 × 4 max-pooling with 2 strides+position embedding
Average global pooling		Transformer encoder

Models	NIN		FitNet-4		ViT-L/16	
	Probabilistic	Evidential	Probabilistic	Evidential	Probabilistic	Evidential
Utility	0.8959	0.8978	0.9353	0.9361	0.9921	0.9908
p-value (McNemar's test)	0.0489		0.0492		0.0452	

Results of imprecise classification



The tolerance to imprecision $\gamma \in [0.5, 1.0]$ models the user's tolerance degree to imprecision:

- $\gamma = 0.5$ for precise classification
- $\gamma = 1$ for completely imprecise classification (all samples assigned to set Ω)
- Higher γ corresponds to more imprecise decisions

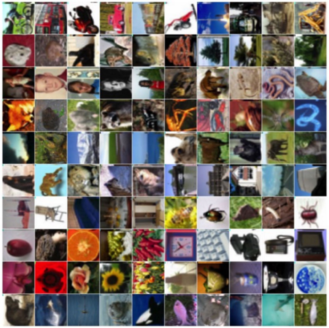
Examples of precise and imprecise classification

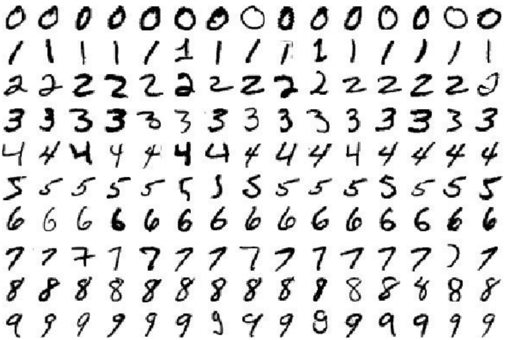
	#1(ω^* =cat)	#2(ω^*=dog)	#3(ω^* =deer)
γ=0.5	{dog}/0	{dog}/1	{deer}/1
γ=0.6	{cat,dog}/0.6	{cat,dog}/0.6	{deer}/1
γ=0.7	{cat,dog}/0.7	{cat,dog}/0.7	{deer,horse}/0.7
γ=0.8	{cat,dog}/0.8	{cat,dog}/0.8	{deer,horse}/0.8
γ=0.9	{cat,dog}/0.9	{cat,dog}/0.9	{cat,deer,dog,horse}/0.71
γ=1.0	Ω/1.0	Ω/1.0	Ω/1.0
			

Datasets for novelty detection

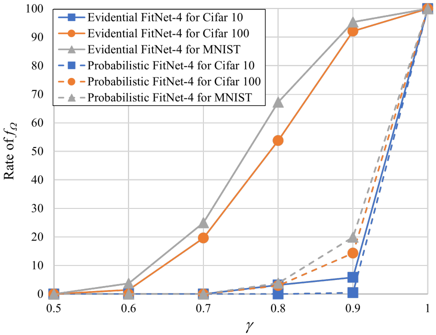
CIFAR-100 and MNIST for novelty-detection performance:

- 100 classes containing 600 images each in CIFAR-100
- 10 classes of handwritten digits containing 600 images each in MNIST





Results of Novelty detection (FitNet-4 backbone)



- The classifiers were trained using the CIFAR-10 dataset; the outliers are from the CIFAR-100 and MNIST datasets.
- A sample is rejected as outlier if it is assigned to set Ω.
- A good classifier has a high rate of assignment to Ω in an outlier set and a low rate of assignment to Ω in an inlier set.

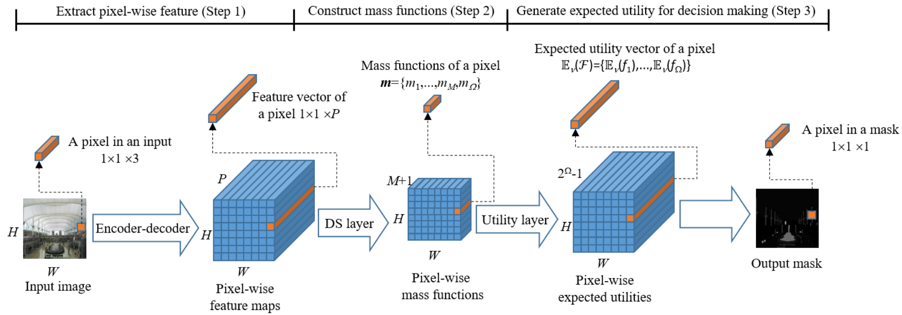
Conclusions about object classification

- Similar phenomena are also observed in the classification problems of signal processing and semantic relationship.
- Conclusions: our approach
  - Improves the CNN performance by assigning ambiguous patterns with uncertain information to multi-class sets.
  - Rejects outliers together with ambiguous patterns.
  - Outperforms the probabilistic CNN classifiers on imprecise classification and novelty detection.
  - Has similar or even better performance than the probabilistic classifiers on precise classification.

Outline

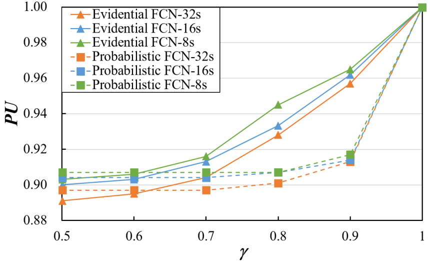
1. Background
   - Dempster-Shafer theory
   - Deep neural network
   - Evidential neural network
2. Evidential deep neural networks
   - Object classification
   - Semantic segmentation
3. Evidential multi-model fusion

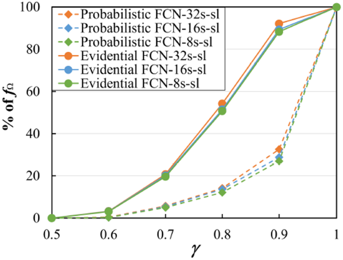
Network architecture



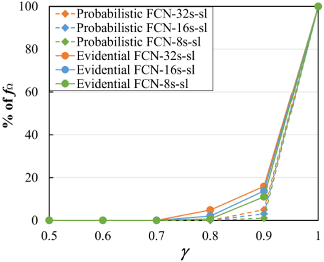
Segmentation results (Pascal VOC)

	Pixel utility (PU)	Utility of IoU
P-FCN-32s	0.8912 ± 0.0019	0.5941 ± 0.0033
P-FCN-16s	0.9001 ± 0.0015	0.6243 ± 0.0025
P-FCN-8s	0.9033 ± 0.0017	0.6269 ± 0.0021
E-FCN-32s	**0.8973** ± 0.0021	**0.6128** ± 0.0024
E-FCN-16s	**0.9045** ± 0.0014	**0.6304** ± 0.0019
E-FCN-8s	**0.9074** ± 0.0015	**0.6337** ± 0.0020



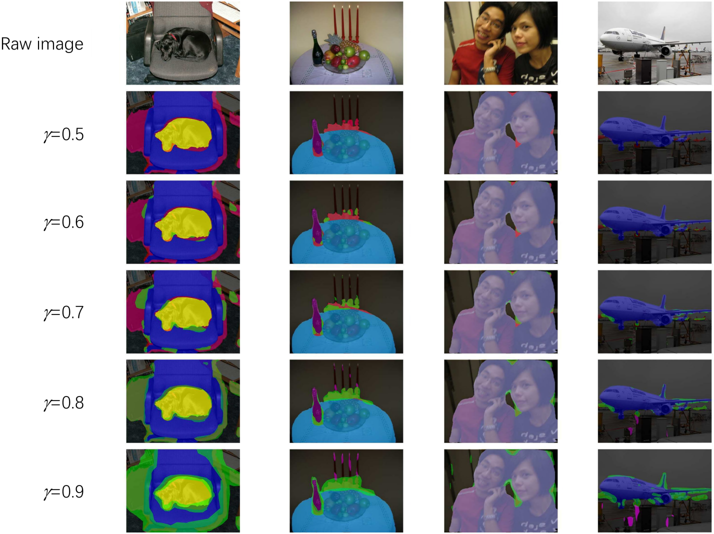


(a) Outlier set



(b) Inlier set

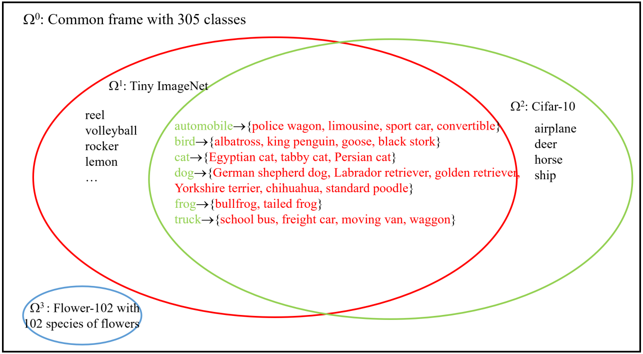
Segmentation examples on Pascal VOC



Outline

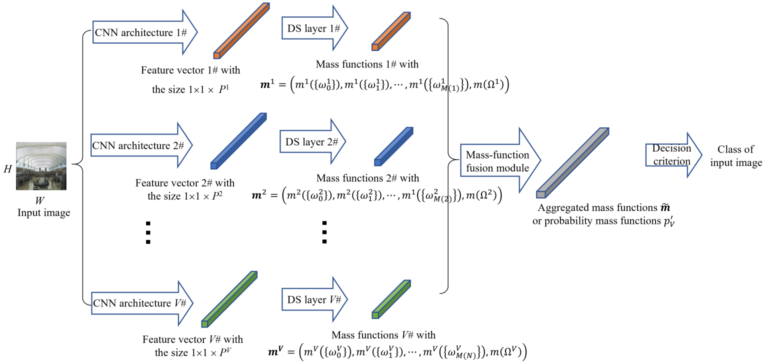
Problem definition

- Many DNNs have been trained using different datasets. How to use these existing networks?
- This is a hard problem, because classifiers trained on different learning sets have different frames of discernment.



- Here, we focus on the fusion of the DNNs with different sets of classes.

Evidential fusion approach (classification problem)



- A mass-function fusion module refines V different frames into a common one Ω^0 and computes the vacuous extensions of different masses in the common frame.
- The contour functions of these vacuous extensions are aggregated by Dempster's rule, as p_V.

Compatible frames with an "anything else" elements

- Not all frames of discernment are compatible.
- We add an "anything else" elements ω_0^v in the v-th frames, $v = 1, \ldots, V$.

Frame	Class
CIFAR-10 Ω^1	airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck, ω_0^1.
Tiny ImageNet Ω^2	reel, volleyball, rocker, police wagon, limousine, ..., (200 classes), ω_0^2.
Flower-102 Ω^3	bengal, boxer, ..., (102 species of flowers), ω_0^3.
Common frame Ω^0	airplane, deer, horse, ship, reel, volleyball, rocker, police wagon, limousine, ..., (200 classes from Tiny ImageNet), buttercup, alpine sea holly, ..., (102 species of flowers).

- Each DS layer has an extra output $m^v(\{\omega_0^v\})$.

Learning with soft labels

- Learned network weights may not be very suitable for the new task:
  - An extra output $m^v(\{\omega_0^v\})$ in each DS layer
  - Soft labels in the dataset
    Cifar-10+Tiny ImageNet with cat class $\rightarrow$ {Egyptian cat, tabby cat, Persian cat}
- Fine-tuning processes:
  - We merge the learning sets of different DNNs into a single one
  - Given a learning sample with a nonempty label $A_* \subseteq \Omega^0$, the aggregated contour function p_V is normalized as

$$p'_V(\omega_i) = \frac{p_V(\omega_i)}{\sum_{j=1}^{M^0} p_V(\omega_j)}, \quad i = 1, \ldots, M^0$$

  - The prediction loss is

$$\mathcal{L}(p'_V, A_*) = -\log \sum_{\omega \in A_*} p'_V(\omega)$$

Comparison study

- Probability-to-mass fusion (PMF): probabilistic networks (softmax output), combination of probabilities (after extension to Ω^0) by Dempster's rule.
- Bayesian-fusion (BF): probability networks (softmax output), probabilities computed on Ω^0 as uniform distributions, combination by Dempster's rule.
- Probabilistic feature-combination (PFC): concatenation of feature vectors extracted by the three networks + softmax layer.
- Evidential feature-combination (EFC): concatenation of the feature vectors extracted by the three networks + DS layer.

Results (ResNet-101 backbones)

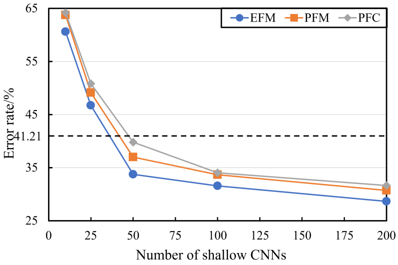
	Classifier	Tiny ImageNet	Flower-102	CIFAR-10	Overall
Before fusion	Evidential CNN	18.66	4.68	4.61	-
	Probabilistic CNN	18.70	4.69	4.66	-
After fusion without E2E learning	Proposed method	18.52	4.68	3.94	10.31
	Probability-to-mass fusion	18.54	4.69	4.42	10.40
	Bayesian-fusion	19.18	5.07	6.04	11.10
After fusion with E2E learning	**Proposed method**	18.50	**4.67**	**3.82**	**10.27**
	Probability-to-mass fusion	**18.49**	4.68	4.28	10.35
	Bayesian-fusion	18.87	4.99	5.74	10.89
	Probabilistic feature-combination	18.59	5.74	4.89	10.94
	Evidential feature-combination	21.68	5.46	7.57	12.56

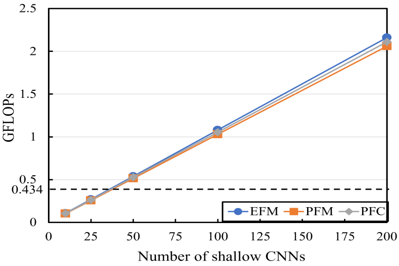
Examples

Instance/label	Before fusion			p' on Ω^0 after fusion
	p' from Tiny ImageNet	p' from CIFAR-10	p' from Flower102	
	p'(Egyptian cat) $= 0.47$	p'(cat) $= 0.87$	p'(buttercup) $= 0.001$	p'(Egytian cat) $= 0.86$
	p'(chihuahua) $= 0.51$	p'(dog) $= 0.12$	p'(camellia) $= 0$	p'(chihuahua) $= 0.13$
	...	...	...	...
Egyptian cat	$p'(\omega_0^1) = 0.001$	$p'(\omega_0^2) = 0.001$	$p'(\omega_0^3) = 0.99$	$p'(\omega_0^0) = 0.001$
	p'(king penguin) $= 0.45$	p'(bird) $= 0.73$	p'(buttercup) $= 0$	p'(king penguin) $= 0.98$
	p'(academic gown) $= 0.53$	$p'(\{\text{frog}\}) = 0.10$	p'(camellia) $= 0.001$	p'(academic gown) $= 0.01$
	...	...	...	...
king pengui	$p'(\omega_0^1) = 0.001$	$p'(\omega_0^2) = 0.004$	$p'(\omega_0^3) = 0.99$	$p'(\omega_0^0) = 0.001$
	p'(bull frog) $= 0.38$	p'(frog) $= 0.97$	p'(buttercup) $= 0.001$	p'(bull frog) $= 0.39$
	p'(tailed frog) $= 0.60$	p'(cat) $= 0.01$	p'(camellia) $= 0$	p'(tailed frog) $= 0.61$
	...	...	...	...
bull frog	$p'(\omega_0^1) = 0$	$p'(\omega_0^2) = 0$	$p'(\omega_0^3) = 0.99$	$p'(\omega_0^0) = 0$

Combining simple DNNs for a complex classification task

- Objective: solve a complex problem with some simple DNNs, instead of a very deep one.
- Approach:
  - Decompose a complex classification problem into simple ones
  - Solve each problem by a simple DNN
  - Combine these DNNs by the evidential fusion approach





Conclusions about multi-model fusion

- Similar results were found in other semantic-segmentation experiments.
- Conclusions: our approach
  - Combines DNNs trained from heterogeneous datasets.
  - Outperforms other decision-level or feature-level fusion strategies for combining DNNs.

General conclusions

- Evidential DNNs
  - Assign ambiguous samples to multi-set
  - Reject outliers together with ambiguous samples
  - Have similar or even better performance for precise problems of classification and segmentation.
- Evidential fusion of heterogeneous DNNs
  - Combines DNNs with different sets of classes
  - Outperforms other decision-level or feature-level fusion strategies for combining DNNs.

Perspectives

- Evidential DNNs
  - Combine with other up-to-date CNNs and FCNs to achieve better performance
  - Combine with other types of DNNs, such as recurrent neural networks for natural language processing
  - Compare other uncertainty quantification methods with the DS layer, such as probabilities with a Dirichlet distribution
  - More metrics to evaluate the performance of evidential DNNs, such as top k-categorical accuracy and learning curves
  - Can be trained by a small dataset?
- Evidential fusion
  - Compares with more information-fusion methods, such as error-correcting output codes
  - Obtains the semantic relationship automatically.

Publications

- International journals:
  - **Z. Tong**, Ph. Xu, T. Denœux. An evidential classifier based on Dempster-Shafer theory and deep learning. *Neurocomputing*, August 2021, 450, 275-293
  - **Z. Tong**, Ph. Xu, T. Denœux. Evidential fully convolutional network for semantic segmentation. *Applied Intelligence*, April 2021, 51, 6376-6399
- International conferences:
  - **Z. Tong**, Ph. Xu, T. Denœux. ConvNet and Dempster-Shafer Theory for Object Recognition. In: *International Conference on Scalable Uncertainty Management* (SUM 2019) , pp. 368-381. Springer, Cham, France, 2019
  - **Z. Tong**, Ph. Xu, T. Denœux. Fusion of evidential CNN classifiers for image classification. In: *International Conference on International Conference on Belief Functions* (BELIEF 2021) . Springer, Shanghai, China, 2021. (Best paper award)
- Available codes:
  - Evidential CNN `https://github.com/tongzheng1992/E-CNN-classifier`
  - Evidential FCN `https://github.com/tongzheng1992/E-FCN`

Thank you!