PCMO: Partial Classification from CNN-Based Model Outputs

Jiarui Xie[1 [0000−0001−6959−1089]](✉), Violaine Antoine[1 [0000−0002−0981−3505]], and Thierry Chateau[2,3 [0000−0003−4854−5686]]

[1] Université Clermont-Auvergne, CNRS, Mines de Saint-Étienne, Clermont-Auvergne-INP, LIMOS, 63000 Clermont-Ferrand, France
{jiarui.xie,violaine.antoine}@uca.fr
[2] Logiroad, Chevroliere, France
[3] Institut Pascal, UMR6602, CNRS, University of Clermont Auvergne, Clermont-Ferrand, France
thierry.chateau@uca.fr

Abstract. The partial classification can assign a sample to a class subset when this sample has similar probabilities for multiple classes. However, the extra information for making such predictions usually comes at the cost of retraining the model, changing the model architecture, or applying a new loss function. In an attempt to alleviate this computational burden, we fulfilled partial classification only based on pre-trained CNN-based model outputs (PCMO), by transforming the model outputs to beliefs for predicted sets under the Dempster-Shafer theory. The PCMO method has been executed on six prevalent datasets, four classical CNN-based models, and compared with three existing methods. For instance, experiments with MNIST and CIFAR10 datasets show the superiority of PCMO in terms of average discounted accuracy (0.23% and 7.71% improvement, respectively) when compared to other methods. The performance demonstrated that the PCMO method makes it possible to improve classification accuracy and to make cautious decisions by assigning a sample to a class subset. Moreover, the PCMO method is simple to implement compared to the existing methods, as the PCMO method does not need to retrain the model or conduct any further modifications.

Keywords: CNN-based model · Decision making· Dempster-Shafer theory · Partial classification.

1 Introduction

The precise or certainty classification [18, 36] is a well-known issue in which a sample is classified into one and only one of the training classes. Unfortunately, such a strict classification sometimes results in misclassification when the input sample does not contain sufficient evidence to identify a certain class. The partial classification [9, 22, 25] is one of the more practical ways to solve this problem. It is defined as the assignment of a sample into a class subset. For example, let us consider a class set $\Omega = \{\omega_1, \omega_2, \omega_3\}$. Here, we cannot manage to reliably

classify a sample into a single class, but it is almost sure that it does not belong to ω_1. Consequently, it is more reasonable to assign it to the subset $\{\omega_2, \omega_3\}$. In practice, high ambiguity emerges in numerous applications, and large-scale datasets contain a fair amount of confusing samples, these are the bedrock of the usage of partial classification. For instance, the goal of road surfaces classification [40] is to produce a prediction with almost null error which can be expected from partial classification.

A considerable amount of literature has been published on partial classification and has always led to different classification strategies. On the one hand, researchers attempted to predict a subset with prior fixed cardinality [30] or with a rejection option [13, 15, 19]. They can be seen as a special case of partial classification by classifying the sample into one specific class subset. On the other hand, a number of authors attempted to modify the loss function [3, 10, 26] or build a new classifier [31, 35, 38] to provide beliefs for predicted sets. Usually, such algorithms are time-consuming. To this end, it is essential to reduce the computation and time complexity by efficiently and sufficiently leveraging the information provided by the pre-trained neural network.

In this paper, we proposed a new partial classification method based on pre-trained CNN-based model outputs (PCMO). Different from the existing methods, the PCMO method simply and efficiently fulfilled partial classification only based on pre-trained CNN-based model outputs, and provided beliefs to predicted sets for further prediction. As manifested in Fig. 1, at first, the CNN-based model extracts features from the input layer through the combination of the feature extraction process and the fully connected layer between the last hidden layer and the output layer. Second, the received features are converted into beliefs under the Dempster-Shafer theory (DST) [32] through the output to possibility and the possibility to belief processes. Finally, the PCMO method performs partial classification based on the produced beliefs by choosing the maximum belief and generating the corresponding class subset as the prediction.

The contributions of our work can be summarized as follows:

- The most striking achievement is that the proposed method is fulfilled only based on model outputs that can be applied to any pre-trained CNN-based model without any demand to retrain the model or conduct any further modifications.
- By considering good features of log function and analyzing the regular pattern of model outputs, a novel and reasonable transformation from model outputs to possibility distribution is proposed.

2 Related Work

Partial classification Partial classification also known as set-valued classification becomes prevalent recently imputable to its capability dealing with ambiguity. At the first glance partial classification seems to be linked to multi-label classification [4, 34]. The confusion comes from the fact that both methods produce a class subset as the prediction. However, the crucial dissimilarity comes in that

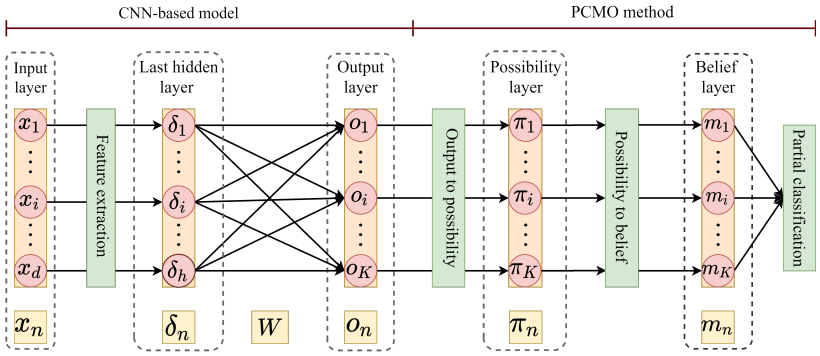


Fig. 1. Framework of the proposed method. The feature extraction process is demonstrated simply, it can be any kind of CNN-based architecture, e.g., fully connected layer, LeNet [16], GoogLeNet [33], or ResNet [11]. The detailed output to possibility, possibility to belief, and partial classification processes are presented in Section 4.

an input sample is labeled by a subset of classes for multi-label classification, whereas for partial classification, only a single class.

The straightforward way to fulfill partial classification is to always predict a fixed number of classes such as the top five most proper classes [30]. However, there is no reason to predict exactly five or any other a prior fixed number of classes all the time. Classification with rejection option [13, 15, 19] is another plain strategy that concerns the treatment of outliers that are not defined by any of the training classes. Depending on this strategy, such samples are assigned to the empty set, or the entire set, reflecting maximum uncertainty. Both the top five and rejection strategies can be seen as a special case of partial classification by giving belief for one specific class subset. Apart from the above methods, there are two directions that aim at modifying loss function or building the new classifier to provide beliefs for predicted sets. On the one hand, Ha [10] introduces a loss function consist of the sum of two terms, one reflecting the loss of missing the ground-truth labels, and the other penalizing imprecision. A similar loss function used in [26] is composed of the uncertainty quantified by conditional class probabilities, and the quality of the predicted set measured by a utility function. Besides, Coz et al. [3] propose a loss function inspired by aggregating precision and recall. On the other hand, Vovk et al. [35] proposed an approach to learn a partial classifier with finite sample confidence guarantees. In the same context, Sadinle et al. [31] designed a classifier that guarantees user-defined levels of coverage while minimizing ambiguity. As we can see, the weakness of the above methods is the time-consuming nature, e.g., retraining model, changing model architecture, and applying the new loss function.

Dempster-Shafer theory Dempster-Shafer theory (DST) [32] is a general framework for reasoning with uncertainty which is proposed by Arthur P. Dempster [5] then refined by Glenn Shafer [32] also know as evidence theory. The existing works related to CNN have the following three main directions. The

first one is classifier fusion, in which the outputs of several classifiers are transformed into belief functions and aggregated by suitable combination rules [1, 20, 42]. Another direction is evidential calibration, the decisions of classifiers are converted into belief functions with some frequency calibration property [21, 23, 24]. The last approach is to design evidential classifiers [6, 7], which transformed the evidence of the input sample into beliefs and combine them by appropriate combination rules.

3 Background

3.1 The pattern of the CNN-based model outputs

The convolutional neural network (CNN) [16] is a machine learning method that uses multiple layers to progressively extract features from raw data as sample representation. Define a traning dataset $\mathcal{D}^{train} = \{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$ has K classes, where $\boldsymbol{x}_n \in \mathbb{R}^d$, and $y_n \in \{1, \ldots, i, \ldots, K\}$. A CNN-based model $f(\boldsymbol{x}; \boldsymbol{\theta})$, with the entire model parameter $\boldsymbol{\theta}$. From the last hidden layer $\boldsymbol{\delta}_n = \{\delta_1, \ldots, \delta_i, \ldots, \delta_h\}$ to the output layer $\boldsymbol{o}_n = \{o_1, \ldots, o_i, \ldots, o_K\}$, the weight $\boldsymbol{W} \in \mathbb{R}^{h \times K}$ defines a transformation, i.e., $\boldsymbol{o}_n = \boldsymbol{W}\boldsymbol{\delta}_n$. In general, the empirical loss $\mathcal{L}(\boldsymbol{\theta})$ over $\mathcal{D}^{train}$ has the following form:

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \ell(f(\boldsymbol{x}_n; \boldsymbol{\theta}), y_n) \tag{1}$$

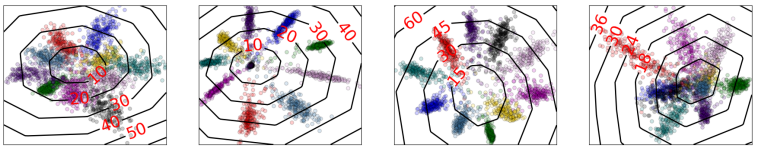
where $\ell(\cdot)$ is the specified loss function, e.g., logistic loss, exponential loss, or cross-entropy loss.

The CNN-based model mentioned in this article respects two usual and reasonable assumptions. The model loss Eq. (1) converges to zero when iteration t approaches infinity, i.e., $\lim_{t \to \infty} \mathcal{L}(\boldsymbol{\theta}_t) = 0$, and the model's last hidden layer and the output layer are fully connected. Based on the two assumptions, [41] demonstrates, both theoretically and empirically, that the last weight layer $\boldsymbol{W}$ of a neural network converges to a support vector machine (SVM) trained on the last hidden layer output $\boldsymbol{\delta}$ with the commonly used cross-entropy loss.

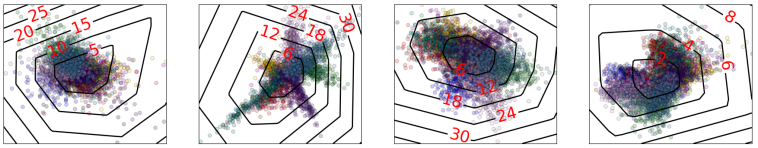
Since $\boldsymbol{W}$ represents a hyperplane, the farther the input sample is from the hyperplane, the greater the corresponding class output, i.e., $max(\boldsymbol{o}_n)$ will be. As illustrated in Fig. 2, the model output contours of the CNN-based model are radiated, becoming higher as the distance from the hyperplane increases.

3.2 Dempster–Shafer theory

The Dempster–Shafer theory [32] (or evidence theory) is a mathematical framework that enables the reflection of partial and uncertain knowledge. Let $\Omega = \{\omega_1, \ldots, \omega_i, \ldots, \omega_K\}$ be the finite class sets. The belief function $m : 2^{\Omega} \to [0, 1]$ applied on $\boldsymbol{x}_n$ measures the degree of belief that the ground-truth label of $\boldsymbol{x}_n$



(a) MNIST dataset



(b) CIFAR10 dataset

Fig. 2. The model output contours on MNIST [17] and CIFAR10 [14] datasets. Different prevalent CNN-based models are verified. From left to right are LeNet [16], GoogLeNet [33], ResNet [11], and MobileNet [12]. For visualization purposes, the h that appears in the last hidden layer is set as two. Meanwhile, according to the minimum and maximum column values of $\boldsymbol{\delta} \in \mathbb{R}^{N\times 2}$ a 2D mesh can be generated. Feed this mesh into the last hidden layer to get the outputs which can be regarded as contours. As we can see, the pattern of the CNN-based model is that a sample far from the training dataset can bring high outputs and lead to high probabilities for several classes. Under this context, partial classification rather than precise classification should be used.

belongs to a subset $A_i \subseteq \Omega$. It satisfies the following equation:

$$\sum_{A_i \subseteq \Omega} m(A_i) = 1 \tag{2}$$

The subset A_i such that $m(A_i) > 0$ is called the focal set of m. When the focal set is nested, m is said to be *consonant*. As we can see the maximum quantity of beliefs is 2^K which is a significant difference from K for probability. Since the maximum quantity of subsets of classes is also 2^K, belief instead of probability is inherently more suitable for partial classification.

4 Proposed Method

As can be seen from Section 3.1, a sample that is far from the training dataset occupies high outputs for several classes leading to high probabilities for the corresponding classes, resulting in the improper execution of precise classification. Consider, from another angle, the high outputs for multiple classes can be regarded as evidence to classify a sample into a class subset. From this point, we proposed to calculate beliefs only based on pre-trained CNN-based model outputs to fulfill partial classification. Moreover, we chose the possibility as the bridge between model outputs and beliefs, then proposed the following transformations.

Sorting $\boldsymbol{o}_n$ by descending order to get $\boldsymbol{o}'_n = \{o'_1 \geq \cdots \geq o'_i \geq \cdots \geq o'_K\}$, where o'_i is the i^{th} largest element in $\boldsymbol{o}_n$. Then, a prerequisite step is to prepare a temporary vector $\boldsymbol{v}_n = \{v_1, \ldots, v_i, \ldots, v_K\}$ based on Eq. (3) that coordinates with Eq. (4) to calculate the target possibility distribution.

$$v_i = \frac{1}{|A_i|} \sum_{k=1}^{i} \log_2(1 + \max(0, o'_k)) \tag{3}$$

where $\frac{1}{|A_i|}$ is used to penalize the ambiguity caused by classifying $\boldsymbol{x}_n$ to A_i. If we consider a reasonable assumption that the desired possibility transformation should keep the original pattern of outputs, escalating the difference for small values while narrowing the difference for bigger values. The $\log_2$ function should be chosen, which tends to be flat after the initial rapid growth. At the same time, in order to avoid the negative possibility, use $\max(0, o'_k)$ to clamp the outputs and move the $\log_2$ to the left by one unit.

After min-max normalization by Eq. (4), we can get the possibility distribution $\boldsymbol{\pi}_n = \{\pi_1, \ldots, \pi_i, \ldots, \pi_K\}$. Following the theory proposed in [2] that any possibility distribution is a plausibility function corresponding to a consonant m. Our possibility distribution $\boldsymbol{\pi}_n$ can be transformed to belief function m according to Eq. (5), the detailed calculation is presented in Fig. 3. In our case, π_K equals zero, which implies that $m(\Omega)$ equals zeros.

$$\boldsymbol{\pi}_n = \frac{\boldsymbol{v}_n - \min(\boldsymbol{v}_n)}{\max(\boldsymbol{v}_n) - \min(\boldsymbol{v}_n)} \tag{4}$$

$$m(A_i) = \begin{cases} \pi_j - \pi_{j+1} & \text{if } A_i = \{\omega_1, \cdots, \omega_j\} \text{ for some } j \in \{1, \cdots, K-1\} \\ \pi_K & \text{if } A_i = \Omega \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

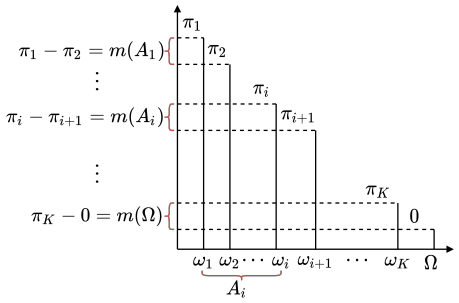


Fig. 3. Calculation of the belief function m [2].

The PCMO classification algorithm is demonstrated in Algorithm 1. Based on the beliefs calculated through Eqs. (3), (4), and (5), we chose the subset with

the maximum belief as the prediction. Suppose, the maximum belief is $m(A_i)$, the PCMO method will generate the predicted set $\{\omega_1, \ldots, \omega_i\}$ corresponding to the top i maximum outputs. In addition, the usage is flexible, for example, $\overline{m(A_1)} = 1 - m(A_1)$ can be regarded as the uncertainty for the classification as used in Section 5.4.

Algorithm 1: Classification process for a sample $\boldsymbol{x}_n$

Data: Model outputs $\boldsymbol{o}_n \in \mathbb{R}^{1\times K}$
Result: Predicted set *predSet*
Sort $\boldsymbol{o}_n$ in descending order to get the sorted index *index* and sorted outputs $\boldsymbol{o}'_n$;
Calculate vector $\boldsymbol{v}_n$ base on $\boldsymbol{o}'_n$ according to Eq. (3);
Calculate possibility distribution $\boldsymbol{\pi}_n$ base on $\boldsymbol{v}_n$ according to Eq. (4);
Calculate belief $\boldsymbol{m}_n$ base on $\boldsymbol{\pi}_n$ according to Eq. (5);
Obtain the maximum belief index $idx = argmax(\boldsymbol{m}_n)$ for the sample $\boldsymbol{x}_n$;
Generate the predicted set $predSet = list(index[0 : i + 1])$, which contains the candidate classes;
return *predSet*;

5 Experiments

5.1 Experiment protocol

There are six datasets involved, a Road Surface dataset manually generated and five prevalent datasets, i.e., modified national institute of standards and technology (MNIST) [17], canadian institute for advanced research 10 (CIFAR10) [14], street view house number (SVHN) [29], large-scale scene understanding challenge (LSUN) [37], and canadian institute for advanced research 100 (CIFAR100) [14]. The characteristics of the datasets are shown in Table 1. Four classical CNN-based models, i.e., LeNet [16], GoogLeNet [33], residential energy services network (ResNet) [11], and MobileNet [12], are adopted to prove the efficiency of the PCMO method. We used the cross-entropy loss as the loss function and the rectified linear unit (ReLU) as the activation function. For method comparison, we chose energy score (based on pre-trained model outputs) [19], dropout score (based on several executions of the pre-trained model on the testing dataset) [13], and ensemble score (based on executions of several pre-trained models on the testing dataset) [15].

5.2 Criteria

Traditional accuracy becomes improper when partial classification is allowed. In this case, Zaffalon [39] proposed the following discounted accuracy:

$$a = \frac{1}{|A_i|} I(y_n \in A_i) \tag{6}$$

Table 1. A quick view of datasets involved.

Name	# Classes	# Training samples	# Testing samples
MNIST [17]	10	55000	10000
CIFAR10 [14]	10	50000	10000
SVHN [29]	10	4000	1000
LSUN [37]	10	2400	600
CIFAR100 [14]	10	8000	2000
Road Surface	3	2040	780

where $I(\cdot)$ is the indicator function.

For a dataset, the accuracy is evaluated by the average discounted accuracy (ADA). The ADA is a single value with the requirement that the better the prediction, the larger the ADA is.

$$\mathrm{ADA} = \frac{1}{N}\sum_{n=1}^{N} a_n \tag{7}$$

In addition, to approximately measure the goodness of calibration, expected calibration error (ECE) [28] defined by Eq. (8) was adopted. This groups the probability interval into B bins with n_b samples inside and assigns each predicted probability to the bin that encompasses it. The calibration error is the difference between the fraction of predictions in the bin that are correct (accuracy) and the mean of the probabilities in the bin (confidence).

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{n_b}{N} |\,\mathrm{acc}(b) - \mathrm{conf}(b)| \tag{8}$$

where $\mathrm{acc}(b)$ and $\mathrm{conf}(b)$ are the accuracy and confidence of bin b, respectively.

5.3 Evaluation of the PCMO method

The PCMO method performs partial classification by choosing the predicted set that occupies the maximum belief. Naturally, the bigger cardinality of the predicted set indicates a more confusing input sample. Thus, in order to verify the efficiency of partial classification and the capacity of reducing the classification risk under the PCMO method. We rejected the most confusing samples according to different rejection rates [27].

On the one hand, we executed the PCMO method for different CNN-based models when rejection rates change from 0.0 to 1.0. Fig. 4 is quite revealing in two ways. First, the ADA increases along with the increase of rejection rates. Second, the selected four classical CNN-based models achieved good ADA values, except for the slightly worse initial accuracy of LeNet and MobileNet due to its simple model architecture. This indicates that the PCMO method performed partial classification based on the calculated beliefs. The performance on the different

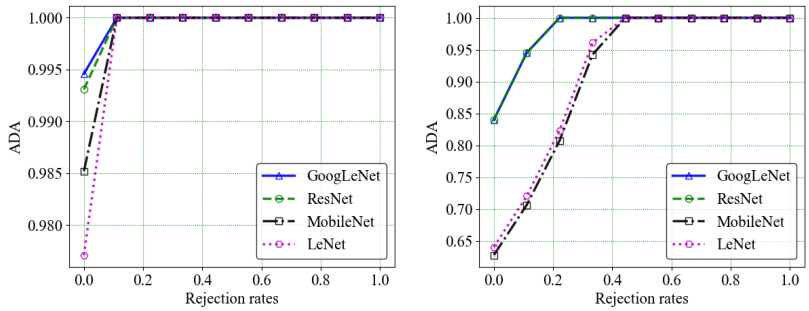


Fig. 4. The performance in terms of ADA values of different CNN-based models with different rejection rates based on MNIST (left) and CIFAR10 (right) datasets.

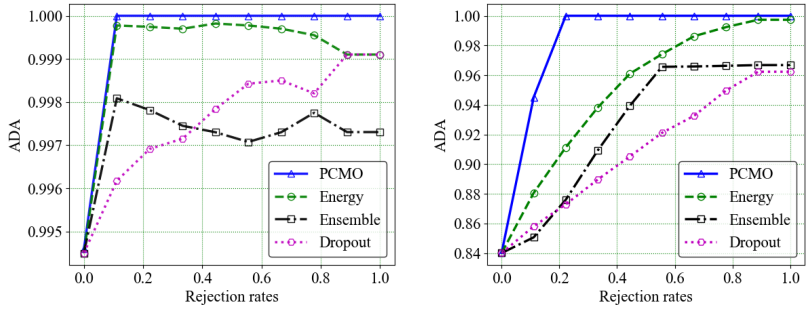


Fig. 5. The performance in terms of ADA values of different methods with different rejection rates based on MNIST (left) and CIFAR10 (right) datasets.

CNN-based models further proves that partial classification can be achieved only based on the CNN-based model outputs.

On the other hand, we verified different methods based on GoogLeNet (the best performing model in the previous step), as demonstrated in Fig. 5. The ADA of the PCMO method increases significantly when the rejection rate increases from 0.0 to 0.1 or 0.2. In contrast, the ADA of the other methods performed fluctuation or insensitivity when the rejection rate increased from 0.0 to 1.0. The striking performance is evidence that the PCMO method makes a well-distributed partial classification while the others only classified samples to a class subset when the rejection rate is large.

To manifest the efficiency of different methods and models when against a small rejection rate, we set rejection rate equals 0.1 and received Table 2. The PCMO method achieves the highest ADA and comparable ECE values among all the methods. Compared to other methods, there is a 0.23% and 7.71% improvement in terms of ADA for MNIST and CIFAR10, respectively, based on GoogelNet. The detailed statistics are demonstrated in Table 3. It is also

Table 2. Comparative experimental results on five datasets for four CNN-based models and four methods when the rejection rate equals 0.1.

Datasets		MNIST		CIFAR10		SVHN		LSUN		CIFAR100	
Criteria		ADA	ECE	ADA	ECE	ADA	ECE	ADA	ECE	ADA	ECE
PCMO based on models	GoogLeNet	1.000	0.004	0.933	0.046	1.000	0.014	0.811	0.096	0.961	0.091
	ResNet	1.000	0.003	0.934	0.040	1.000	0.018	0.924	0.078	0.994	0.082
	MobileNet	1.000	0.006	0.712	0.019	0.790	0.071	0.333	0.077	0.406	0.178
	LeNet	1.000	0.009	0.698	0.031	0.209	0.015	0.135	0.041	0.144	0.037
Methods based on GoogLeNet	PCMO	**1.000**	**0.004**	**0.933**	**0.046**	**1.000**	**0.014**	**0.811**	**0.096**	**0.961**	**0.091**
	Energy	1.000	0.001	0.877	0.041	0.987	0.008	0.765	0.102	0.912	0.072
	Ensemble	0.998	0.001	0.847	0.039	0.967	0.016	0.732	0.098	0.856	0.085
	Dropout	0.995	0.003	0.844	0.043	0.964	0.014	0.736	0.102	0.873	0.079
Methods based on ResNet	PCMO	**1.000**	**0.003**	**0.934**	**0.040**	**1.000**	**0.018**	**0.924**	**0.078**	**0.994**	**0.082**
	Energy	1.000	0.002	0.878	0.033	0.991	0.008	0.898	0.054	0.939	0.053
	Ensemble	0.998	0.001	0.846	0.034	0.967	0.015	0.824	0.075	0.884	0.071
	Dropout	0.993	0.003	0.840	0.037	0.956	0.019	0.811	0.080	0.878	0.081
Methods based on MobileNet	PCMO	**1.000**	**0.006**	**0.712**	**0.019**	**0.790**	**0.071**	**0.333**	**0.077**	**0.406**	**0.178**
	Energy	0.989	0.004	0.668	0.018	0.743	0.073	0.312	0.077	0.370	0.181
	Ensemble	0.998	0.016	0.671	0.028	0.752	0.079	0.325	0.080	0.365	0.166
	Dropout	0.976	0.006	0.639	0.016	0.708	0.061	0.298	0.075	0.354	0.174
Methods based on LeNet	PCMO	**1.000**	**0.009**	**0.698**	**0.031**	**0.209**	**0.015**	**0.135**	**0.041**	**0.144**	**0.037**
	Energy	0.999	0.003	0.662	0.030	0.188	0.018	0.144	0.039	0.155	0.037
	Ensemble	0.998	0.003	0.667	0.016	0.206	0.013	0.144	0.039	0.166	0.050
	Dropout	0.984	0.009	0.627	0.032	0.192	0.011	0.137	0.032	0.149	0.033

Table 3. The detailed performance improvement statistics. This represents the subtraction between the ADA value produced by the PCMO method and the averaged ADA value of the other three methods.

Models	Datasets				
	MNIST	CIFAR10	SVHN	LSUN	CIFAR100
GoogLeNet	0.23%	7.71%	2.74%	2.74%	2.74%
ResNet	0.31%	7.90%	2.89%	2.89%	2.89%
MobileNet	1.20%	5.24%	5.56%	5.56%	5.56%
LeNet	0.63%	4.60%	1.36%	1.36%	1.36%

clear, for each CNN-based model, that no matter what dataset is involved, the PCMO method performs better than other methods. This proved that the PCMO method is effective in reducing the misclassification of the confusing sample with the beliefs calculated only based on pre-trained CNN-based model outputs.

5.4 Practical usage of the PCMO method

The PCMO method has a wide range of application contexts, e.g, autonomous driving [40]. To prove the PCMO method is effective in reducing classification risk practically, we applied it to the road surface classification, which is an essential part of autonomous driving. The Road Surface dataset is mixed manually from

the Crack [40] and Pothole [8] datasets, which contains three classes, i.e, crack, pothole, and normal. Several samples are shown in Fig. 6

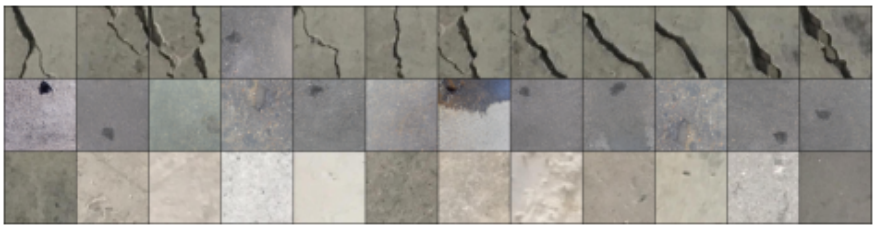


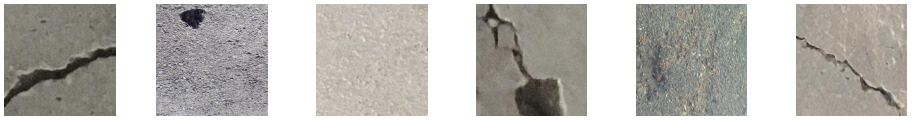
Fig. 6. The visualization of the Road Surface dataset, which contains cracks (the first row), potholes (the second row), and normals (the third row). As we can see, each class contains several confusing samples, e.g, the fourth one, the third one, and the second one for each class, respectively. The PCMO method aims to detect confusing samples, reducing the classification risk.

Fig. 7(a) shows a certain crack sample, in which the crack area is obvious and clean. Thus, the PCMO method produced high belief $m(crack) = 0.85$ and low uncertainty $\overline{m(crack)} = 0.15$ to indicate the certain classification. Similarly, Fig. 7(b) and Fig. 7(c) show a certain pothole and normal sample, respectively, which contains sufficient evidence (obvious class characteristics) to support a certain prediction.

Inversely, Fig. 7(d) demonstrated a confusing crack sample that is difficult to identify from a pothole sample due to its circle-shaped crack. Fig. 7(e) manifested a confusing pothole sample whose pothole is too shallow to identify from a normal sample. And Fig. 7(f) demonstrated a confusing normal sample whose crack indicates it is a crack sample rather than a normal sample. Correspondingly, the PCMO method produces low belief and high uncertainty as illustrated in Fig. 7. Based on the PCMO method, we can reduce classification risk and enhance the quality of the target system by rejecting confusing samples.

6 Conclusion

In this paper, we proposed a new partial classification method named PCMO, which is fulfilled based on pre-trained CNN-based model outputs. At first, we theoretically and empirically proved our hypothesis that a sample far from the training dataset can provide high outputs and lead to high probabilities for several classes. Second, we adopted possibility as the bridge fulfilling the transformation from model outputs to beliefs for the predicted sets. Then, we verified the PCMO method with different CNN-based models, as well as different methods based on five datasets. Finally, to demonstrate the practical usage of the PCMO method,



(a) $m(crack) = 0.85$, $\overline{m(crack)} = 0.15$

(b) $m(pothole) = 0.80$, $\overline{m(pothole)} = 0.20$

(c) $m(normal) = 0.86$, $\overline{m(normal)} = 0.14$

(d) $m(crack) = 0.30$, $\overline{m(crack)} = 0.70$

(e) $m(pothole) = 0.20$, $\overline{m(pothole)} = 0.80$

(f) $m(normal) = 0.11$, $\overline{m(normal)} = 0.89$

Fig. 7. The beliefs calculated based on the proposed method for different types of samples. (a) Certain crack sample, (b) certain pothole sample, (c) certain normal sample, (d) confusing crack sample, (e) confusing pothole sample, and (f) confusing normal sample.

we conducted experiments based on a manually generated road surface dataset. From the production of ADA and ECE criteria, we can tell that the PCMO method performs better than the existing methods, as it can provide a high belief to a certain sample, as well as a high uncertainty to a confusing sample. The PCMO method proved effective in increasing prediction accuracy and ultimately reducing the classification risk. In the future, we plan to explore the feasibility of PCMO methods on all types of pre-trained models and try to develop a method to generate beliefs for all subsets of the entire set.

Acknowledgments

This work was funded by the Auvergne Rhône Alpes region: project AUDACE2018. The authors would like to thank the reviewers for their very insightful comments that helped to improve the article.

References

1. Bi, Y.: The impact of diversity on the accuracy of evidential classifier ensembles. International Journal of Approximate Reasoning **53**(4), 584–607 (2012)
2. D. Dubois, H.P.: On several representations of an uncertainty body of evidence. Fuzzy Information and Decision Processes p. 167–181 (1982)
3. Del Coz, J.J., Diez, J., Bahamonde, A.: Learning nondeterministic classifiers. Journal of Machine Learning Research **10**(10) (2009)
4. Dembczyński, K., Waegeman, W., Cheng, W., Hüllermeier, E.: On label dependence and loss minimization in multi-label classification. Machine Learning **88**(1-2), 5–45 (2012)
5. Dempster, A.P.: Upper and lower probabilities induced by a multivalued mapping. In: Classic works of the Dempster-Shafer theory of belief functions, p. 325–339 (1967)
6. Denœux, T.: A neural network classifier based on dempster-shafer theory. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans **30**(2), 131–150 (2000)

7. Denœux, T., Kanjanatarakul, O., Sriboonchitta, S.: A new evidential k-nearest neighbor rule based on contextual discounting with partially supervised learning. International Journal of Approximate Reasoning **113**, 287–302 (2019)
8. Fan, R., Ai, X., Dahnoun, N.: Road surface 3D reconstruction based on dense subpixel disparity map estimation. IEEE Transactions on Image Processing **27**(6), 3025–3035 (2018)
9. Ha, T.: The optimum class-selective rejection rule. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(6), 608–615 (1997)
10. Ha, T.: The optimum class-selective rejection rule. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(6), 608–615 (1997)
11. He, K., Zhang, X., Ren, S., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
12. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR **abs/1704.04861** (2017)
13. Kingma, D.P., Salimans, T., Welling, M.: Variational dropout and the local reparameterization trick. Advances in neural information processing systems **28**, 2575–2583 (2015)
14. Krizhevsky, A.: Learning multiple layers of features from tiny images. University of Toronto (05 2012)
15. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. In: Advances in neural information processing systems. pp. 6402–6413 (2017)
16. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
17. LeCun, Y., Cortes, C., Burges, C.: MNIST handwritten digit database. ATT Labs **2** (2010)
18. Leng, B., Liu, Y., Yu, K., Zhang, X., Xiong, Z.: 3D object understanding with 3D convolutional neural networks. Information sciences **366**, 188–201 (2016)
19. Liu, W., Wang, X., Owens, J., Li, Y.: Energy-based out-of-distribution detection. Advances in Neural Information Processing Systems (2020)
20. Liu, Z., Pan, Q., Dezert, J., Han, J.W., He, Y.: Classifier fusion with contextual reliability evaluation. IEEE transactions on cybernetics **48**(5), 1605–1618 (2017)
21. Ma, H., Xiong, R., Wang, Y., Kodagoda, S., Shi, L.: Towards open-set semantic labeling in 3D point clouds: Analysis on the unknown class. Neurocomputing **275**, 1282–1294 (2018)
22. Ma, L., Denœux, T.: Partial classification in the belief function framework. Knowledge-Based Systems p. 106742 (2021)
23. Minary, P., Pichon, F., Mercier, D., Lefevre, E., Droit, B.: Face pixel detection using evidential calibration and fusion. International Journal of Approximate Reasoning **91**, 202–215 (2017)
24. Minary, P., Pichon, F., Mercier, D., Lefevre, E., Droit, B.: Evidential joint calibration of binary svm classifiers. Soft Computing **23**(13), 4655–4671 (2019)
25. Mortier, T., Wydmuch, M., Dembczyński, K., Hüllermeier, E., Waegeman, W.: Efficient set-valued prediction in multi-class classification. Data Mining and Knowledge Discovery pp. 1–35 (2021)
26. Mortier, T., Wydmuch, M., Dembczyński, K., Hüllermeier, E., Waegeman, W.: Efficient set-valued prediction in multi-class classification. Data Mining and Knowledge Discovery pp. 1–35 (2021)

27. Nadeem, M.S.A., Zucker, J.D., Hanczar, B.: Accuracy-rejection curves (ARCs) for comparing classification methods with a reject option. vol. 8, pp. 65–81 (2009)
28. Naeini, M.P., Cooper, G., Hauskrecht, M.: Obtaining well calibrated probabilities using bayesian binning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 29 (2015)
29. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. NIPS Workshop on Deep Learning and Unsupervised Feature Learning (2011)
30. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
31. Sadinle, M., Lei, J., Wasserman, L.: Least ambiguous set-valued classifiers with bounded error levels. Journal of the American Statistical Association **114**(525), 223–234 (2019)
32. Shafer, G.: A mathematical theory of evidence, vol. 42. Princeton university press (1976)
33. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1–9 (2015)
34. Vovk, V.: Conditional validity of inductive conformal predictors. In: Asian conference on machine learning. pp. 475–490 (2012)
35. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic learning in a random world. Springer Science & Business Media (2005)
36. Wang, J., Ju, R., Chen, Y., Liu, G., Yi, Z.: Automated diagnosis of neonatal encephalopathy on aEEG using deep neural networks. Neurocomputing **398**, 95–107 (2020)
37. Yu, F., Zhang, Y., Song, S., Seff, A., Xiao, J.: LSUN: construction of a large-scale image dataset using deep learning with humans in the loop (2015)
38. Zaffalon, M.: The naive credal classifier. Journal of Statistical Planning and Inference **105**(1), 5–21 (2002)
39. Zaffalon, M., Corani, G., Mauá, D.: Evaluating credal classifiers by utility-discounted predictive accuracy. International Journal of Approximate Reasoning **53**(8), 1282–1301 (2012)
40. Zhang, L., Yang, F., Zhang, Y.D., Zhu, Y.J.: Road crack detection using deep convolutional neural network. In: 2016 IEEE international conference on image processing (ICIP). pp. 3708–3712 (2016)
41. Zhang, Y., Liao, S.: A kernel perspective for the decision boundary of deep neural networks. In: 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence. pp. 653–660 (2020)
42. Zhou, C., Lu, X., Huang, M.: Dempster–shafer theory-based robust least squares support vector machine for stochastic modelling. Neurocomputing **182**, 145–153 (2016)